FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM71-10143-5E

FR60

32-BIT MICROCONTROLLER

MB91313A Series
HARDWARE MANUAL

FUJITSU

FR60

32-BIT MICROCONTROLLER

MB91313A Series
HARDWARE MANUAL

For the information for microcontroller supports, see the following web site.
This web site includes the **"Customer Design Review Supplement"** which provides the latest cautions on system development and the minimal requirements to be checked to prevent problems before the system development.

http://edevice.fujitsu.com/micom/en-support/

FUJITSU SEMICONDUCTOR LIMITED

Preface

■ Purpose of This Document and Intended Reader

Thank you very much for your continued patronage of Fujitsu Semiconductor products.

FR family is a line of single-chip microcontrollers based on the 32-bit high-performance RISC CPU while integrating a variety of I/O resources for embedded control applications which require high-performance, high-speed CPU processing.

This manual describes the functions and operation of the MB91313A series for engineers who actually use this semiconductor to design products. Please read this manual first.

FR, the abbreviation of FUJITSU RISC controller, is a line of products of FUJITSU SEMICONDUCTOR LIMITED.

■ Trademark

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

■ Organization of This Manual

This manual consists of the following 19 chapters and appendix.

CHAPTER 1 OVERVIEW

This chapter explains the features and basic specification of the MB91313A series.

CHAPTER 2 CPU AND CONTROL BLOCK

This chapter provides the CPU core of the FR family, covering its architecture, specifications, and instructions.

CHAPTER 3 EXTERNAL BUS INTERFACE

This chapter explains the features, block diagram, I/O pins, and registers of the external bus interface.

CHAPTER 4 I/O PORTS

This chapter gives an overview of I/O ports and describes their register configuration and functions.

CHAPTER 5 16-BIT RELOAD TIMER

This chapter explains an overview, register configuration/functions and operations of the 16-bit reload timer.

CHAPTER 6 PPG (PROGRAMMABLE PULSE GENERATOR)

This chapter gives an outline of the PPG (Programmable Pulse Generator) timer and explains the register configuration and functions and the timer operations.

CHAPTER 7 16-BIT PULSE WIDTH COUNTER

This chapter gives an overview of the 16-bit pulse width counter and explains the register configuration and functions and the counter operation.

CHAPTER 8 MULTIFUNCTION TIMER

This chapter gives an overview of the multifunction timer and explains the register configuration and functions and the timer operation.

CHAPTER 9 OTHER TIMERS

This chapter explains the main oscillation stabilization wait timer, interval timer, and watch timer.

CHAPTER 10 INTERRUPT CONTROLLER

This chapter gives an overview of the interrupt controller and explains its register configuration/functions and its operations.

CHAPTER 11 EXTERNAL INTERRUPT CONTROL UNIT

This chapters gives an overview of the external interrupt control unit and describes its register configuration/functions and its operations.

CHAPTER 12 DELAY INTERRUPT MODULE

This chapter explains the overview of the delay interrupt module, configuration/functions of the registers, and operations of the module.

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

Copyright ©2007-2010 FUJITSU SEMICONDUCTOR LIMITED All rights reserved.

CONTENTS

Main changes in this edition

Page	Section	Change Results
-	-	Changed Series name. MB91313/A → MB91313A
5	CHAPTER 1 OVERVIEW 1.2 Block Diagram	Changed Figure 1.2-1.
6	CHAPTER 1 OVERVIEW 1.3 Pin Assignment	Corrected Figure 1.3-1.
10	CHAPTER 1 OVERVIEW 1.5 List of Pin Functions	Corrected Table 1.5-1. I/O circuit type of pin number 32, 33 D/B → B
20	CHAPTER 1 OVERVIEW 1.7 Precautions on Handling the Device	Added The explanation. Turning on/off these power supplies (VDDI/Analog/VDDE) simultaneously causes no problem.
23		Deleted "● External interrupt INT17, INT19 (MB91F313 only)".
166	CHAPTER 5 16-BIT RELOAD TIMER 5.2.1 Control Status Register (TMCSR)	Corrected Figure 5.2-2.
325	CHAPTER 15 MULTI FUNCTION SERIAL INTERFACE 15.9 Notes on UART Mode	Added Item.
373	CHAPTER 15 MULTI FUNCTION SERIAL INTERFACE 15.16 Notes on CSIO Mode	Added Item.
431 to 442	CHAPTER 15 MULTI FUNCTION SERIAL INTERFACE 15.22 Examples of I^2C Flowchart	Corrected Flowchart.
443, 444	CHAPTER 15 MULTI FUNCTION SERIAL INTERFACE 15.23 Notes on I^2C Mode	Added Item.
499 to 531	CHAPTER 18 FLASH MEMORY	The overall revision.

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

CHAPTER 1
OVERVIEW

This chapter explains the features and basic specification of the MB91313A series.

1.1 Features

The FR family is a line of microcontrollers based on a 32-bit high-performance RISC CPU and integrating a variety of I/O resources for embedded control applications which require high-performance, high-speed processing by the CPU.
The microcontrollers are equipped with the multiple channels of the communication macro and most suitable for embedded applications such as television control.

■ FR CPU

- 32-bit RISC, load/store architecture, 5-stage pipeline
- Operating frequency of 33 MHz [PLL used: 16.5-MHz oscillation frequency: multiply by 2]
- 16-bit fixed-length instructions (basic instructions), one instruction per cycle
- Memory-to-memory transfer, bit manipulation, barrel shift, and other instructions ... Instructions appropriate for embedded applications
- Function entry/exit instructions, multi-register load/store instructions ... Instructions compatible with high-level languages
- Register interlock function ... Facilitating assembly-language coding
- Built-in multiplier/instruction-level support:

  Signed 32-bit multiplication ... 5 cycles

  Signed 16-bit multiplication ... 3 cycles
- Interrupts (saving of PC and PS) ... 6 cycles, 16 priority levels
- Harvard architecture enabling simultaneous execution of program access and data access
- 4 words queuing in CPU enabling advanced fetch of instructions
- Instruction-compatible with the FR family

■ Simplified External Bus Interface

Setting in the program enables the 8-bit or 16-bit multiplex bus to run.

- Maximum operating frequency of 16.5 MHz
- 8-/16- bit data/address multiplex I/O
- Totally independent 4-area chip select output that can be defined at a minimum of 64 KB
- Basic bus cycle ... 3 cycles
- Automatic wait cycle generator that can be programmed for each area
- Unused data/address/control signal pins can be used for general-purpose I/O

■ Built-in Memory

- Flash memory 544 KB, RAM/32 KB

■ DMAC (DMA Controller)

- 5 channels
- 2 transfer sources: internal peripheral/software
- Addressing mode: 20/24-bit full address specifications (increase/decrease/fixed)
- Transfer mode: burst transfer/step transfer/block transfer
- Transfer data size: can be selected from 8, 16, or 32 bits

■ Bit Search Module (Used by REALOS)

Searches for the position of the first bit varying between 1 and 0 in the MSB of a word.

■ 16-bit Reload Timer (Including One Channel for REALOS)

- 6 channels
- Internal clock: selectable from among results of dividing the machine clock frequency by 2, 8, and 32

■ Multi Function Serial Interface

- 11 channels
- Full-duplex double buffer
- 4 types communication mode: asynchronous (start-stop synchronous), clock-synchronous communication (maximum 8.25 Mbps), I^2C standard mode (maximum 100 kbps), and I^2C high-speed mode (maximum 400 kbps)
- Parity selectable from enabled or disabled
- Built-in baud rate generator for each channel
- Various error detection functions: parity, frame, overrun
- External clock that can be used as transfer clock
- ch.0 to ch.2 : the DMA transfer / the FIFO function with 16 bytes for each of the transmission and reception
- ch.8 to ch.10 : 5-V tolerant.
- ch.8 : the open drain output
- I^2C bridge function (among 0 channel, 1 channel, and 2 channel)
- SPI mode

■ Interrupt Controller

- Total of 24 external interrupts (external interrupt pins INT23 to INT0)
- Interrupts from internal peripherals
- Programmable priority level (16 levels)
- Available for wakeup in STOP mode

■ 10-bit A/D Converter

- 10 channels
- Successive approximation type: conversion time : approximately 7.94 µs
- Conversion modes: single conversion mode and scan conversion mode
- Activation sources: software and external trigger

■ PPG

- 4 channels
- 16-bit counter, 16-bit data register with buffer to set the cycle
- Internal clock: selectable from among results of dividing the machine clock frequency by 1, 4, 16, and 64
- Automatic cycle setting supported with the DMA transfer
- Remote control transmission support function
- Open drain output

■ PWC

- 1 channel (1 input)
- 16-bit up counter
- Simplified LOWPASS filter

■ Multifunction Timer

- 4 channels
- Removes the noise, if lower than specified clock, using the Low Pass filter
- Pulse width can be measured using 7 types of clock signal
- Event count function from pin input
- Interval timer function using 7 types of clock and external input clock
- Built-in HSYNC counter mode

■ HDMI-CEC/Remote Control Reception

- 2 channels
- HDMI-CEC reception function (with automatic ACK response function)
- Remote control reception function (with 4 byte reception buffers)

■ Other Interval Timers

- Watch timer (32 kHz, Max 60 s count)
- Watchdog timer

■ I/O Ports

Maximum 86 ports

■ Other Features

- Built-in oscillation circuit as a clock source
- INITX provided as a reset pin
- Watchdog timer and software resets available
- Supporting stop mode and sleep mode as low-power consumption modes
- Gear function
  - Built-in time-base timer
  - 5 V tolerant I/O (for a part of pins)
- Package: LQFP-120, 0.5 mm pitch, 16 mm × 16 mm
- CMOS technology: 0.18 µm
- Power supply voltage 3.3±0.3 V, 1.8±0.15 V, 2 power supply units
- I^2C license

  Purchase of Fujitsu I^2C components conveys a license under the Philips I^2C Patent Rights to use, these components in an I^2C system provided that the system conforms to the I^2C Standard Specification as defined by Philips.

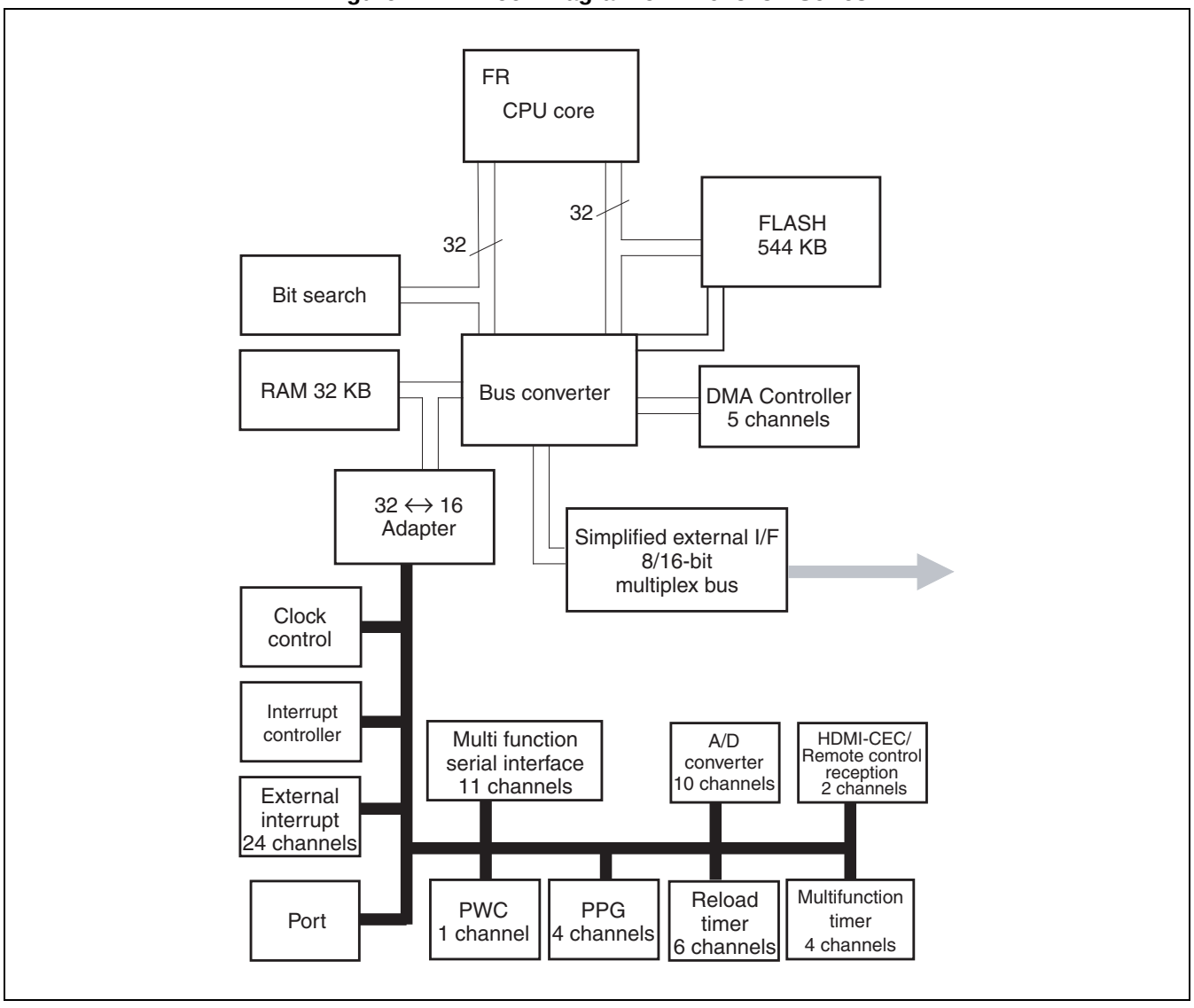
MB91313A Series

1.2 Block Diagram

Figure 1.2-1 shows a block diagram of MB91313A Series.

■ Block Diagram of MB91313A Series

Figure 1.2-1 Block Diagram of MB91313A Series

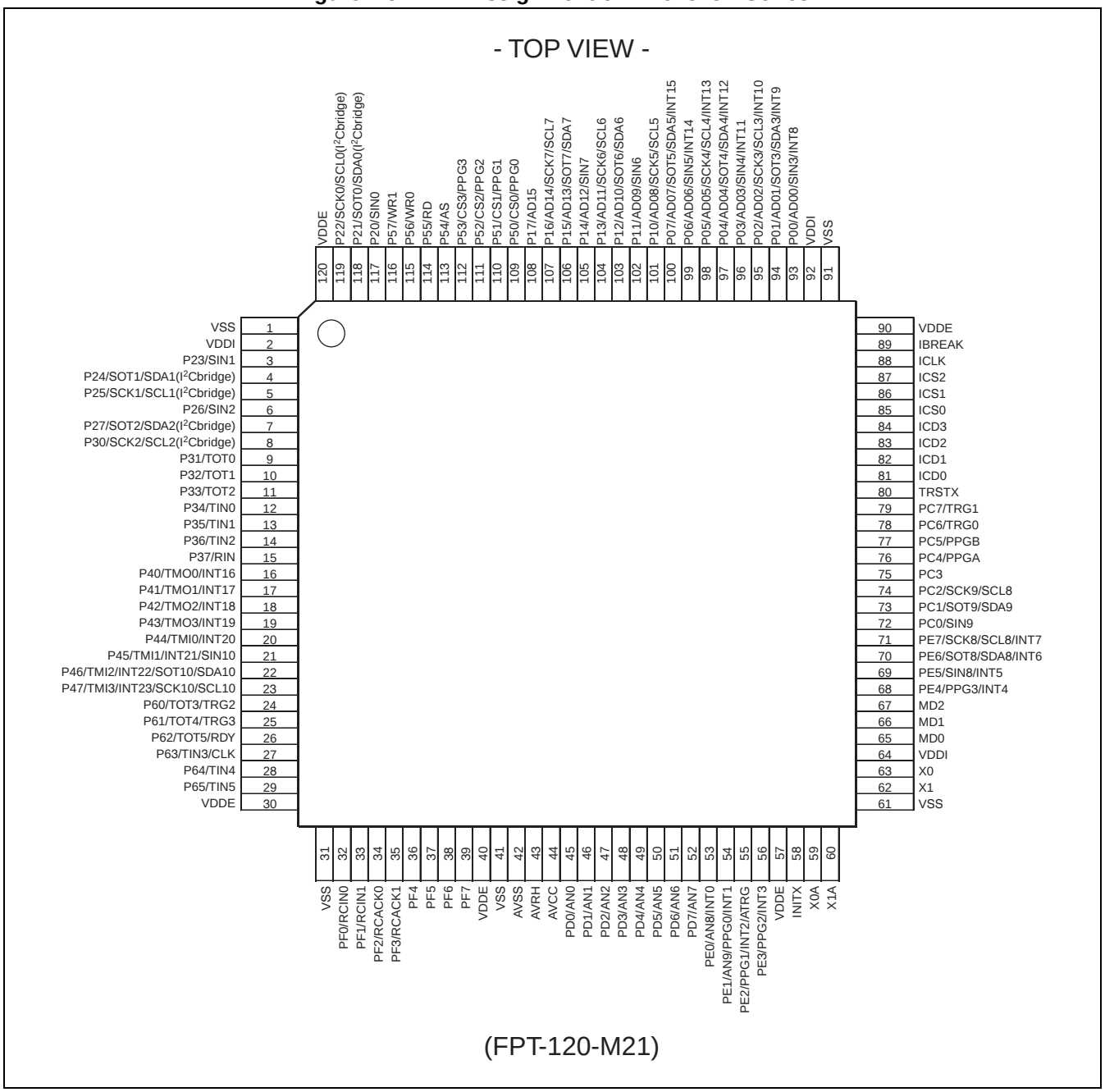


1.3 Pin Assignment

Figure 1.3-1 shows the pin assignment of MB91313A Series.

■ Pin Assignment of MB91313A Series

Figure 1.3-1 Pin Assignment of MB91313A Series



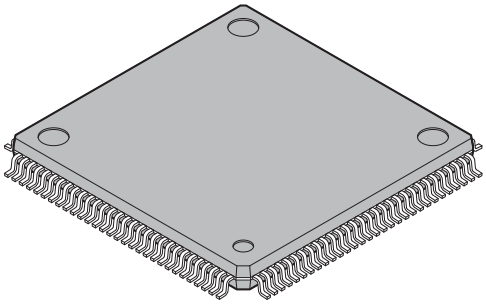
(FPT-120-M21)

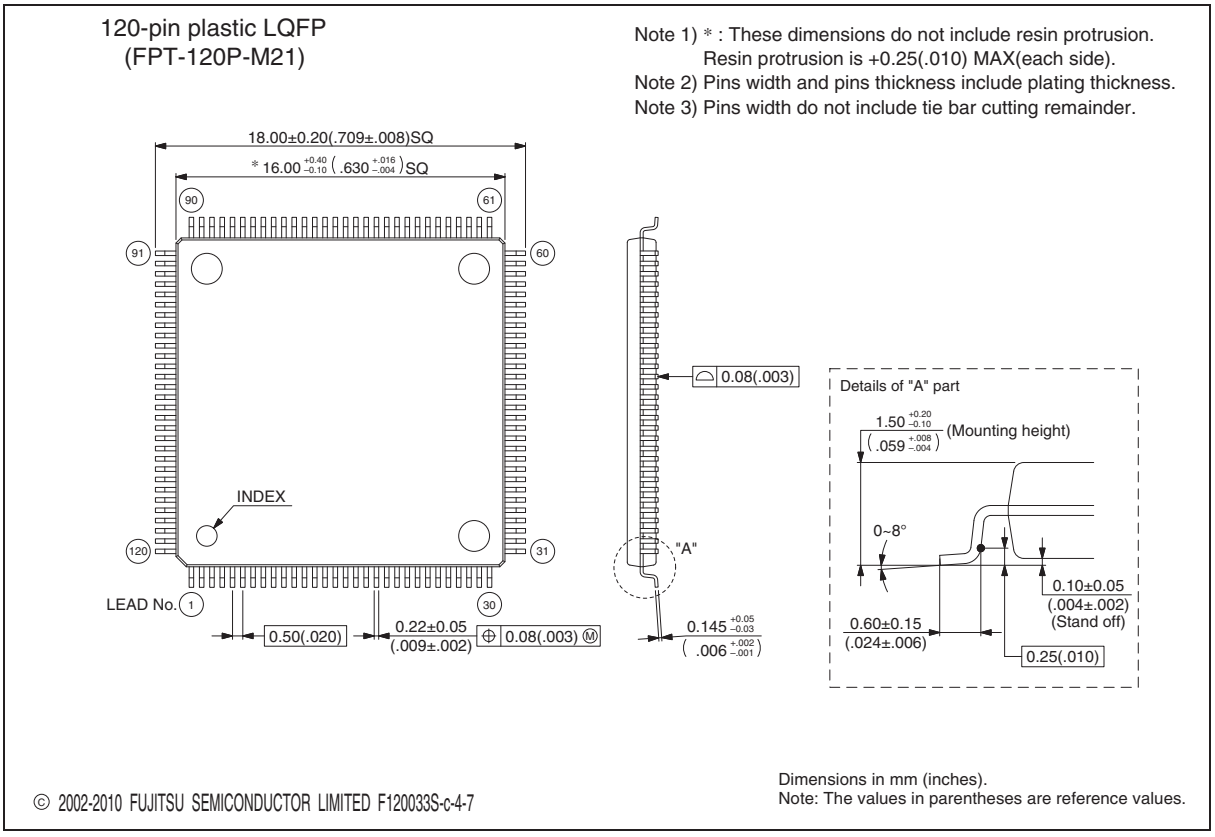
MB91313A Series

1.4 Package Dimensions

Figure 1.4-1 shows the package dimensions of MB91313A Series.

Figure 1.4-1 Package Dimensions of MB91313A Series

120-pin plastic LQFP  (FPT-120P-M21)	Lead pitch	0.50 mm
	Package width × package length	16.0 × 16.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.88 g
	Code (Reference)	P-LFQFP120-16×16-0.50



Please check the latest package dimensions at the following URL.

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

1.5 List of Pin Functions

Table 1.5-1 lists the pin functions of MB91313A Series.

■ Pin Functions of MB91313A Series

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

Pin No.	Pin name	I/O circuit type *1	Function
1	VSS	-	GND pin
2	VDDI	-	1.8 V power supply pin
3	P23	D	General-purpose port
	SIN1		Serial data input pin
4	P24	L	General-purpose port
	SOT1/SDA1 (I^2C bridge)		Serial data output pin I^2C data I/O pin
5	P25	L	General-purpose port
	SCK1/SCL1 (I^2C bridge)		Clock pulse I/O pin for serial communication I^2C clock I/O pin
6	P26	D	General-purpose port
	SIN2		Serial data input pin
7	P27	L	General-purpose port
	SOT2/SDA2 (I^2C bridge)		Serial data output pin I^2C data I/O pin
8	P30	L	General-purpose port
	SCK2/SCL2 (I^2C bridge)		Clock pulse I/O pin for serial communication I^2C clock I/O pin
9	P31	D	General-purpose port
	TOT0		Output pin for reload timer
10	P32	D	General-purpose port
	TOT1		Output pin for reload timer
11	P33	D	General-purpose port
	TOT2		Output pin for reload timer
12	P34	D	General-purpose port
	TIN0		Event input pin for reload timer
13	P35	D	General-purpose port
	TIN1		Event input pin for reload timer
14	P36	D	General-purpose port
	TIN2		Event input pin for reload timer

Table 1.5-1 List of Pin Functions (2 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
15	P37	D	General-purpose port
	RIN		PWC input pin
16	P40	B	General-purpose port
	TMO0		Multifunction timer output pin
	INT16		External interrupt request input pin
17	P41	B	General-purpose port
	TMO1		Multifunction timer output pin
	INT17		External interrupt request input pin
18	P42	B	General-purpose port
	TMO2		Multifunction timer output pin
	INT18		External interrupt request input pin
19	P43	B	General-purpose port
	TMO3		Multifunction timer output pin
	INT19		External interrupt request input pin
20	P44	B	General-purpose port
	TMI0		Multifunction timer input pin
	INT20		External interrupt request input pin
21	P45	B	General-purpose port
	TMI1		Multifunction timer input pin
	INT21		External interrupt request input pin
	SIN10		Serial data input pin
22	P46	B	General-purpose port
	TMI2		Multifunction timer input pin
	INT22		External interrupt request input pin
	SOT10/SDA10		Serial data output pin I^2C data I/O pin
23	P47	B	General-purpose port
	TMI3		Multifunction timer input pin
	INT23		External interrupt request input pin
	SCK10/SCL10		Clock pulse input/output pin for serial communication I^2C clock I/O pin
24	P60	C	General-purpose port
	TOT3		Output pin for reload timer
	TRG2		PPG trigger input pin

Table 1.5-1 List of Pin Functions (3 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
25	P61	C	General-purpose port
	TOT4		Output pin for reload timer
	TRG3		PPG trigger input pin
26	P62	C	General-purpose port
	TOT5		Output pin for reload timer
	RDY		External ready input pin
27	P63	C	General-purpose port
	TIN3		Event input pin for reload timer
	CLK		External clock output pin
28	P64	C	General-purpose port
	TIN4		Event input pin for reload timer
29	P65	C	General-purpose port
	TIN5		Event input pin for reload timer
30	VDDE	-	3.3 V power supply
31	VSS	-	GND pin
32	PF0	B	General-purpose port
	RCIN0		HDMI-CEC/Remote control 0 I/O pin
33	PF1	B	General-purpose port
	RCIN1		HDMI-CEC/Remote control 1 I/O pin
34	PF2	D	General-purpose port
	RCACK0		HDMI-CEC/Remote control 0ACK output pin
35	PF3	D	General-purpose port
	RCACK1		HDMI-CEC/Remote control 1ACK output pin
36	PF4	D	General-purpose port
37	PF5	D	General-purpose port
38	PF6	D	General-purpose port
39	PF7	D	General-purpose port
40	VDDE	-	3.3 V power supply
41	VSS	-	GND pin
42	AVSS	-	A/D converter GND pin
43	AVRH	-	A/D converter reference voltage pin
44	AVCC	-	A/D converter power supply pin
45	PD0	L	General-purpose port pin
	AN0		A/D converter analog input pin
46	PD1	L	General-purpose port
	AN1		A/D converter analog input pin

Table 1.5-1 List of Pin Functions (4 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
47	PD2	L	General-purpose port
	AN2		A/D converter analog input pin
48	PD3	L	General-purpose port
	AN3		A/D converter analog input pin
49	PD4	L	General-purpose port
	AN4		A/D converter analog input pin
50	PD5	L	General-purpose port
	AN5		A/D converter Analog input pin
51	PD6	L	General-purpose port
	AN6		A/D converter Analog input pin
52	PD7	L	General-purpose port
	AN7		A/D converter Analog input pin
53	PE0	L	General-purpose port
	AN8		A/D converter Analog input pin
	INT0		External interrupt request input pin
54	PE1	L	General-purpose port
	AN9		A/D converter Analog input pin
	PPG0 *2		Output pin for PPG
	INT1		External interrupt request input pin
55	PE2	B	General-purpose port
	PPG1 *2		Output pin for PPG
	INT2		External interrupt request input pin
	ATRG		Trigger input pin for A/D converter
56	PE3	B	General-purpose port
	PPG2 *2		Output pin for PPG
	INT3		External interrupt request input pin
57	VDDE	-	3.3 V power supply
58	INITX	G	Initial reset pin
59	X0A	A	Sub clock input pin
60	X1A	A	Sub clock output pin
61	VSS	-	GND pin
62	X1	A	Main clock output pin
63	X0	A	Main clock input pin
64	VDDI	-	1.8 V power supply
65	MD0	F	Mode pin
66	MD1	F	Mode pin

Table 1.5-1 List of Pin Functions (5 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
67	MD2	F	Mode pin
68	PE4	B	General-purpose port
	PPG3 *2		Output pin for PPG
	INT4		External interrupt request input pin
69	PE5	B	General-purpose port
	SIN8		Serial data input pin
	INT5		External interrupt request input pin
70	PE6	B	General-purpose port
	SOT8/SDA8		External interrupt request input pin I^2C data I/O pin
	INT6		Serial data output pin
71	PE7	B	General-purpose port
	SCK8/SCL8		Clock pulse input/output pin for serial communication I^2C clock I/O pin
	INT7		External interrupt request input pin
72	PC0	B	General-purpose port
	SIN9		Serial data input pin
73	PC1	B	General-purpose port
	SOT9/SDA9		Serial data output pin I^2C data I/O pin
74	PC2	B	General-purpose port
	SCK9/SCL9		Clock pulse input/output pin for serial communication I^2C clock I/O pin
75	PC3	B	General-purpose port
76	PC4	B	General-purpose port
	PPGA		Output pin for PPG
77	PC5	B	General-purpose port
	PPGB		Output pin for PPG
78	PC6	B	General-purpose port
	TRG0		PPG trigger input
79	PC7	B	General-purpose port
	TRG1		PPG trigger input
80	TRSTX	G	Reset pin for development tool
81	ICD0	K	Data pin for development tool
82	ICD1	K	Data pin for development tool
83	ICD2	K	Data pin for development tool
84	ICD3	K	Data pin for development tool

Table 1.5-1 List of Pin Functions (6 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
85	ICS0	H	Status pin for development tool
86	ICS1	H	Status pin for development tool
87	ICS2	H	Status pin for development tool
88	ICLK	H	Clock pin for development tool
89	IBREAK	I	Break pin for development tool
90	VDDE	-	3.3 V power supply
91	VSS	-	GND pin
92	VDDI	-	1.8 V power supply
93	P00	C	General-purpose port
	AD00		External address/data bus I/O pin
	SIN3		Serial data input pin
	INT8		External interrupt request input pin
94	P01	C	General-purpose port
	AD01		External address/data bus I/O pin
	SOT3/SDA3		Serial data output pin I^2C data I/O pin
	INT9		External interrupt request input pin
95	P02	C	General-purpose port
	AD02		External address/data bus I/O pin
	SCK3/SCL3		Clock pulse input/output pin for serial communication I^2C clock I/O pin
	INT10		External interrupt request input pin
96	P03	C	General-purpose port
	AD03		External address/data bus I/O pin
	SIN4		Serial data input pin
	INT11		External interrupt request input pin
97	P04	C	General-purpose port
	AD04		External address/data bus I/O pin
	SOT4/SDA4		Serial data output pin I^2C data I/O pin
	INT12		External interrupt request input pin
98	P05	C	General-purpose port
	AD05		External address/data bus I/O pin
	SCK4/SCL4		Clock pulse input/output pin for serial communication I^2C clock I/O pin
	INT13		External interrupt request input pin

Table 1.5-1 List of Pin Functions (7 / 8)

Pin No.	Pin name	I/O circuit type *1	Function
99	P06	C	General-purpose port
	AD06		External address/data bus I/O pin
	SIN5		Serial data input pin
	INT14		External interrupt request input pin
100	P07	C	General-purpose port
	AD07		External address/data bus I/O pin
	SOT5/SDA5		Serial data output pin I^2C data I/O pin
	INT15		External interrupt request input pin
101	P10	C	General-purpose port
	AD08		External address/data bus I/O pin
	SCK5/SCL5		Clock pulse input/output pin for serial communication I^2C clock I/O pin
102	P11	C	General-purpose port
	AD09		External address/data bus I/O pin
	SIN6		Serial data input pin
103	P12	C	General-purpose port
	AD10		External address/data bus I/O pin
	SOT6/SDA6		Serial data output pin I^2C data I/O pin
104	P13	C	General-purpose port
	AD11		External address/data bus I/O pin
	SCK6/SCL6		Clock pulse input/output pin for serial communication I^2C clock I/O pin
105	P14	C	General-purpose port
	AD12		External address/data bus I/O pin
	SIN7		Serial data input pin
106	P15	C	General-purpose port
	AD13		External address/data bus I/O pin
	SOT7/SDA7		Serial data output pin I^2C data I/O pin
107	P16	C	General-purpose port
	AD14		External address/data bus I/O pin
	SCK7/SCL7		Clock pulse input/output pin for serial communication I^2C clock I/O pin
108	P17	C	General-purpose port
	AD15		External address/data bus I/O pin

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

Pin No.	Pin name	I/O circuit type *1	Function
109	P50	C	General-purpose port
	CS0X		External chip select pin
	PPG0 *2		Output pin for PPG
110	P51	C	General-purpose port
	CS1X		External chip select pin
	PPG1 *2		Output pin for PPG
111	P52	C	General-purpose port
	CS2X		External chip select pin
	PPG2 *2		Output pin for PPG
112	P53	C	General-purpose port
	CS3X		External chip select pin
	PPG3 *2		Output pin for PPG
113	P54	C	General-purpose port
	ASX		External address strobe output pin
114	P55	C	General-purpose port
	RDX		External read strobe output pin
115	P56	C	General-purpose port
	WR0X		External data bus write strobe output pin
116	P57	C	General-purpose port
	WR1X		External data bus write strobe output pin
117	P20	D	General-purpose port
	SIN0		Serial data input pin
118	P21	L	General-purpose port
	SOT0/SDA0 (I^2C bridge)		Serial data output pin I^2C data I/O pin
119	P22	L	General-purpose port
	SCK0/SCL0 (I^2C bridge)		Clock pulse input/output pin for serial communication I^2C clock I/O pin
120	VDDE	-	3.3 V power supply pin

*1: For the I/O circuit type, refer to "1.6 I/O Circuit Types".
*2: For PPG0, PPG1, PPG2, PPG3 on pins 54, 55, 56, 68, and on pins 109, 110, 111, 112 respectively, select and use each one of them via PFR. For the PFR, refer to "CHAPTER 4 I/O PORTS".

1.6 I/O Circuit Types

Table 1.6-1 lists the I/O circuit types of MB91313A series.

■ I/O Circuit Types

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

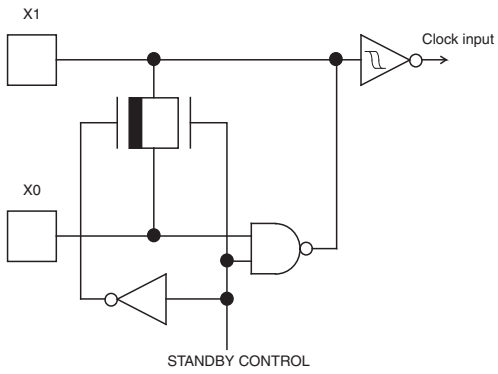
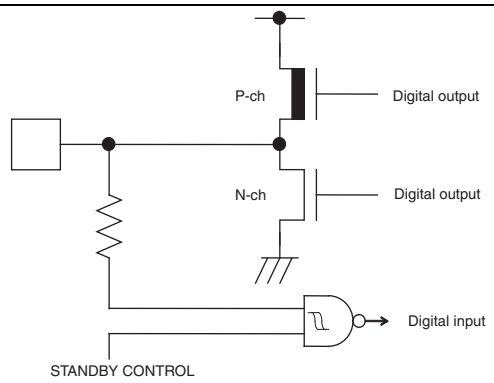
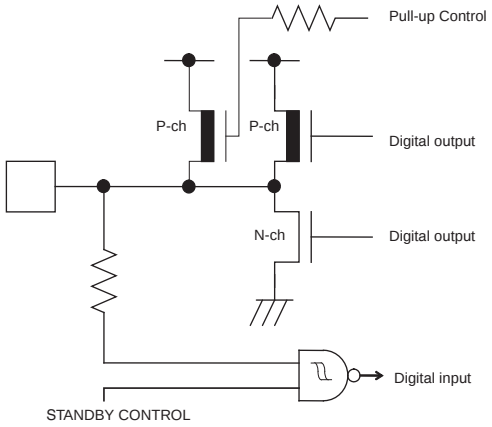
Type	Circuit type	Remarks
A		Oscillation circuit Built-in feedback resistor X0-X1: 1MΩ X0A-X1A: None
B		• CMOS level output I_{OH} = 4 mA • CMOS level hysteresis input $V_{IH} = 0.7 \times V_{DDE}$ • Standby control supported • 5 V tolerant
C		• CMOS level output I_{OH} = 4mA • CMOS level hysteresis input $V_{IH} = 0.8 \times V_{DDE}$ • Standby control supported • Pull-up control supported • Pull-up resistor available (33 kΩ)

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

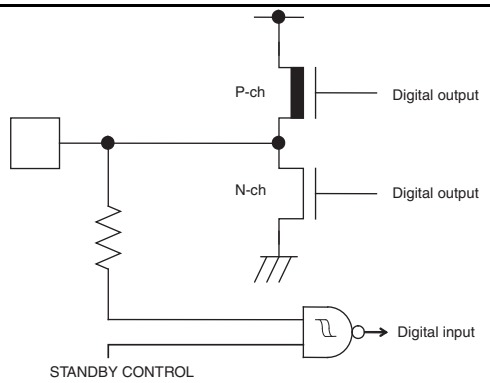
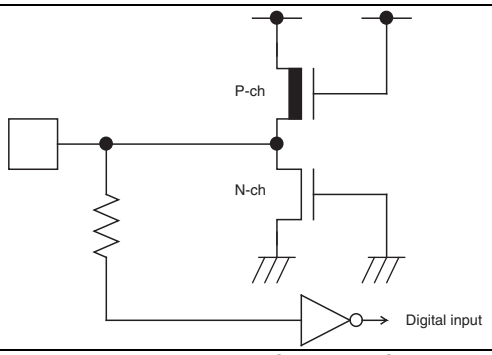
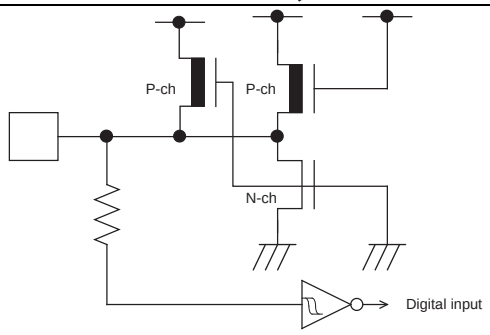
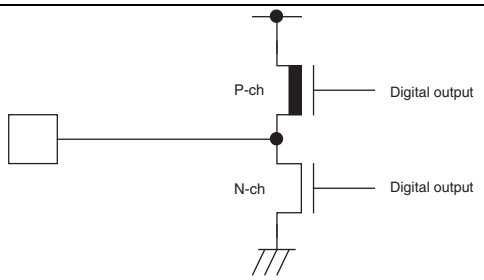
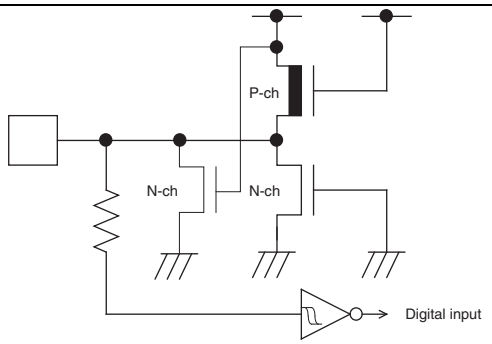
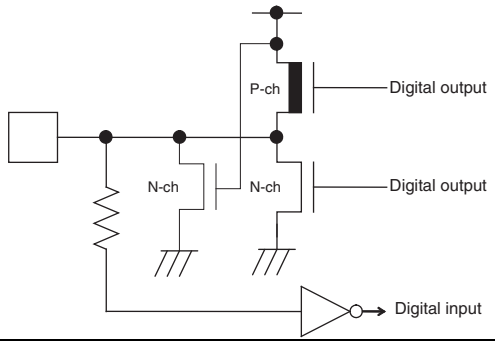
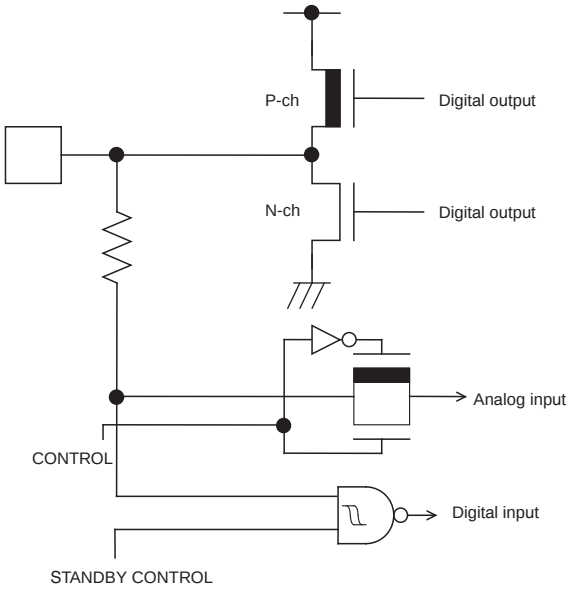
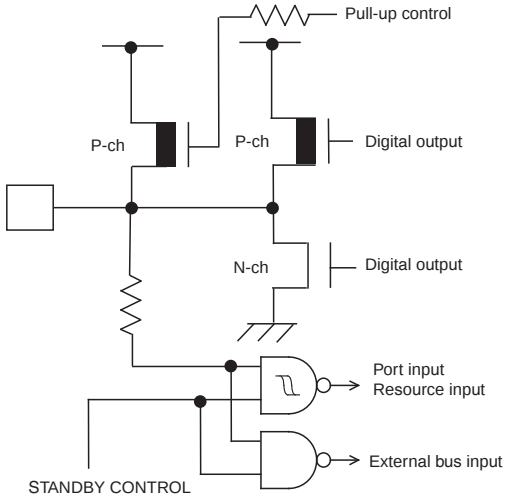
Type	Circuit type	Remarks
D		• CMOS level output I_{OH} = 4 mA • CMOS level hysteresis input $V_{IH} = 0.8 \times V_{DDE}$ • Standby control supported • Pull-up resistor not available
F		• CMOS level input • Standby control not supported
G		• CMOS hysteresis input • Pull-up resistor available
H		CMOS level output
I		• CMOS hysteresis input • Pull-down resistor available • Standby control not supported

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

Type	Circuit type	Remarks
K		• CMOS level output • CMOS level input • Standby control not supported • Pull-down resistor available
L		• CMOS level output • CMOS level hysteresis input • Standby control supported • Analog input with switch
O		• CMOS level output I_{OH}=4mA • CMOS input (external bus IF) • CMOS level hysteresis input (port, resource) V_{IH}=0.8 × VDDE • Standby control supported • Pull-up control supported • Pull-up resistor available(33kΩ)

1.7 Precautions on Handling the Device

This section contains information on the prevention of latch-ups, pin processing, handling of circuits, input at power-on and so on.

■ Preventing a Latch-up

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

■ Unused Input Pins

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

■ Power Supply Pins

In MB91313A series, devices including multiple of VDDE pins, VDDI pins and VSS pins are designed as follows; pins necessary to be at the same potential are interconnected internally to prevent malfunctions such as latch-up. All of the power supply pin and GND pin must be externally connected to the power supply and ground respectively in order to reduce unnecessary radiation, to prevent strobe signal malfunctions due to the ground level rising and to follow the total output current ratings. Furthermore, the VDDE pins, VDDI pins and VSS pins of the MB91313A series must be connected to the current supply source via a low impedance. It is also recommended to connect a ceramic capacitor of approximately 0.1μF as a bypass capacitor between VDDE pins, VDDI pins and VSS pins near this device.

■ Crystal Oscillator Circuit

Noise in proximity to the X0 and X1 (X0A, X1A) pins can cause the device to operate abnormally. Printed circuit boards should be designed so that the X0 (X0A) and X1 (X1A) pins, and crystal oscillator, as well as bypass capacitors connected to ground, are located near the device and ground.

It is recommended that the printed circuit board artwork be designed such that the X0 and X1 pins or X0A and X1A pins are surrounded by ground plane for the stable operation.

Please request the oscillator manufacturer to evaluate the oscillational characteristics of the crystal and this device.

■ Mode Pins (MD0 to MD2)

When using mode pins, connect them directly to power supply pin or GND pin. To prevent the device from entering test mode accidentally due to noise, minimize the lengths of the patterns between each mode pin and power supply pin or GND pin on the printed circuit board as possible and connect them with low impedance.

■ Operation at Power-On

Ensure that the INITX pin is reset and the settings are initialized (INIT) immediately after the power is turned on.

Maintain the "L" level input to the INITX pin during the stabilization wait time immediately after the power on to ensure the stabilization wait time as required by the oscillator circuit (the stabilization wait time is reset to the minimum value when INIT is asserted using the INITX pin).

■ Note on Oscillator Input at Power-On

At power-on, ensure that the clock is input until the oscillator stabilization wait time has elapsed.

■ Notes on the Turning On/Off VDDI Pin (1.8 V Internal Power Supply) and VDDE Pin (3.3 V External Pin Power Supply)

Do not apply only VDDE pin (external power supply) voltage continuously (more than one minute) while the VDDI pin (internal power supply) is disconnected as it will adversely affect the reliability of the LSI.

When the VDDE pin (external power supply) returns from the off state to the on state, the circuit may not be able to maintain its internal state, for example, due to power supply noise.

Power on	VDD pin (internal power supply) → VDDE pin (external power supply) → Analog→ Signal
Power off	Signal → Analog→ VDDE pin (external power supply) → VDDI pin (internal power supply)

Turning on/off these power supplies (VDDI/Analog/VDDE) simultaneously causes no problem.

When the power is turned on, the states of the output pins may remain undefined until the internal power supply becomes stable.

■ Notes on Using an External Clock

When using the external clock as a general rule you should simultaneously supply X0 (X0A) and X1 (X1A) pins. And also, the clock signal to X0 (X0A) should be supplied a clock signal with the reverse phase to X1 (X1A) pins. However, in this case the stop mode (oscillation stop mode) must not be used (This is because the X1 (X1A) pin stops at "H" output in STOP mode). Furthermore, supply a clock to X0 (X0A) pin only if the device is operating in less than 12.5 MHz.

Figure 1.7-1 Using an External Clock (Normal Method)

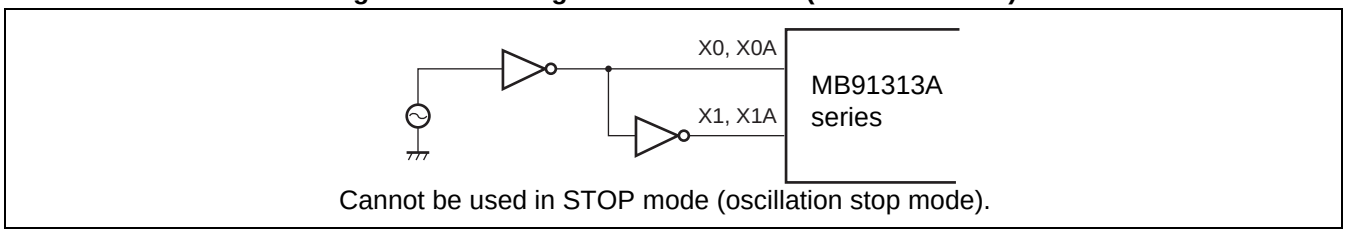
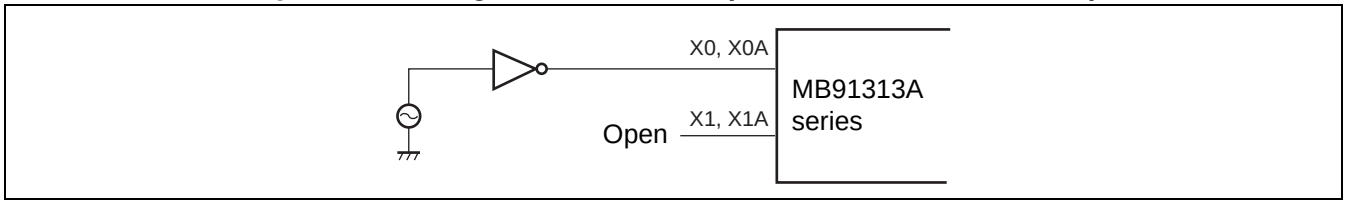


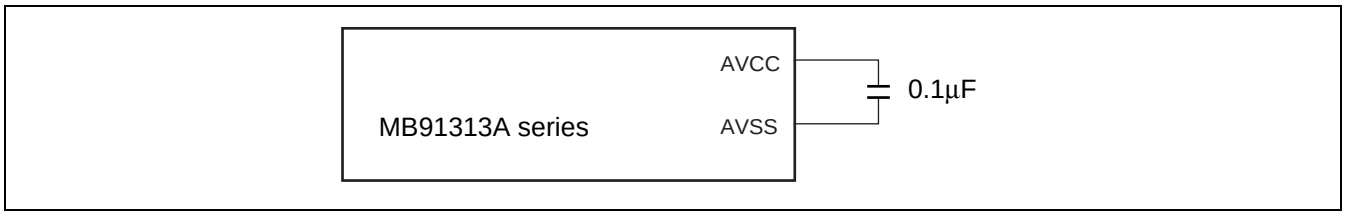
Figure 1.7-2 Using an External Clock (available at 12.5 MHz or less)



Note: : When operating at a frequency of 10 MHz, the delay between the X0 (X0A) and X1 signals should be less than 15 ns.

■ AVCC Pin

The MB91313A has a built-in A/D converter. A capacitor of approximately 0.1µF must be connected between the AVCC pin and AVSS pin.



■ Notes when not Using the Emulator

To operate the evaluation MCU on the user system without connecting the emulator, treat each input pin on the evaluation MCU connected to the emulator interface on the user system as shown below.

Note that switching circuits or other measures may be needed on the user system.

Table 1.7-1 Emulator Interface Pin Treatment

Evaluation MCU Pin Name	Pin Connection
TRSTX	Connect to the reset output circuit on the user system.
INITX	Connect to the reset output circuit on the user system.
Other Pins	Open

■ Notes on Selecting PLL Clocks

If the crystal oscillator is disconnected or the clock input stops while the PLL clock is selected, the microcontroller may continue to operate at the free-running frequency of the self-oscillating circuit within the PLL. However, this operation is not guaranteed.

■ Restrictions

● Clock control block

When an "L" level is input to the INITX pin, ensure that it is maintained for the duration of the oscillation stabilization wait time.

● Bit Search Module

The bit search data register for 0-detection (BSD0), bit search data register for 1-detection (BSD1), and bit search data register for change point detection (BSDC) can be accessed in word.

● I/O Ports

Ports can only be accessed in byte.

● Low Power Consumption Mode

- To place the device in standby mode, use the synchronous standby mode (set with bit 8 (SYNCS bit) of the timebase counter control register, TBCR) and be sure to use the following sequence :

```
(LDI#value_of_standby, r0)   ; value_of_standby is the data to write to STCR
(LDI#_STCR, R12)             ; _STCR is the address of STCR (481H)
STB     R0, @R12             ; Write to the standby control register (STCR)
LDUB    @R12, R0             ; Read STCR for synchronous standby
LDUB    @R12, R0             ; Perform an additional dummy read of STCR
NOP                          ; 5 × NOP for timing adjustment
NOP
NOP
NOP
NOP
```

- Do not perform any of the following actions when using the monitor debugger.
  - Set a breakpoint within the sequence of instructions shown above
  - Perform step execution of the sequence of instructions shown above

● Notes on the PS register

Some instructions write to the PS register in advance before executing. When a debugger is being used, execution may break within an interrupt handler routine, or the values of the flags within the PS register may be updated due to exception processing. However, the microcontroller is designed to reprocess correctly after returning from the EIT, and to execute before and after the EIT proceeds according to the specifications.

- In any following situation, the previous instructions before a DIV0U or DIV0S instruction may take the processing in (1) to (3).
  - A user interrupt or NMI is accepted
  - Step execution is performed
  - A break occurs due to a data event or by being selected from the emulator menu
    1. The D0 and D1 flags are updated in advance.
    2. The EIT handling routine (user interrupt/NMI or emulator) is executed.
    3. Upon returning from the EIT, the DIV0U or DIV0S instruction is executed and the D0/D1 flags are updated back to the same value as in step (1).
- If any of the OR CCR, ST ILM, or MOV Ri, PS instructions are executed to enable a user interrupt or NMI interrupt source when that interrupt has occurred, the following operation will be performed.
  1. The PS register is updated in advance.
  2. The EIT handling routine (user interrupt/NMI or emulator) is executed.
  3. Upon returning from the EIT, the above instructions are executed and the PS registers are updated back to the same value as in step (1).

● Watchdog timer

The watchdog timer has a function to monitors the program to check that it delays a reset within a certain period of time, and resets the CPU if the program runs out of control and fails to delay the reset. Once the watchdog timer has been enabled, it keeps running until reset. As an exception, the reset is automatically delayed in conditions where the execution of the CPU program stops. It is possible that the watchdog timer will not be triggered if these conditions arise as a result of the system running out of control. In that case, please reset (INIT) using the external INITX pin.

● Notes on using the A/D converter

Do not supply a voltage higher than the VDDE pin to the AVCC pin.

● Software reset in synchronous mode

When using the software reset in synchronous mode, the following two conditions should be satisfied before setting the SRST bit in STCR (standby control register) to "0".

- The interrupt enable flag (I-Flag) is set to interrupts disabled (I-Flag = 0) .
- The NMI is not being used.

CHAPTER 2

CPU AND CONTROL BLOCK

This chapter provides the CPU core of the FR family, covering its architecture, specifications, and instructions.

2.1 Memory Space

The logical address space of the FR family is 4GB (2^{32} locations) and the CPU performs linear access.

■ Direct Addressing Area

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

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

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

⇒ byte data access : 000_H to $0FF_H$

⇒ half word data access : 000_H to $1FF_H$

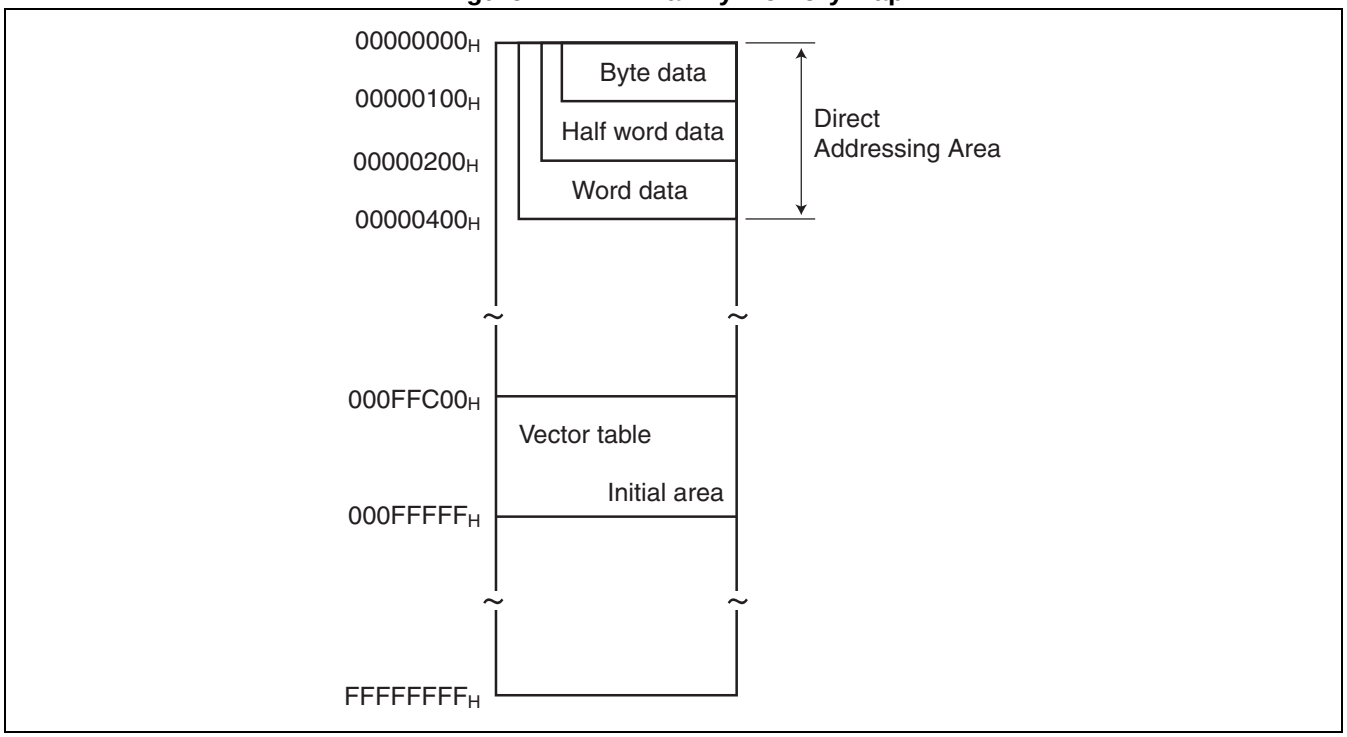
⇒ word data access : 000_H to $3FF_H$

■ FR Family Memory Map

Figure 2.1-1 shows the memory map of the FR family.

The FR family has a 32-bit linear address space.

Figure 2.1-1 FR Family Memory Map



● Initial Area of the Vector Table

The area from "$000FFC00_H$" to "$000FFFFF_H$" is used as the initial area of the EIT vector table.

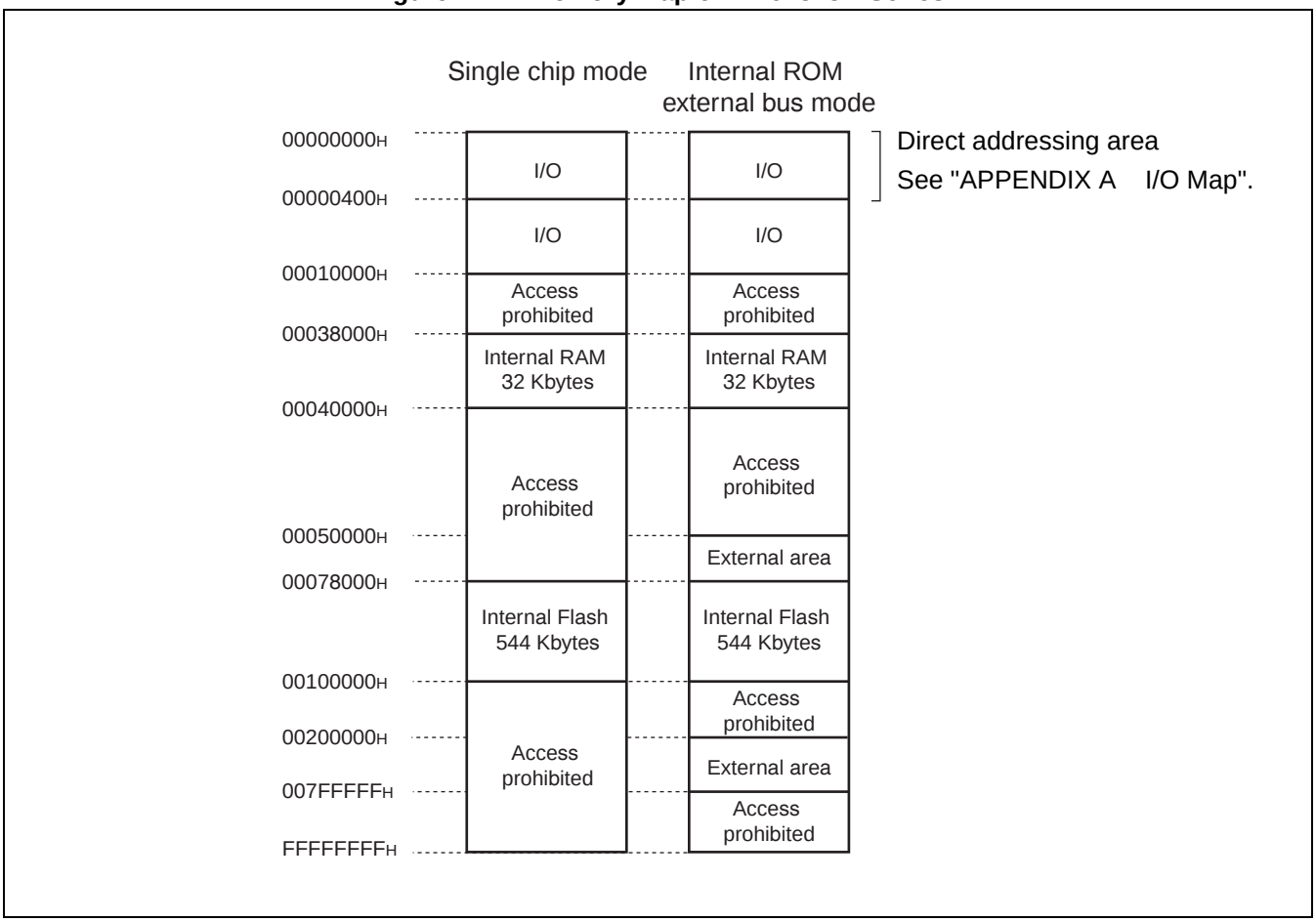
The vector table to be used during EIT processing can be placed at any address by rewriting TBR. However, the table will be relocated to this address when initialized by a reset.

MB91313A Series

■ Memory Map of MB91313A Series

Figure 2.1-2 shows the memory map.

Figure 2.1-2 Memory Map of MB91313A Series



2.2 Internal Architecture

As well as adopting a RISC architecture, the FR family CPU is a high performance core featuring advanced instructions for embedded applications.

■ Features of Internal Architecture

- Adoption of a RISC architecture
  Basic instruction: one instruction = one cycle
- 32-bit architecture
  General-purpose registers: 32 bits × 16 registers
- Linear memory space of 4 GB
- Multipliers provided
  32-bit by 32-bit multiplication: 5 cycles
  16-bit by 16-bit multiplication: 3 cycles
- Reinforced interrupt processing function
  High-speed response speed (6 cycles)
  Multiple interruption supported
  Level mask function (16 levels)
- Reinforced instruction for I/O operation
  Memory-to-memory transfer instruction
  Bit manipulation instruction
- Basic instruction word length: 16 bits
- Low-power consumption
  Sleep mode / Stop mode
  Gear function

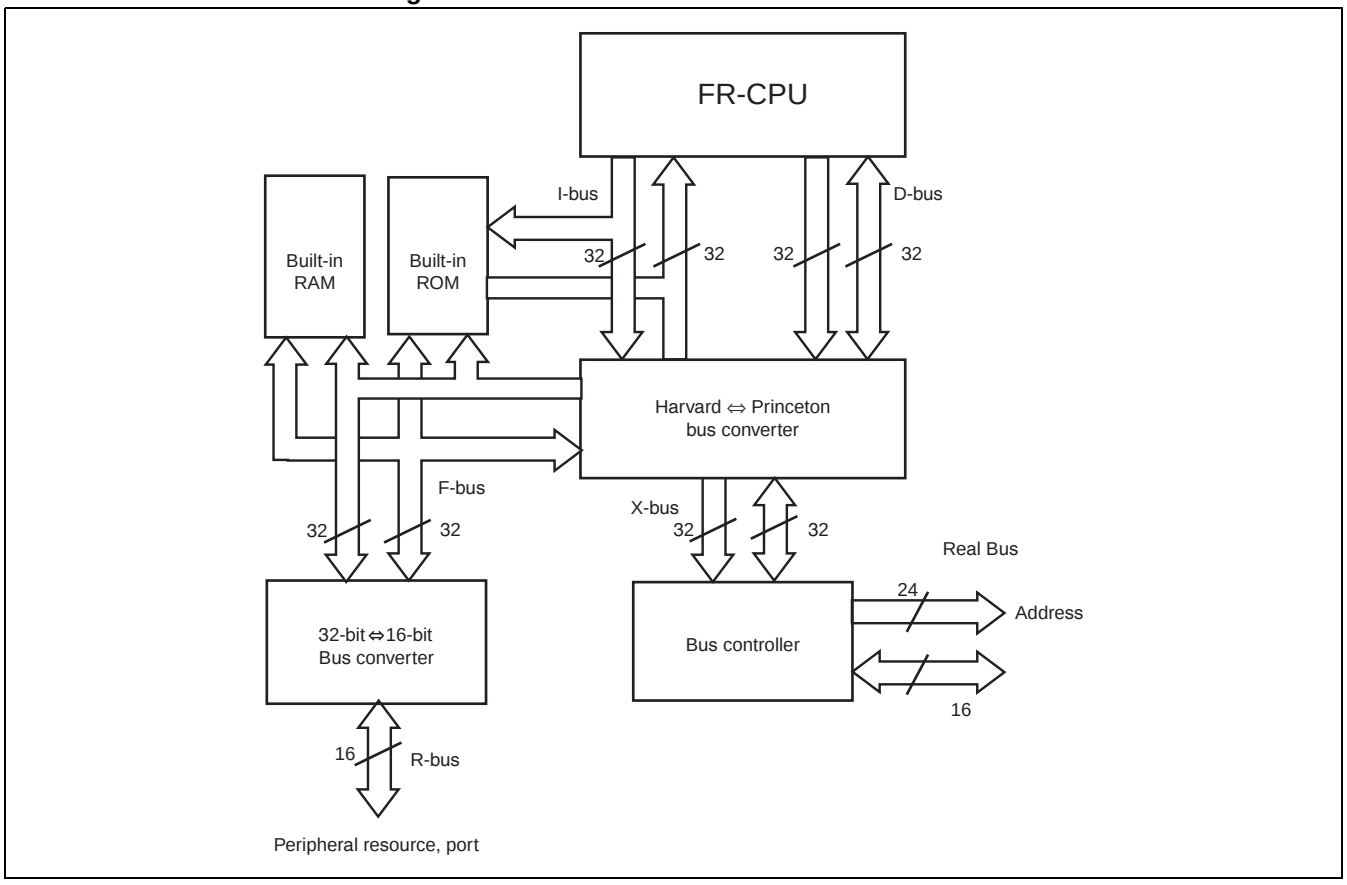
■ Structure of Internal Architecture

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

A 32-bit ←→ 16-bit bus converter is connected to the 32-bit bus (F-bus) to provide an interface between the CPU and peripherals.

A Harvard ←→ Princeton bus converter is connected to the I-bus and D-bus to provide an interface between the CPU and the bus controller.

Figure 2.2-1 Structure of Internal Architecture



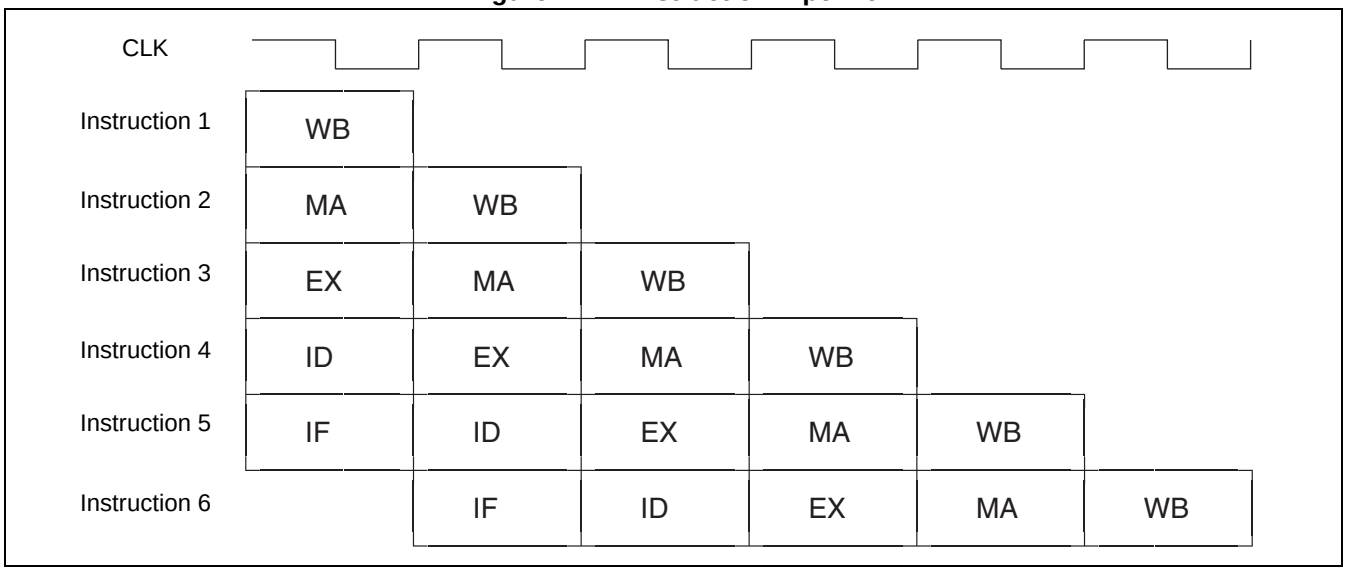
■ CPU

The 32-bit RISC architecture of the FR family has been compactly implemented in this CPU. A 5-stage instruction pipeline system is used to enable the execution of one instruction per cycle. The pipeline is composed of the following stages.

Figure 2.2-2 shows the configuration of the instruction pipeline.

- Instruction fetch (IF) : Outputs an instruction address and fetches the instruction.
- Instruction decode (ID) : Decodes the fetched instruction. It also reads from a register.
- Execution (EX) : Executes an arithmetic operation.
- Memory access (MA) : Accesses the memory for load or store operation.
- Write-back (WB) : Writes the arithmetic operation result (or loaded memory data) to a register.

Figure 2.2-2 Instruction Pipeline



Instructions are always executed in the correct order. This means that if Instruction A enters the pipeline before Instruction B, Instruction A always reaches the write-back stage before Instruction B does.

As a rule, the execution speed of instructions is based on one instruction per cycle. However, more than one cycle are required to execute instructions such as load/store instructions with memory wait, branch instructions without a delay slot, and multi-cycle instructions. Moreover, the execution speed of an instruction also decreases, when the supply of the instruction is delayed.

■ 32-bit ←→ 16-bit Bus Converter

The 32-bit ←→ 16-bit bus converter provides an interface between the F-bus (high-speed 32-bit access) and the R-bus (16-bit access) to enable data access from the CPU to the built-in peripheral circuits.

When 32-bit access is performed from the CPU to the R-bus, the bus converter converts 32-bit access into two sets of 16-bit access. Some built-in peripheral circuits have access-width restrictions.

■ Harvard ←→ Princeton Bus Converter

The Harvard ←→ Princeton bus converter adjusts instruction access from the CPU to data access to provide a seamless interface with an external bus.

The CPU is structured in the Harvard architecture in which the instruction bus exists independently from the data bus. On the other hand, the bus controller, which controls the external bus, is structured in the single-bus-based Princeton architecture. This bus converter assigns an order of priority for the instruction access from the CPU and data access to control access to the bus controller. This mechanism allows the external bus access order to be always optimized.

2.3 Instructions

The FR family supports not only a regular RISC instruction set but also logic operations, bit manipulation and direct addressing instructions, which are optimized for embedded applications. The instruction set is listed in "APPENDIX D Instruction Lists".
Each instruction is 16-bit long (some are 32 or 48-bit long) to ensure excellent efficiency in memory usage.
The instruction set can be divided into the following function groups:

- **Arithmetic Operation**
- **Load and Store**
- **Branching**
- **Arithmetic Operation and Bit Manipulation**
- **Direct Addressing**
- **Others**

■ Arithmetic Operation

There are the standard arithmetic operation instructions (addition, subtraction and comparison) and shift instructions (logic shift and arithmetic operation shift) available. For addition and subtraction, the following operations are also possible: operation with carry, used for multi-word operation; and operation with an unchanged flag value, useful for address calculation.

The provided instructions also include 32-bit-by-32-bit and 16-bit-by-16-bit multiplication instructions, 32-bit-by-32-bit step division instructions as well as immediate transfer instructions that set an immediate value in a register, and register-to-register transfer instructions.

All arithmetic operation instructions are performed using general-purpose registers and multiplication and division registers in the CPU.

■ Load and Store

Load and store instructions are used to read from and write to external memory. They are also used to read from and write to peripheral circuits (I/O) on the chip.

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

■ Branching

Branch instructions are used for branching, calling, interrupting and returning purposes. Some have a delay slot while the others do not, enabling instruction optimization for each application. The branch instructions are detailed later in the chapter.

■ Arithmetic Operation and Bit Manipulation

The logic operation instruction can be used to perform a logic operation such as AND, OR, EOR between general-purpose registers or between a general-purpose register and the memory (and I/O). The bit manipulation instruction can be used to directly manipulate the content of the memory (and I/O).

General register indirect memory addressing is supported.

■ Direct Addressing

The direct addressing instruction is used to provide access between I/O and a general-purpose register or between I/O and the memory. I/O address can be specified directly in the instruction, rather than indirectly by a register, to enable high-speed, highly efficient access. Some instructions support register-indirect memory addressing with the register increment/decrement feature.

■ Overview of Other Instructions

Other instructions are used to set a flag in the PS register, perform stack operation, and add sign/zero extension. They also include function entry/exit instructions supporting high-level language as well as register multi-load/store instructions.

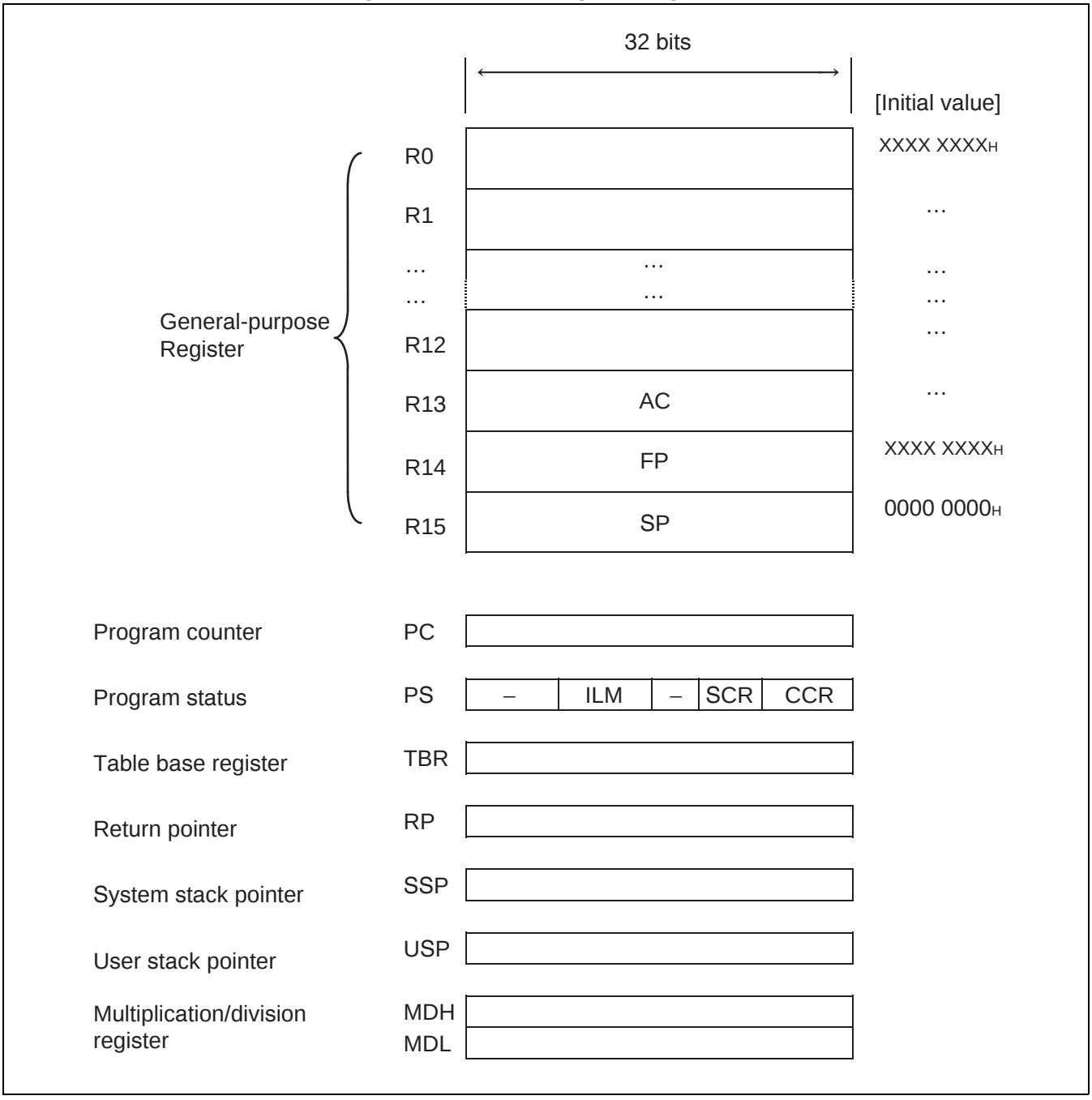
2.4 Programming Model

This section explains the programming model, general-purpose registers and dedicated registers of the FR family.

■ Basic Programming Model

Figure 2.4-1 shows the basic programming model of the FR family.

Figure 2.4-1 Basic Programming Model



MB91313A Series

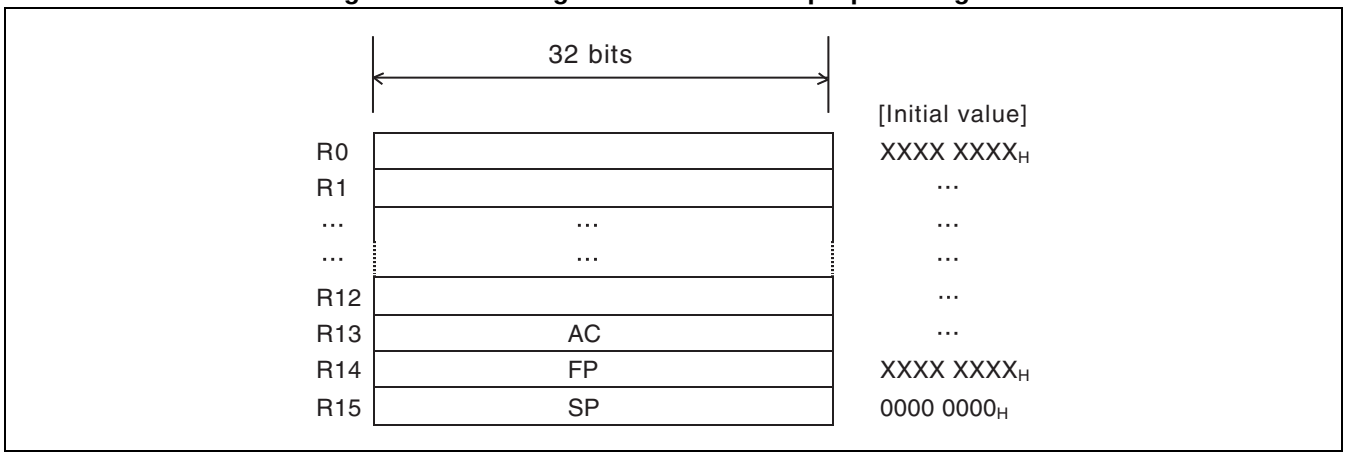
2.4.1 General-purpose Register

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

■ General-purpose Register

Figure 2.4-2 shows the configuration of a general-purpose register.

Figure 2.4-2 Configuration of General-purpose Register



Of the 16 registers, the following are intended for special applications; therefore, they have some advanced instructions.

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

The initial value at a reset is undefined for R0 to R14, but defined as "00000000_H" (SSP value) for R15.

2.4.2 Dedicated Register

A dedicated register is used for a specific purpose.
The FR family is provided with the following dedicated register:

- **PS (Program Status)**
- **CCR (Condition Code Register)**
- **SCR (System Condition code Register)**
- **ILM (Interrupt Level Mask register)**
- **PC (Program Counter)**
- **TBR (Table Base Register)**
- **RP (Return Pointer)**
- **SSP (System Stack Pointer)**
- **USP (User Stack Pointer)**
- **MDH, MDL (Multiply & Divide register)**

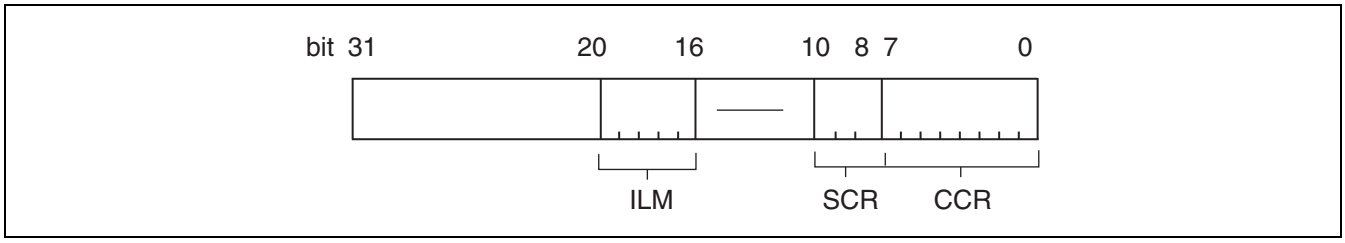
■ PS (Program Status)

PS is a register that retains the program status and divided into three parts: ILM, SCR and CCR.

All of the undefined bits are reserved. Reading them always returns "0".

Writing is invalid.

The register configuration of PS (Program Status) is shown below.



■ CCR (Condition Code Register)

The register configuration of CCR (Condition Code Register) is shown below.

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

[bit5] S: Stack flag

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

Value	Description
0	SSP is used as R15. The value is automatically set to "0" when EIT occurs. (Note that the value before the bit was cleared is saved on the stack.)
1	USP is used as R15.

•It is cleared to "0" by a reset.

•Set it to "0" when executing the RETI instruction.

[bit4] I: Interrupt enable flag

This bit enables and disables a user interrupt request.

Value	Description
0	Disables user interrupt. Cleared to "0" when INT instruction is executed. (Note that the value before the bit was cleared is saved on the stack.)
1	Enables user interrupt. Controls the masking of a user interrupt request using the value retained in ILM.

•It is cleared to "0" by a reset.

[bit3] N: Negative flag

This bit indicates the sign when the arithmetic operation result is represented as a complement integer for "2".

Value	Description
0	Indicates that the operation has resulted in a positive value.
1	Indicates that the operation has resulted in a negative value.

•The initial state at a reset is undefined.

[bit2] Z: Zero flag

This bit indicates whether the operation result is "0".

Value	Description
0	Indicates that the operation has resulted in a value other than "0".
1	Indicates that the operation has resulted in "0".

•The initial state at a reset is undefined.

[bit1] V: Overflow flag

This bit assumes the operand used in an operation as a complement integer for "2" and indicates whether an overflow has occurred due to the operation.

Value	Description
0	Indicates that no overflow has occurred due to the operation.
1	Indicates that an overflow has occurred due to the operation.

•The initial state at a reset is undefined.

[bit0] C: Carry flag

This bit indicates whether an operation has resulted in a carry or borrow from the most significant bit.

Value	Description
0	Indicates that neither a carry nor borrow has occurred.
1	Indicates that either a carry or borrow has occurred.

•The initial state at a reset is undefined.

■ SCR (System Condition code Register)

The register configuration of SCR (System Condition code Register) is shown below.

bit	10	9	8	[Initial value]
	D1	D0	T	$XX0_B$

[bit10, bit9] D1, D0: Step division flag

These bits hold the intermediate data obtained when step division is executed.

Do not modify these bits while division processing is being executed. To perform other processing while executing a step division, save and restore the value of the PS register to ensure that the step division is restarted.

- The initial state at a reset is undefined.
- To set these bits, execute the DIV0S instruction with the dividend and the divisor to be referenced.
- To forcibly clear these bits, execute the DIV0U instruction.
- Do not perform the process before EIT branching in the EIT processing routine, which simultaneously accepts DIV0S/DIV0U instruction, user interrupt and NMI.
- The D0 and D1 bits in the PS register may not indicate the correct value if the operation is stopped by break or step execution immediately before the DIV0S/DIV0U instruction. Note, however, that the correct value will be calculated after return.

[bit8] T: Step trace trap flag

This flag specifies whether the step trace trap is to be enabled.

Value	Description
0	Disables the step trace trap.
1	Enables the step trace trap. With this setting, all the user NMI and user interrupts are prohibited.

- This bit is initialized to "0" by a reset.
- The step trace trap function is used by an emulator. When used by the emulator, this function cannot be used in the user program.

■ ILM (Interrupt Level Mask Register)

The register configuration of ILM is shown below.

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

This register retains an interrupt level mask value.

The CPU accepts only interrupt requests sent to it with an interrupt level higher than the level indicated by the ILM.

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

There are restrictions on the values that can be set by the program.

- If the original value is between 16 and 31:
  The new value must be between 16 and 31. If an instruction that sets a value between 0 and 15 is executed, the specified value plus 16 is transferred.
- If the original value is between 0 and 15:
  An arbitrary value between 0 and 31 may be set.

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

■ PC (Program Counter)

The register configuration of PC (Program Counter) is shown blow.

	bit 31	0	[Initial value]
PC			$XXXXXXXX_H$

[bit31 to bit0]

These bits indicate the address of the instruction executed with the program counter.

Bit0 is set to "0", when PC is updated with the execution of an instruction. When an odd-numbered location is specified as the branch target address, bit0 is invalid; therefore, an instruction must be placed at the address with a multiple of "2".

The initial value after a reset is undefined.

■ TBR (Table Base Register)

The register configuration of TBR (Table Base Register) is shown below.

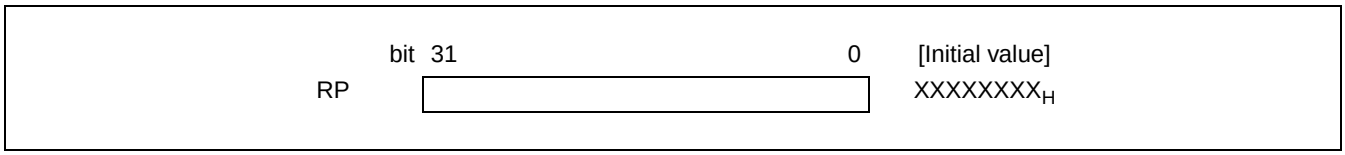
	bit 31	0	[Initial value]
TBR			$000FFC00_H$

TBR is used to retain the start address of the vector table to be used for EIT processing.

The initial value at a reset is "$000FFC00_H$".

■ RP (Return Pointer)

The register configuration of RP (Return Pointer) is shown below.



RP retains the address used for returning from the sub routine.

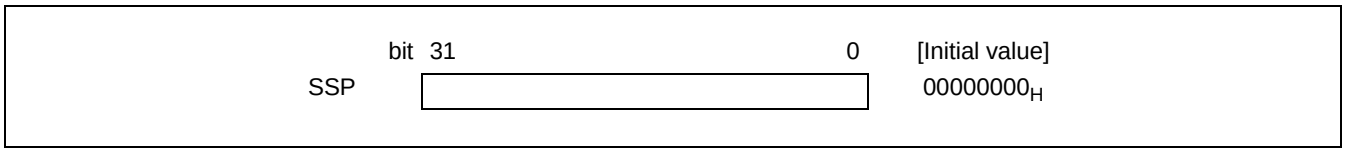
When the CALL instruction is executed, the PC value is transferred to the RP.

When the RET instruction is executed, the content of RP is transferred to PC.

The initial value after a reset is undefined.

■ SSP (System Stack Pointer)

The register configuration of SSP (System Stack Pointer) is shown below.



SSP stands for system stack pointer.

It serves as R15 when the S flag is set to "0".

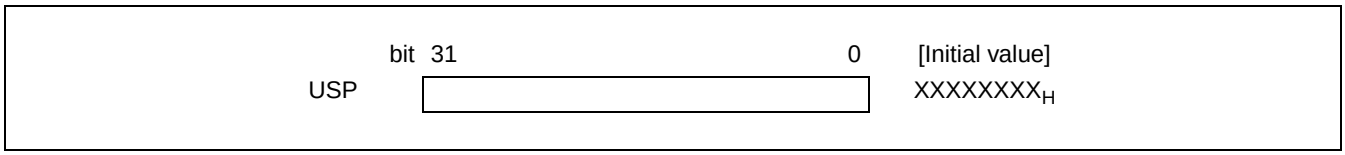
The SSP can also be specified by direct instruction.

Moreover, it can be used as a stack pointer to specify the stack that will save the PS and PC when EIT occurs.

The initial value at a reset is "00000000_H".

■ USP (User Stack Pointer)

The register configuration of USP (User Stack Pointer) is shown below.



USP stands for user stack pointer.

It serves as R15 when the S flag is set to "1".

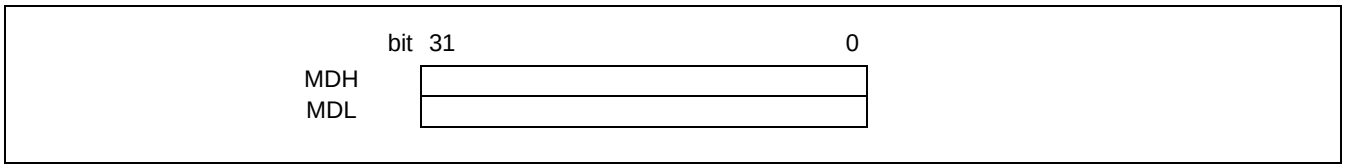
USP can also be specified by direct instruction.

The initial value after a reset is undefined.

It cannot be used with the RETI instruction.

■ MDH, MDL (Multiply & Divide Register)

The register configuration of the Multiply & Divide register is shown below.



This register is used for multiplication and division. Each is 32-bit long.

The initial value after a reset is undefined.

- For multiplication:
  The 64-bit operation result from a 32-bit-by-32-bit multiplication is stored in the multiplication/division result storage register in the following format.
  MDH : Upper 32 bits
  MDL : Lower 32 bits
  For a 16-bit-by-16-bit multiplication, the result is stored in the following format.
  MDH : Undefined
  MDL : Result of 32 bits
- For division:
  The dividend is stored in MDL when the calculation starts.
  When a division is performed by executing DIV0S/DIV0U, DIV1, DIV2, DIV3 or DIV4S instruction, the result is stored in MDL and MDH.
  MDH : Remainder
  MDL : Quotient

MB91313A Series

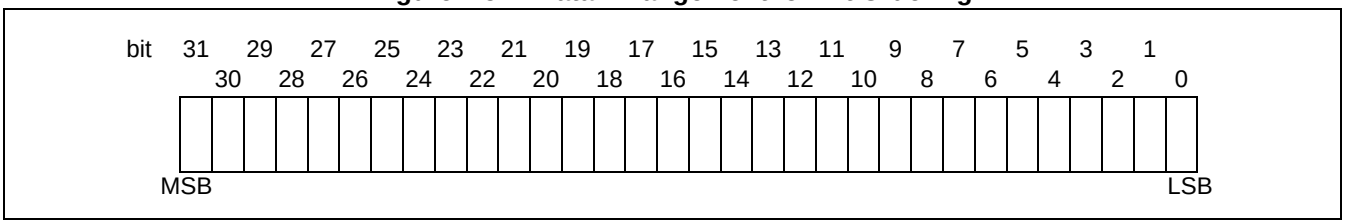
2.5 Data Structure

This section explains the data structure of the FR family.

■ Bit Ordering

The FR family has adopted a little endian system for bit ordering. Figure 2.5-1 shows the data arrangement for bit ordering.

Figure 2.5-1 Data Arrangement for Bit Ordering

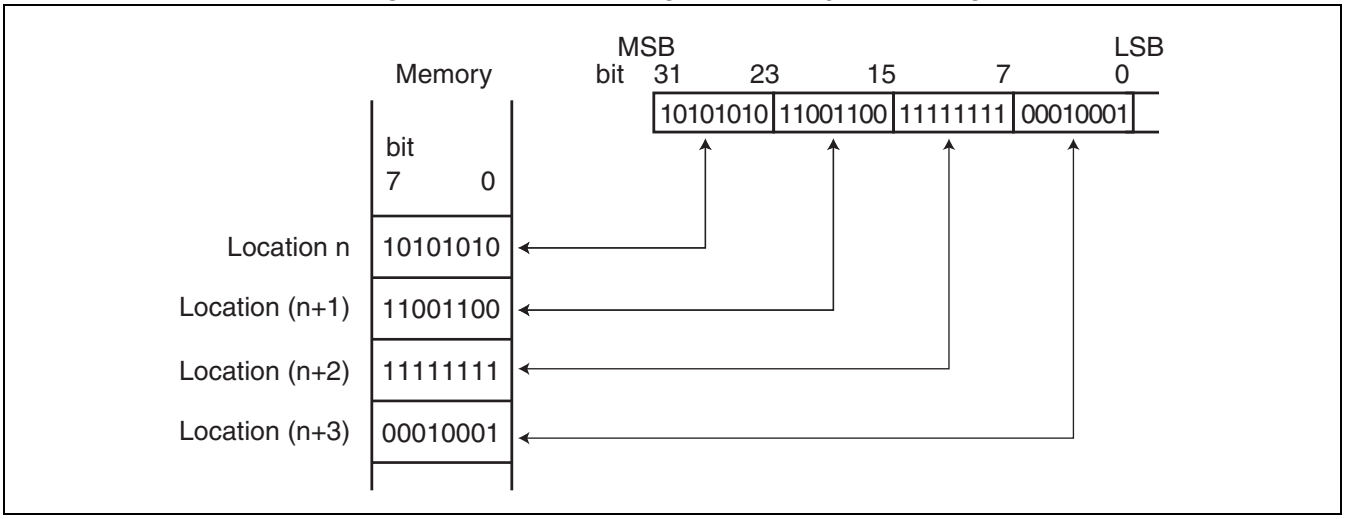


■ Byte Ordering

The FR family has adopted a big endian system for byte ordering.

Figure 2.5-2 shows the data arrangement for byte ordering.

Figure 2.5-2 Data Arrangement for Byte Ordering



■ Word Alignment

● Program access

The program for the FR family must be placed at the address with a multiple of 2.

Bit0 in the PC is set to "0" when the PC is updated with the execution of an instruction.

When an odd-numbered location is specified as the branch target address, bit0 is invalid; therefore, an instruction must be placed at the address with a multiple of "2".

There is no exception for the use of an odd-numbered address.

● Data access

In the FR family, addresses are aligned forcibly, depending on the width of the data to be accessed, as shown below.

Word access: The address has a multiple of "4". (The lowest 2 bits are forcibly set to "00".)

Half word access: The address has a multiple of "2". (The lowest bit is forcibly set to "0".)

Byte access: -

In word and half-word data access, some bits are forcibly set to "0" only for the calculation result of the effective address.

In the addressing mode @(R13, Ri), for example, the register before addition is used as it is (even when the least significant bit is "1"), and the low bits of the addition result are masked. In other words, the register before the calculation is not masked.

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

R13 00002222_H

R2 00000003_H

+)

Addition result 00002225_H

↓ Low 2 bits forcibly masked

Address pin 00002224_H

MB91313A Series

2.6 Branch Instruction

This section explains the branch instructions of the FR family.

■ Overview of Branch Instructions

In the FR family, a branch instruction can be specified to operate either with or without a delay slot.

2.6.1 Operation with a Delay Slot

This section explains the branch instruction operation with a delay slot.

■ Instruction for Operation with a Delay Slot

The instructions listed below perform operation with a delay slot.

JMP:D	@Ri	CALL:D	label12	CALL:	@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

■ Operating Explanation of Operation with a Delay Slot

In the operation with a delay slot, branching occurs after the instruction placed immediately after the branch instruction (called "delay slot") is executed before the instruction at the branch target is executed.

The dummy execution speed is 1 cycle because the instruction with a delay slot is executed before branching. However, if a valid instruction cannot be placed in the delay slot, a NOP instruction must be placed instead.

[Example]

```
;     Alignment of instructions
          ADD   R1, R2        ;
          BRA:D LABEL         ; Branch instruction
          MOV   R2, R3        ; Delay slot …… executed before branching
          …
LABEL:    ST    R3, @R4       ; Branch target
```

In case of a conditional branch instruction, the instruction placed in the delay slot is executed regardless of whether the branch condition is satisfied.

In case of delayed branch instructions, the execution order of some instructions appears to be inverted. This is however only applicable to the updating of PC. Other operations (updating/referencing a register, etc.) are executed in the order as described.

The following section provides specific details.

1) Ri, which is referenced by the JMP:D @Ri / CALL:D @Ri instruction, is not affected even when updated by the instruction in the delay slot.

[Example]

```
LDI:32    #Label,   R0
JMP:D     @R0             ; Branch to Label
LDI:8     #0,       R0    ; Branch target address not affected
…
```

2) RP, which is referenced by the RET:D instruction, is not affected even when updated by the instruction in the delay slot.

[Example]

```
RET:D                ; Branch to the preset address indicated by RP
MOV     R8,  RP      ; Return operation not affected
…
```

3) The flag referenced by the Bcc:D rel instruction is also not affected by the instruction in the delay slot.

[Example]

```
ADD     #1,  R0      ; Flag change
BC:D    Overflow     ; Branching based on the execution result of the above instruction
AND CCR #0           ; This flag update is not referenced by the above branch instruction.
…
```

4) When RP is referenced by the instruction in the delay slot of the CALL:D instruction, the updated content is read by the CALL:D instruction.

[Example]

```
CALL:D  Label        ; RP update and branching
MOV     RP,  R0      ; RP of the execution result in the CALL:D above is transferred
…
```

■ Restrictions on the Operation with a Delay Slot

● Instructions that can be placed in the delay slot

Only the instructions which meet the following conditions can be executed in the delay slot.

- 1-cycle instruction
- Not a branch instruction
- Instruction that does not affect the operation even if the order is changed

The "1-cycle instruction" refers to an instruction with "1", "a", "b", "c" or "d" indicated in the column for the number of cycles on the instruction list.

● Step trace trap

Step trace trap does not occur between the execution of a branch instruction with a delay slot and the delay slot.

● Interrupt and NMI

Neither an interrupt nor NMI can be accepted between the execution of a branch instruction with a delay slot and the delay slot.

● Undefined instruction exception

If the delay slot contains an undefined instruction, no undefined instruction exception occurs. In this case, the undefined instruction operates as a NOP instruction.

2.6.2 Operation with No Delay Slot

This section explains the operation with no delay slot specified for a branch instruction.

■ Instructions that Operate without a Delay Slot

The instructions listed below perform branching without a delay slot.

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

■ Explanation of Operation without a Delay Slot

The operation without a delay slot is executed strictly according to the order of instructions as arranged.

The immediately following instruction is never executed before branching.

[Example]

```
; Alignment of instructions
        ADD   R1, R2     ;
        BRA   LABEL      ; Branch instruction (no delay slot)
        MOV   R2, R3     ; Not executed
        …
LABEL:  ST    R3, @R4    ; Branch target
```

The number of execution cycles for a branch instruction without a delay slot is 2 cycles for branching, and 1 cycle for no branching.

As a branch instruction without a delay slot cannot contain an appropriate instruction in a delay slot, it can have higher instruction code efficiency than a branch instruction with a delay slot, which describes NOP.

High execution speed and code efficiency can be both achieved by selecting the operation with a delay slot when a valid instruction can be placed in the delay slot, and selecting the operation without a delay slot otherwise.

2.7 EIT (Exception, Interruption, and Trap)

EIT, which is the generic term for "Exception", "Interrupt", and "Trap", indicates that the current program is suspended due to an event generated when another program is being executed while the current one is still running.
The exception is an event which occurs in relation to the context under execution. Execution continues from the instruction that caused the exception.
The interruption is an event which occurs without any relation to the context under execution. The event source is hardware.
The trap is an event which occurs in relation to the context under execution. Some traps, such as system calls, are specified by the program. Execution continues from the instruction after the instruction that caused the trap.

■ Features of EIT

- Interruption supporting multiple interrupt
- Level mask function for interruption (15 levels available to the user)
- Trap instruction (INT)
- EIT for activating an emulator (hardware/software)

■ EIT Sources

The following are used as EIT sources:

- Reset
- User interrupt (internal resource and external interrupts)
- NMI
- Delayed interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap
- Coprocessor absence trap
- Coprocessor Error Trap

Note:

There are EIT-related restrictions on the delay slot of a branch instruction. For details, see "2.6 Branch Instruction".

■ Returning from EIT

RETI instruction

2.7.1 EIT Interrupt Levels

The interrupt levels range from 0 to 31, which are managed by 5 bits.

■ EIT Interrupt Levels

Table 2.7-1 shows the interrupt levels.

Table 2.7-1 Interrupt Levels

Level		Interrupt resource	Remarks
Binary	Decimal		
00000	0	(Reserved for system)	
…	…	…	
…	…	…	
00011	3	(Reserved for system)	
00100	4	INTE instruction Step trace trap	If the original value of ILM is between 16 and 31, the value of this range cannot be set in the ILM by the program.
00101	5	(Reserved for system)	
…	…	…	
…	…	…	
01110	14	(Reserved for system)	
01111	15	NMI (for user)	
10000	16	Interrupt	When ILM is set, a user interrupt is disabled.
10001	17	Interrupt	
…	…	…	
…	…	…	
11110	30	Interrupt	
11111	31	—	When ICR is set, an interrupt is disabled.

The operation is enabled at levels 16 to 31.

The interrupt level does not affect the undefined interrupt exception, coprocessor absent trap, coprocessor error trap or INT instruction. It also does not change ILM.

■ I Flag

I flag is used to enable or disable interrupts. It is provided as bit4 in CCR of the PS register.

Value	Description
0	Interrupts disabled Cleared to "0" when INT instruction is executed. (Note that the value before the bit was cleared is saved on the stack.)
1	Interrupt enabled The masking of the interrupt request is controlled by the value retained in ILM.

■ ILM

ILM is a PS register (bit20 to bit16) that retains the interrupt level mask value.

The CPU accepts only interrupt requests sent to it with an interrupt level higher than the level indicated by the ILM.

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

There are restrictions on the values that can be set by the program. If the original value is between 16 and 31, a new value may be set between 16 and 31. If an instruction that sets a value between 0 and 15 is executed, the specified value plus 16 is transferred.

If the original value is between 0 and 15, an arbitrary value between 0 and 31 may be set. ST ILM instruction is used to set an arbitrary value.

■ Level Masking for Interrupt/NMI

When NMI or interrupt request is generated, the interrupt level for the interrupt source (see Table 2.7-1) is compared with the level mask value retained in ILM. Then, if the following condition is satisfied in level strength, it will be masked and the request will not be accepted.

Interrupt level for interrupt source ≥ Level mask value

2.7.2 ICR (Interrupt Control Register)

ICR is a register located in the interrupt controller and used to set a certain level to each interrupt request. ICR is provided to support the input of various interrupt requests. ICR is mapped in the I/O space and accessed from the CPU via a bus.

■ Bit Configuration of ICR

The bit configuration of ICR is shown below.

bit	7	6	5	4	3	2	1	0	Initial value
	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---111111$_B$
	-	-	-	R	R/W	R/W	R/W	R/W	

[bit4] ICR4

ICR4 is always set to "1".

[bit3 to bit0] ICR3 to ICR0

These are lower 4 bits of the interrupt level of the corresponding interrupt source. They are readable and writable. Combined with bit4, ICR can be used to set any value between 16 and 31.

■ ICR Mapping

Table 2.7-2 lists the assignment of interrupt sources, interrupt control registers, and interrupt vectors.

Table 2.7-2 Interrupt Sources, Interrupt Control Registers and Interrupt Vectors

Interrupt Source	Interrupt Control Register		Corresponding Interrupt Vector		
	No.	Address	No.		Address
			Hexa-decimal	Decimal	
IRQ00	ICR00	00000440$_H$	10$_H$	16	TBR+3BC$_H$
IRQ01	ICR01	00000441$_H$	11$_H$	17	TBR+3B8$_H$
IRQ02	ICR02	00000442$_H$	12$_H$	18	TBR+3B4$_H$
…	…	…	…	…	…
…	…	…	…	…	…
IRQ45	ICR45	0000046D$_H$	3D$_H$	61	TBR+308$_H$
IRQ46	ICR46	0000046E$_H$	3E$_H$	62	TBR+304$_H$
IRQ47	ICR47	0000046F$_H$	3F$_H$	63	TBR+300$_H$

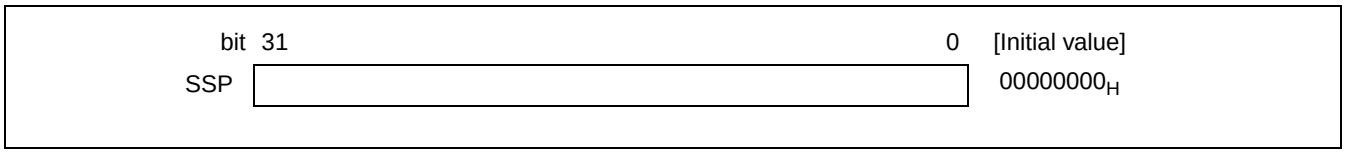
- TBR initial value: "000FFC00$_H$"
- For details, see "CHAPTER 10 INTERRUPT CONTROLLER"

2.7.3 SSP (System Stack Pointer)

SSP (System Stack Pointer) is used as the pointer that indicates the stack for saving and restoring data at the acceptance of EIT or return.

■ SSP (System Stack Pointer)

The register configuration of SSP (System Stack Pointer) is shown below.



The content is stored by data of SSP-8 during EIT processing and data of SSP-8 is stored to it during return from the EIT executed by the RETI instruction.

The initial value at a reset is "00000000_H".

SSP (System Stack Pointer) also serves as R15 (general-purpose register) when the S flag in CCR is set to "0".

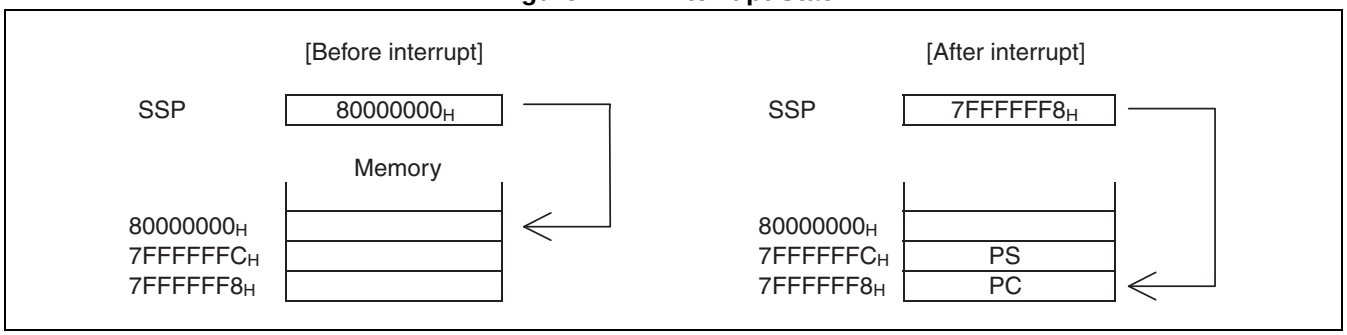
■ Interrupt Stack

This is the area indicated by SSP (System Stack Pointer), where the PC and PS values are saved and restored.

After an interrupt, the PC is stored at the address indicated by SSP (System Stack Pointer), and the PS at the address (SSP + 4).

Figure 2.7-1 shows an example of the interrupt stack.

Figure 2.7-1 Interrupt Stack

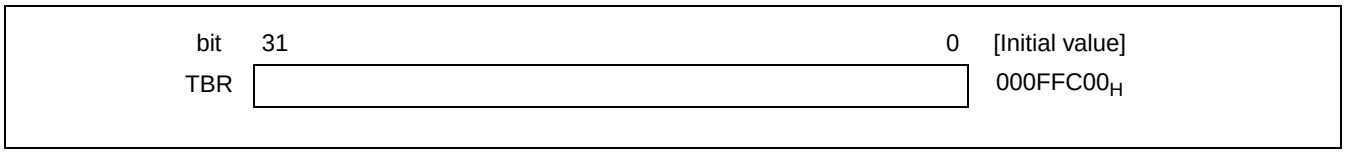


2.7.4 TBR (Table Base Register)

TBR (Table Base Register) indicates the start address of the EIT vector table.

■ TBR (Table Base Register)

The register configuration of TBR is shown below.



The vector address is calculated by adding the TBR (Table Base Register) and offset value determined for each EIT source.

The initial value at a reset is "$000FFC00_H$".

■ EIT Vector Table

The EIT vector area is the 1 KB area starting from the address indicated by TBR.

Each vector consists of 4 bytes. The relationship between the vector number and vector address is as follows.

vctadr = TBR + vctofs

= TBR + ($3FC_H$ - 4 × vct)

vctadr : Vector address

vctofs : Vector offset

vct : Vector number

The lower 2 bits of the addition result are always treated as "00".

The initial area of the vector table by a reset is the area between "$000FFC00_H$" and "$000FFFFF_H$".

Special functions are assigned to some vectors.

Table 2.7-3 shows the vector table on the architecture.

Table 2.7-3 Vector Table (1 / 4)

Interrupt Source	Interrupt No.		Interrupt Level	Offset	TBR Default Address
	Decimal	Hexa-decimal			
Reset *1	0	00	-	$3FC_H$	$000FFFFC_H$
Mode vector *1	1	01	-	$3F8_H$	$000FFFF8_H$
System-reserved	2	02	-	$3F4_H$	$000FFFF4_H$
System-reserved	3	03	-	$3F0_H$	$000FFFF0_H$
System-reserved	4	04	-	$3EC_H$	$000FFFEC_H$

*1: Even when the TBR value is modified, the fixed addresses "$000FFFFC_H$" and "$000FFFF8_H$" are used for the reset vector and mode vector respectively.

Table 2.7-3 Vector Table (2 / 4)

Interrupt Source	Interrupt No.		Interrupt Level	Offset	TBR Default Address
	Decimal	Hexa-decimal			
System-reserved	5	05	-	$3E8_H$	$000FFFE8_H$
System-reserved	6	06	-	$3E4_H$	$000FFFE4_H$
Coprocessor absence trap	7	07	-	$3E0_H$	$000FFFE0_H$
Coprocessor error trap	8	08	-	$3DC_H$	$000FFFDC_H$
INTE instruction	9	09	-	$3D8_H$	$000FFFD8_H$
System-reserved	10	0A	-	$3D4_H$	$000FFFD4_H$
System-reserved	11	0B	-	$3D0_H$	$000FFFD0_H$
Step trace trap	12	0C	-	$3CC_H$	$000FFFCC_H$
NMI request (tool)	13	0D	-	$3C8_H$	$000FFFC8_H$
Undefined instruction exception	14	0E	-	$3C4_H$	$000FFFC4_H$
NMI request	15	0F	$15(F_H)$ fixed	$3C0_H$	$000FFFC0_H$
External interrupt 0	16	10	ICR00	$3BC_H$	$000FFFBC_H$
External interrupt 1	17	11	ICR01	$3B8_H$	$000FFFB8_H$
External interrupt 2	18	12	ICR02	$3B4_H$	$000FFFB4_H$
External interrupt 3	19	13	ICR03	$3B0_H$	$000FFFB0_H$
External interrupt 4	20	14	ICR04	$3AC_H$	$000FFFAC_H$
External interrupt 5	21	15	ICR05	$3A8_H$	$000FFFA8_H$
External interrupt 6	22	16	ICR06	$3A4_H$	$000FFFA4_H$
External interrupt 7	23	17	ICR07	$3A0_H$	$000FFFA0_H$
Reload timer 0	24	18	ICR08	$39C_H$	$000FFF9C_H$
Reload timer 1	25	19	ICR09	398_H	$000FFF98_H$
Reload timer 2	26	1A	ICR10	394_H	$000FFF94_H$
Maskable interrupt source *2	27	1B	ICR11	390_H	$000FFF90_H$
Maskable interrupt source *2	28	1C	ICR12	$38C_H$	$000FFF8C_H$
Maskable interrupt source *2	29	1D	ICR13	388_H	$000FFF88_H$
Maskable interrupt source *2	30	1E	ICR14	384_H	$000FFF84_H$
Maskable interrupt source *2	31	1F	ICR15	380_H	$000FFF80_H$
Maskable interrupt source *2	32	20	ICR16	$37C_H$	$000FFF7C_H$
Maskable interrupt source *2	33	21	ICR17	378_H	$000FFF78_H$
Maskable interrupt source *2	34	22	ICR18	374_H	$000FFF74_H$
Maskable interrupt source *2	35	23	ICR19	370_H	$000FFF70_H$
Maskable interrupt source *2	36	24	ICR20	$36C_H$	$000FFF6C_H$
Maskable interrupt source *2	37	25	ICR21	368_H	$000FFF68_H$
Maskable interrupt source *2	38	26	ICR22	364_H	$000FFF64_H$
Maskable interrupt source *2	39	27	ICR23	360_H	$000FFF60_H$

*2: The maskable interrupt source is defined for each model. For the vector table used in this model, see Appendix B-1.

Table 2.7-3 Vector Table (3 / 4)

Interrupt Source	Interrupt No.		Interrupt Level	Offset	TBR Default Address
	Decimal	Hexa-decimal			
Maskable interrupt source *2	40	28	ICR24	$35C_H$	$000FFF5C_H$
Maskable interrupt source *2	41	29	ICR25	358_H	$000FFF58_H$
Maskable interrupt source *2	42	2A	ICR26	354_H	$000FFF54_H$
Maskable interrupt source *2	43	2B	ICR27	350_H	$000FFF50_H$
Maskable interrupt source *2	44	2C	ICR28	$34C_H$	$000FFF4C_H$
Maskable interrupt source *2	45	2D	ICR29	348_H	$000FFF48_H$
Maskable interrupt source *2	46	2E	ICR30	344_H	$000FFF44_H$
Overflow of time-base timer	47	2F	ICR31	340_H	$000FFF40_H$
Maskable interrupt source *2	48	30	ICR32	$33C_H$	$000FFF3C_H$
Maskable interrupt source *2	49	31	ICR33	338_H	$000FFF38_H$
Maskable interrupt source *2	50	32	ICR34	334_H	$000FFF34_H$
Maskable interrupt source *2	51	33	ICR35	330_H	$000FFF30_H$
Maskable interrupt source *2	52	34	ICR36	$32C_H$	$000FFF2C_H$
Maskable interrupt source *2	53	35	ICR37	328_H	$000FFF28_H$
Maskable interrupt source *2	54	36	ICR38	324_H	$000FFF24_H$
Maskable interrupt source *2	55	37	ICR39	320_H	$000FFF20_H$
Maskable interrupt source *2	56	38	ICR40	$31C_H$	$000FFF1C_H$
Maskable interrupt source *2	57	39	ICR41	318_H	$000FFF18_H$
Maskable interrupt source *2	58	3A	ICR42	314_H	$000FFF14_H$
Maskable interrupt source *2	59	3B	ICR43	310_H	$000FFF10_H$
Maskable interrupt source *2	60	3C	ICR44	$30C_H$	$000FFF0C_H$
Maskable interrupt source *2	61	3D	ICR45	308_H	$000FFF08_H$
Maskable interrupt source *2	62	3E	ICR46	304_H	$000FFF04_H$
Delayed interrupt source bit	63	3F	ICR47	300_H	$000FFF00_H$
Reserved for system (used in REALOS)	64	40	-	$2FC_H$	$000FFEFC_H$
Reserved for system (used in REALOS)	65	41	-	$2F8_H$	$000FFEF8_H$
System-reserved	66	42	-	$2F4_H$	$000FFEF4_H$
System-reserved	67	43	-	$2F0_H$	$000FFEF0_H$
System-reserved	68	44	-	$2EC_H$	$000FFEEC_H$
System-reserved	69	45	-	$2E8_H$	$000FFEE8_H$
System-reserved	70	46	-	$2E4_H$	$000FFEE4_H$
System-reserved	71	47	-	$2E0_H$	$000FFEE0_H$
System-reserved	72	48	-	$2DC_H$	$000FFEDC_H$
System-reserved	73	49	-	$2D8_H$	$000FFED8_H$
System-reserved	74	4A	-	$2D4_H$	$000FFED4_H$

*2: The maskable interrupt source is defined for each model. For the vector table used in this model, see Appendix B-1.

Table 2.7-3 Vector Table (4 / 4)

Interrupt Source	Interrupt No.		Interrupt Level	Offset	TBR Default Address
	Decimal	Hexa-decimal			
System-reserved	75	4B	-	$2D0_H$	$000FFED0_H$
System-reserved	76	4C	-	$2CC_H$	$000FFECC_H$
System-reserved	77	4D	-	$2C8_H$	$000FFEC8_H$
System-reserved	78	4E	-	$2C4_H$	$000FFEC4_H$
System-reserved	79	4F	-	$2C0_H$	$000FFEC0_H$
Used for INT instruction	80 to 255	50 to FF	-	$2BC_H$ to 000_H	$000FFEBC_H$ to $000FFC00_H$

2.7.5 Multiple EIT Processing

When more than one EIT source occur at the same time, the CPU selects and accepts only one source. After executing the EIT sequence, it detects another EIT source to continue the operation.
When acceptable EIT sources can no longer be detected, the CPU executes the instruction of the handler for the last accepted EIT source.
For this reason, the following 2 elements determine the handler execution sequence for EIT sources that occur at the same time.

- **Priority order for accepting EIT sources**
- **Masking condition for other sources when one is accepted**

■ Priority Levels for Accepting EIT Sources

The priority level for accepting EIT sources is the level used to select the source for executing the EIT sequence in which PS and PC are saved, PC is updated (if necessary), and other sources are masked.

The handler of the first accepted source is not necessarily executed first.

Table 2.7-4 shows the priority levels for accepting EIT sources and the masking condition for other sources.

Table 2.7-4 Priority Levels for Accepting EIT Sources and Masking Condition for Other Sources

Acceptance priority	Source	Masking of Other Sources
1	Reset	Discard other sources.
2	Undefined instruction exception	Cancel
3	INTE instruction	ILM=4 Discard other sources.
4	INT instruction	I flag=0
5	Coprocessor absence trap Coprocessor error trap	–
6	User interrupt	ILM= Level of accepted source
7	NMI (for user)	ILM=15 (This product has no NMI.)
8	NMI (for emulator)	ILM=4
9	Step trace trap	ILM=4

Table 2.7-5 shows the execution sequence of the handlers of EIT sources that occur at the same time, in conjunction with the masking process for the other sources after one is accepted.

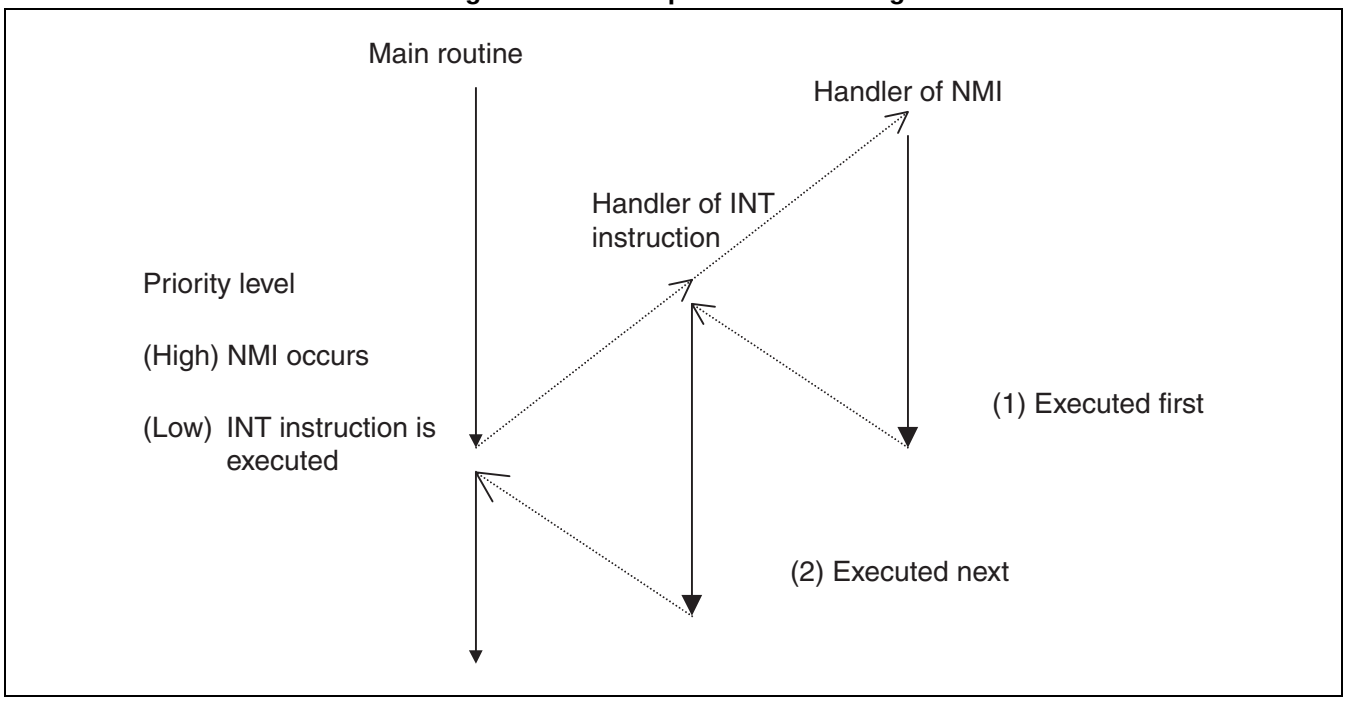
Table 2.7-5 Execution Sequence of EIT Handlers

Handler execution order	Source
1	Reset*
2	Undefined instruction exception
3	INTE instruction*
4	Step trace trap
5	NMI (for user. Note that this product has no NMI.)
6	INT instruction
7	User interrupt
8	Coprocessor absence trap, Coprocessor error trap

*: Other sources are discarded.

Figure 2.7-2 shows an example of the multiple EIT processing.

Figure 2.7-2 Multiple EIT Processing



2.7.6 Operations of EIT

This section explains various types of operations in the FR family.
In the following explanation, "PC" for the origin of transfer refers to the address of the instruction that detected each EIT source. Depending on the instruction which has detected EIT, the "address of the next instruction" varies as follows.

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

■ Operation of User Interrupt and NMI

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

[Determining whether or not to accept interrupt request]

1) The interrupt levels of requests generated at the same time are compared, and the request with the highest level (the smallest numeric value) is selected and retained.
   For the level used for the comparison, the value held in the corresponding ICR is used for a maskable interrupt and the predefined constant is used for the NMI.
2) If multiple requests holding the same level are generated, the request with the smallest interrupt number is selected.
3) When the interrupt level is the same as the level mask value or greater, the interrupt request is masked and is not accepted.
   When the interrupt level is lower than the level mask value, go to 4).
4) If the I flag is "0" when the selected interrupt request is intended for a maskable interrupt, the interrupt request is masked and is not accepted. If the I flag is "1", go to 5).
   If the selected interrupt request is intended for NMI, go to 5) regardless of the value of the I flag.
5) When the above conditions are met, the interrupt request is accepted at a boundary between instruction processing sessions.

If a user interrupt/NMI request is accepted upon the detection of an EIT request, the CPU operates as described below, using the interrupt number corresponding to the accepted interrupt request.

Note: What is contained in parentheses represents a register-specified address.

[Operation]

1) SSP – 4 → SSP
2) PS → (SSP)
3) SSP – 4 → SSP
4) Address of next instruction → (SSP)
5) Interrupt level of accepted request → ILM
6) "0" → S flag
7) (TBR + Vector offset of accepted interrupt request) → PC

A new EIT is detected before executing the first instruction of the handler upon the completion of the interrupt sequence. If an acceptable EIT has been generated at this point, the CPU moves to the EIT processing sequence.

If the OR CCR, ST ILM, MOV Ri or PS instruction is executed to enable an interrupt while a user interrupt or NMI source is being generated, the above instruction may be executed twice, before and after the interrupt handler. Note however that this does not affect the operation as it is only the same value that is set twice.

Do not perform the process before EIT branching in the EIT processing routine.

■ Operation of INT Instruction

INT #u8:

Branches 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 + $3FC_H$-4 × u8) → PC

■ Operation of INTE Instruction

INTE:

Branches to the interrupt handler for the vector #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 + $3D8_H$) → PC

Do not use the INTE instruction during the processing routine for the INTE instruction and step trace trap. Also, EIT is not generated by INTE during the step execution.

■ Operation of Step Trace Trap

A trap occurs on the execution of each instruction and execution breaks, if the T flag in SCR of PS is set to enable the step trace function.

[Conditions for Step Trace Trap Detection]

1) T flag =1
2) Not a delayed branch instruction
3) During the execution of processing routine other than for INTE instruction and step trace trap

When the above conditions are met, execution breaks at a boundary of instruction operation.

[Operation]

1) SSP – 4 → SSP
2) PS → (SSP)
3) SSP – 4 → SSP
4) Address of next instruction → (SSP)
5) "00100_B" → ILM
6) "0" → S flag
7) (TBR + $3CC_H$) → PC

When the T flag is set to enable step trace trap, user NMI and user interrupt are disabled. Moreover, EIT is not generated by the INTE instruction.

In the FR family, a trap is generated from the instruction following the instruction that set the T flag.

■ Operation of Undefined Instruction Exception

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

[Conditions for detecting undefined instruction exception]

1) Undefined instruction detected during instruction decoding
2) Placed outside a delay slot (Not immediately after a delayed branch instruction)
   When the above conditions are met, an undefined instruction exception occurs and execution breaks.

[Operation]

1) SSP – 4 → SSP
2) PS → (SSP)
3) SSP – 4 → SSP
4) PC → (SSP)
5) "0" → S flag
6) (TBR + $3C4_H$) → PC

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

■ Coprocessor Absence Trap

A coprocessor absent trap occurs, if a coprocessor instruction is executed to use an unmounted coprocessor.

[Operation]

1) SSP – 4 → SSP
2) PS → (SSP)
3) SSP – 4 → SSP
4) Address of next instruction → (SSP)
5) "0" → S flag
6) (TBR + $3E0_H$) → PC

■ Coprocessor Error Trap

A coprocessor error trap occurs, if an error occurs while using a coprocessor and then a coprocessor instruction is executed to operate that coprocessor.

[Operation]

1) SSP – 4 → SSP
2) PS → (SSP)
3) SSP – 4 → SSP
4) Address of next instruction → (SSP)
5) "0" → S flag
6) (TBR + $3DC_H$) → PC

■ Operation of RETI Instruction

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

[Operation]

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

The RETI instruction must be executed when the S flag is "0".

2.8 Reset (Device Initialization)

This section explains the reset operation, or initialization, of this model.

■ Overview of Reset (Device Initialization)

If a reset source occurs, the device stops all the programs and hardware operations, and initializes the state. This state is called the reset state.

When a reset source no longer exists, the device starts programs and hardware operations from their initial state. The series of operations from the reset state to the start of operations is called the reset sequence.

2.8.1 Reset Level

The reset operations of the FR family are classified into two levels, each of which has different reset sources and initialization operations.
This section explains these reset levels.

■ Setting Initialization Reset (INIT)

The highest-level reset, which initializes all settings, is called a settings initialization reset (INIT).

The settings initialization reset (INIT) mainly initializes the following items:

[Items initialized in a settings initialization reset (INIT)]

- •Device operation mode (bus mode and external bus width settings)
- •All internal clock settings (clock source selection, PLL control, and division ratio setting)
- •All settings on external bus CS0 area
- •Any other settings for pin condition
- •All the items initialized by operation initialization reset (RST)

For more information, see the description of each of these functions.

After power-on, be sure to apply the settings initialization reset (INIT) at the INITX pin.

■ Operation Initialization Reset (RST)

A normal-level reset that initializes the operation of a program is called an operation initialization reset (RST).

During a settings initialization reset (INIT), an operation initialization reset (RST) also occurs at the same time.

The operation initialization reset (RST) mainly initializes the following items:

[Items initialized by an operation initialization reset (RST)]

- •Program operation
- •CPU and internal buses
- •Register settings of peripheral circuits
- •I/O port settings
- •All settings on external bus CS0 area

For more information, see the description of each of these functions.

2.8.2 Reset Sources

**This section explains the reset sources and the reset levels in this model.
To determine reset sources that have occurred in the past, read the RSRR (reset source register). For more information about registers and flags described in this section, see Section "2.9.5 Block Diagram of Clock Generation Control Block" and "2.9.6 Registers of Clock Generation Control Block".**

■ INITX Pin Input (Settings Initialization Reset Pin)

The INITX pin, which is an external pin, is used as the settings initialization reset pin.

A settings initialization reset (INIT) request is generated while the Low level is being input to this pin.

A settings initialization reset (INIT) request is cleared by inputting the High level to this pin.

If a settings initialization reset (INIT) is generated in response to a request from this pin, INIT (bit15) of RSRR (reset source register) is set. Because a settings initialization reset (INIT) in response to a request from this pin has the highest interrupt level among all reset sources, it has precedence over any other input, operation, or state.

After power-on, be sure to apply the settings initialization reset (INIT) at the INITX pin. Immediately after turning the power on, in addition, hold the "L" level input to the INITX pin during the stabilization wait time required for the oscillation circuit to reserve the regulator stabilization wait time. (INIT at the INITX pin initializes the oscillation stabilization wait time to the minimum value.)

- Reset source : "L" level input to external INITX pin
- Source of clearing : "H" level input to external INITX pin
- Generation level : Setting initialization reset (INIT)
- Corresponding flag: bit15:INIT

■ STCR:SRST Bit Writing (Software Reset)

If "0" is written to SRST (bit4) of STCR (standby control register), a software reset request occurs.

A software reset request is an operation initialization reset (RST) request.

When the request is accepted and a operation initialization reset (RST) is generated, the software reset request is cleared.

If an operation initialization reset (RST) is generated due to a software reset request, the SRST (bit11) in RSRR (reset source register) is set.

An operation initialization reset (RST) is generated due to a software reset request only after all bus access has stopped and if SYNCR (bit9) of TBCR (time base counter control register) has been set (synchronization reset mode).

Thus, depending on the bus usage status, a long time is required before an operation initialization reset (RST) occurs.

Please refer to the limitations of the bit9:SYNCR bit of TBCR (time-base counter control register) for the use of reset of the software of the synchronous mode.

Note:

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

- Reset source:
  Writing "0" to SRST (bit4) of STCR (standby control register)
- Source of clearing:
  Generation of an operation initialization reset (RST)
- Generation level:
  Operation initialization reset (RST)
- Corresponding flag:
  bit11:SRST

■ Watchdog Reset

Writing to the RSRR (watchdog timer control register) starts the watchdog timer. Unless "$A5_H$" / "$5A_H$" is written to the WPR (watchdog reset postpone register) within the cycle specified in WT1 (bit9) and WT0 (bit8) in RSRR, a watchdog reset request occurs.

A watchdog reset request is a settings initialization reset (INIT) request. If, after the request is accepted, a settings initialization reset (INIT) occurs or an operation initialization reset (RST) occurs, the watchdog reset request is cleared.

If a settings initialization reset (INIT) is generated due to a watchdog reset request, WDOG (bit13) in RSRR (reset source register) is set.

Note that, if a settings initialization reset (INIT) is generated due to a watchdog reset request, the oscillation stabilization wait time is not initialized.

- Reset source:
  Setting cycle of the watchdog timer elapses
- Source of clearing:
  Generation of a settings initialization reset (INIT) or an operation initialization reset (RST)
- Generation level:
  Setting initialization reset (INIT)
- Corresponding flag:
  bit13:WDOG

2.8.3 Reset Sequence

When a reset source no longer exists, the device starts to execute the reset sequence. A reset sequence has different operations depending on the reset level. This section explains the operations of the reset sequence for different reset levels.

■ Setting Initialization Reset (INIT) Clear Sequence

If a settings initialization reset (INIT) request is cleared, the following operations are performed one step at a time for the device.

1) Clear the settings initialization reset (INIT) and enter the oscillation stabilization wait state.
2) For the oscillation stabilization wait time (set with OS1 (bit3), OS0 (bit2) in STCR), maintain the operation initialization reset (RST) state and stop the internal clock.
3) In the operation initialization reset (RST) state, start internal clock operation.
4) Clear the operation initialization reset (RST) and enter the normal operating state.
5) Read the mode vector from address $000FFFF8_H$.
6) Write the mode vector to the MODR (mode register) at address $000007FD_H$.
7) Read the reset vector from address $000FFFFC_H$.
8) Write the reset vector to the PC (program counter).
9) The program starts execution from the address loaded in the PC (program counter).

■ Operation Initialization Reset (RST) Clear Sequence

If an operation initialization reset (RST) request is cleared, the following operations are performed one step at a time for the device.

1) Clear the operation initialization reset (RST) and enter the normal operating state.
2) Read the mode vector from address $000FFFF8_H$.
3) Write the mode vector to the MODR (mode register) at address $000007FD_H$.
4) Read the reset vector from address $000FFFFC_H$.
5) Write the reset vector to the PC (program counter).
6) The program starts execution from the address loaded in the PC (program counter).

2.8.4 Oscillation Stabilization Wait Time

If a device returns from the state in which the original oscillation was or may have been stopped, the device automatically enters the oscillation stabilization wait state.
This function prevents the use of oscillator output after starting before oscillation has stabilized.
For the oscillation stabilization wait time, neither an internal nor an external clock is supplied; only the built-in time base counter runs until the stabilization wait time set in the STCR (standby control register) has elapsed.
This section explains the oscillation stabilization wait operation.

■ Sources of an Oscillation Stabilization Wait

The following lists sources of an oscillation stabilization wait.

● Clearing of a settings initialization reset (INIT)

The device enters the oscillation stabilization wait state if a settings initialization reset (INIT) is cleared for a variety of reasons.

When the oscillation stabilization wait time has elapsed, the device enters the operation initialization reset (RST) state.

● Returning from stop mode

The device enters the oscillation stabilization wait state immediately after stop mode is cleared. However, if it is cleared by a settings initialization reset (INIT) request, the device enters the settings initialization reset (INIT) state. Then, after the settings initialization reset (INIT) is cleared, the device enters the oscillation stabilization wait state.

When the oscillation stabilization wait time has elapsed, the device enters the state corresponding to the source that cleared stop mode.

- Return due to input of a valid external interrupt request (including NMI) and generation of a watch timer/a main oscillation stabilization wait time interrupt:
  The device enters the normal operating state.
- Return due to a settings initialization reset (INIT) request:
  The device enters the operation initialization reset (RST) state.

● Returning from an abnormal state when PLL is selected

If, while the device is operating with PLL as the source clock, an abnormal condition* occurs in PLL control, the device automatically enters an oscillation stabilization wait state to assure the PLL lock time.

When the oscillation stabilization wait time has elapsed, the device enters the normal operating state.

*: The multiplication rate is changed while PLL is working, or an incorrect bit such as a bit equivalent to PLL operation enable bit is generated.

● Generating a watchdog reset when the main oscillation stops in sub clock

If, while the device is operating with sub clock as the source clock, a watchdog reset is generated when the main oscillation is stopped by OSCDS1 (bit8) of OSCCR (oscillation control register), the device automatically enters an oscillation stabilization wait state immediately after a reset (INIT) is cleared.

When the oscillation stabilization wait time has elapsed, the device enters the operation initialization reset (RST) state.

Note that an oscillation stabilization wait is not generated when the OSCDS1 bit of the sub clock is "0" and the device is in the main clock mode.

■ Selecting an Oscillation Stabilization Wait Time

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

If a source for an oscillation stabilization wait occurs and the device enters the oscillation stabilization wait state, the built-in time base counter is initialized and then it starts to measure the oscillation stabilization wait time.

Using the OS1 (bit3) and OS0 (bit2) of STCR (standby control register), select and set one of the four types of oscillation stabilization wait time.

Once selected, a setting is initialized only if a settings initialization reset (INIT) is generated due to the external INITX pin. The oscillation stabilization wait time that has been set before a reset is maintained if a settings initialization reset (INIT) is generated or an operation initialization reset (RST) is generated due to a watchdog reset.

2.8.5 Reset Operation Modes

Two modes for an operation initialization reset (RST) are provided: normal (asynchronous) reset mode and synchronous reset mode. The operation initialization reset mode is selected with SYNCR (bit9) of TBCR (time base counter control register). This setting is initialized only by a settings initialization reset (INIT).
A settings initialization reset (INIT) always results in an asynchronous reset.

■ Normal Reset Operation

Normal reset operation refers to entering the operation initialization reset (RST) state immediately after an operation initialization reset (RST) request occurs.

If a reset (RST) request is accepted, the device immediately enters the reset (RST) state regardless of the operating state of the internal bus.

The result of bus access performed at the time of transition to each status is not guaranteed. However, these requests can certainly be accepted.

If the SYNCR (bit9) of TBCR (time base counter control register) is set to "0", normal reset mode is selected.

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

■ Synchronous Reset Operation

Synchronous reset operation refers to entering the operation initialization reset (RST) state after all the bus accesses have stopped when an operation initialization reset (RST) request is generated.

If a reset (RST) request is accepted, the device does not enter the reset (RST) state while internal bus access is in progress.

If the above request is accepted, a sleep request is issued to the internal buses. If all the buses stop and enter the sleep state, the device enters the operation initialization reset (RST) state.

The result of all bus accesses is guaranteed because all the bus accesses are stopped at the time of transition to each state. If bus access does not stop for some reason, no requests can be accepted while the bus access is in progress. (Even in this case, the settings initialization reset (INIT) is immediately valid.)

Bus access may not stop in the following cases:

- A RDY (ready request) continues to be input to the external extended bus interface and bus wait is valid. (In the following cases, the device eventually enters another state but only after a long time.)

References:

- Please refer to the limitations of the bit9:SYNCR bit of TBCR (time-base counter control register) for the use of reset of the software of the synchronous mode.
- The DMA controller, which stops transfer when a request is accepted, does not delay transition to another state.
- If the SYNCR (bit9) of TBCR (time base counter control register) is set to "1", synchronous reset mode is selected.
- For using software reset on the synchronous mode, see the limitations of the bit9:SYNCR bit of TBCR (time-base timer counter control register).

The initial value returns to normal reset mode after a settings initialization reset (INIT).

MB91313A Series

2.9 Clock Generation Control

This section explains the clock generation control.

■ Generating Internal Operating Clock

The internal operating clock is generated as follows:

- Selecting the source clock:
  The clock supply source is selected.
- Generating the base clock:
  The base clock is generated by dividing the source clock by 2 or using PLL oscillation.
- Generating each internal clock:
  The base clock is divided to generate 4 types of operating clocks to be supplied to each block.

The following section explains how to generate and control each clock.

For detail of the registers and flags in the following explanation, See Section "2.9.5 Block Diagram of Clock Generation Control Block" and "2.9.6 Registers of Clock Generation Control Block".

■ Selecting the Source Clock

This section explains how the source clock is selected.

The source clock is the source oscillation generated in the built-in oscillation circuit by connecting an oscillator to the X0/X1 and X0A/X1A external oscillator pin inputs.

All clock sources including the external bus clock are supplied from within this model.

The external oscillator pins and built-in oscillation circuit can use 2 types of clocks (main clock and sub clock) and also switch between them during operation at any time.

- Main clock : Generated from the X0 and X1 pin inputs and intended for use as the high-speed clock.
- Sub clock : Generated from the X0A and X1A pin inputs and intended for use as the low-speed clock.

The main and sub clocks are multiplied by using the independently controllable built-in main PLL.

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

- Main clock divided by 2
- Main clock multiplied using the main PLL
- Sub clock as it is

ϕ is the base clock that is generated from the source clock divided by two or by using PLL oscillation. Therefore, the system base clock is a clock generated in the above-mentioned internal base clock generation.

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

2.9.1 PLL Control

The PLL oscillation circuit for the main clock can be controlled by enabling or disabling its operation (oscillation) and setting the multiplication rate.
Each control operation is performed by setting the clock source control register (CLKR).
This section explains the control operations.

■ Enabling PLL Operation

PLL1EN (bit10) in CLKR (clock source control register) is used to enable/disable the oscillation operation of the main PLL.

PLL2EN (bit11) in CLKR (clock source control register) is used to enable/disable the oscillation operation of the sub clock.

Both the PLL1EN and PLL2EN bits are initialized to "0" after a setting initialization reset (INIT) and the PLL oscillation operation is stopped. While it is stopped, the PLL output cannot be selected as the source clock.

Once program operation has started, set the multiplication rate for the PLL to be used as the clock source and enable its operation, and then wait for the PLL lock wait time to elapse before switching the source clock. In this case, it is recommended to use the time-base timer interrupt for the PLL lock wait time.

The PLL cannot be halted while the PLL output is selected as the source clock. (Writing to the register is ignored.) When you wish to stop the PLL in such case as changing to stop mode, select the main clock divided by 2 as the source clock before halting the PLL.

Note that if OSCD1 (bit0) and OSCD2 (bit1) in STCR (standby control register) are set so that oscillation is stopped during stop mode, the corresponding PLL is automatically stopped when moving to stop mode. Therefore, it is not necessary to set the bits again to stop the operation. Afterwards, when returning from the stop mode, PLL automatically begins the oscillation operation. The PLL does not stop automatically if the oscillation is set to continue during stop mode. In this case, stop the operation before changing to stop mode if necessary.

■ PLL Multiplication Rate

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

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

[Setting PLL multiplication rate]

When changing the PLL multiplication rate from its initial value, change it before or at the same time as enabling the PLL operation after the program operation starts. Then wait for the PLL lock wait time to elapse before switching the source clock. In this case, it is recommended to use the time-base timer interrupt for the PLL lock wait time.

If you wish to change the PLL multiplication rate during operation, first change the source clock to any clock other than the corresponding PLL. Then wait for the PLL lock wait time to elapse before switching the source clock, as described above.

The PLL multiplication rate setting can be changed while the PLL is in use. In this case, however, it automatically enters the oscillation stabilization wait state after the multiplication rate setting is rewritten, and the program operation is stopped until the set oscillation stabilization wait time elapses. The program operation does not stop when the clock source is switched to a clock other than the PLL.

2.9.2 Oscillation Stabilization Wait and PLL Lock Wait Time

If the operation of the clock selected as the source clock is not stable, an oscillation stabilization wait time is required. (See "2.8.4 Oscillation Stabilization Wait Time".) After the PLL starts operating, a wait time is required until the PLL locks in order to allow the output to stabilize at the specified frequency.
This section explains the wait time used in various situations.

■ Wait Time after Power-up

After power-up, it is necessary to input "L" level to the INITX pin input (setting initialization reset pin). In this state, as none of the PLL's are allowed to operate, it is not required to consider a lock wait time.

■ Wait Time after Setting Initialization

When a setting initialization reset (INIT) is released, the device goes to the oscillation stabilization wait state. The set oscillation stabilization wait time is internally generated.

In this state, as none of the PLL's are allowed to operate, it is not required to consider a lock wait time.

■ Wait Time after Enabling PLL Operation

If you enable the PLL in the stop state to operate after the program operation starts, the output of that PLL cannot be used until the lock wait time elapses.

If the corresponding PLL is not selected as the source clock, the program can be executed even during the lock wait time.

It is recommended to use the time-base timer interrupt for the PLL lock wait time.

■ Wait Time after Changing PLL Multiplication Rate

Even if you change the multiplication rate setting of the currently operating PLL after the program operation starts, the output of that PLL must not be used until the lock wait time elapses.

If the corresponding PLL is not selected as the source clock, the program can be executed even during the lock wait time.

The time-base timer interrupt can be used for the PLL lock wait time.

■ Wait Time after Returning from Stop Mode

After the program operation starts, the oscillation stabilization wait time set by the program is generated internally after the device moves to stop mode and then returns from that mode.

If the device is set to halt the oscillation circuit for the clock selected as the source clock during stop mode, the longer of the oscillation stabilization wait time for the oscillation circuit and the lock wait time for the PLL in use is required as the wait time. Therefore, set the oscillation stabilization wait time before changing to stop mode.

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

■ Wait Time after Switching from the Sub Clock to the Main Clock

When using the PLL after switching from the sub clock to the main clock, the output of that PLL must not be used regardless of the value of PLL1EN (bit2) in CLKR (clock source register), until the lock wait time elapses.

If the corresponding PLL is not selected as the source clock, the program can be executed even during the lock wait time.

It is recommended to use the time-base timer interrupt for the PLL lock wait time.

2.9.3 Clock Distribution

The operating clock for each function is generated based on the base clock generated from the source clock.
There are 3 different internal operating clocks in total, and each clock can set its own division ratio, independently from the other clocks.

■ CPU Clock (CLKB)

This clock is used for the CPU, internal memory and internal bus.

The circuits which use this clock are listed below:

- CPU
- Built-in RAM and built-in ROM
- Bit Search Module
- I-bus, D-bus, X-bus, F-bus
- DMA controller
- DSU

The maximum operable frequency is 33 MHz. Therefore, do not set any frequency combination of the multiplication rate and division ratio that will exceed this frequency.

■ Peripheral Clock (CLKP)

This clock is used for peripheral circuits and peripheral bus.

The circuits which use this clock are listed below:

- Peripheral bus
- Clock control block (bus interface component only)
- Interrupt controller
- Peripheral I/O port
- I/O port bus
- External interrupt input
- UART
- 16-bit timer
- A/D converter
- Free-run timer
- Reload timer
- Up/down counter
- Input capture
- Output compare
- I^2C interface
- PPG

The maximum operable frequency is 33 MHz. Therefore, do not set any frequency combination of the multiplication rate and division ratio that will exceed this frequency.

■ External Bus Clock (CLKT)

This clock is used for external extended bus interfaces.

The circuits which use this clock are listed below:

- External extended bus interface
- External CLK output

The maximum operable frequency is 16.5 MHz. Therefore, do not set any frequency combination of the multiplication rate and division ratio that will exceed this frequency.

Note:

The processing performance of CPU is influenced from the setting of flash memory wait register (FLWC). Adjust the setting of this register to the best value and use it. See Section "18.2.2 Wait Register (FLWC)".

2.9.4 Clock Division

Each internal operating clock can independently set its own division ratio from the base clock from the other clocks. This function allows the most suitable operating frequency to be provided to each circuit.

■ Setting Division Ratio

The division ratio is set by DIVR0 (basic clock division setting register 0) and DIVR1 (basic clock division setting register 1).

Each register contains 4 setting bits which correspond to each of the operating clocks, and the value "register setting + 1" is used as the division ratio for the base clock of that particular clock. Even when the division ratio is set to an odd number, Duty is always 50%.

If the setting is modified, the modified division ratio becomes valid from the rising edge of the next clock signal.

■ Initializing the Division Ratio Setting

Even when an operation initialization reset (RST) occurs, the division ratio setting is not initialized and the setting before the occurrence of the reset is maintained. The setting is initialized only when a setting initialization reset (INIT) occurs. In the initial state, the division ratio is "1" for all except the peripheral clock (CLKP). Therefore, make sure to set the division ratio before changing the source clock to a faster one.

Note:

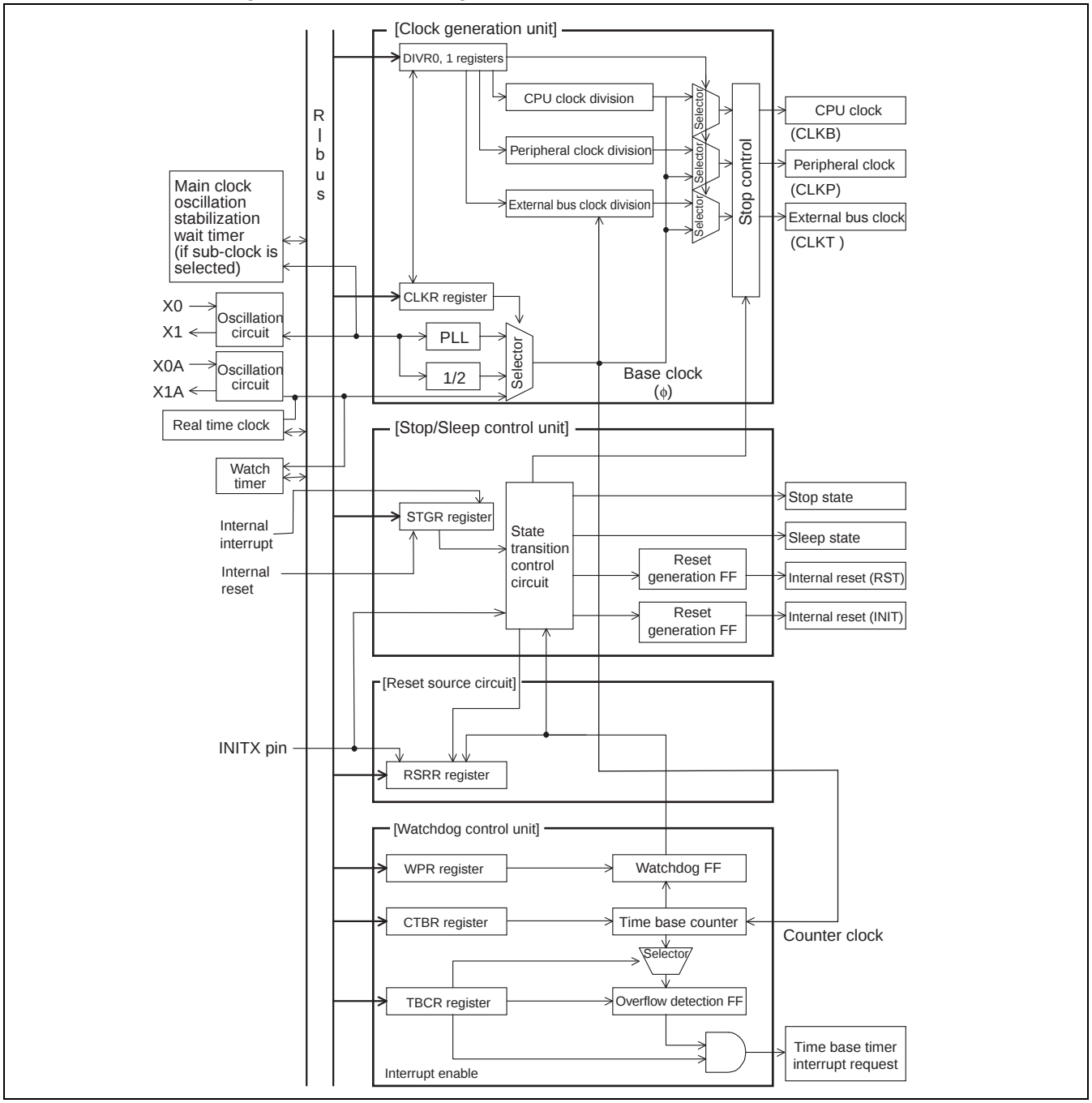
The maximum operable frequency is defined for each clock. Operation is not guaranteed if a frequency, in combination with the source clock selection, PLL multiplication rate setting and division ratio setting, is set to exceed the maximum frequency. In particular, take care to follow the correct order in conjunction with modifying the source clock selection setting.

2.9.5 Block Diagram of Clock Generation Control Block

Figure 2.9-1 shows a block diagram of the clock generation control block.
For details of the registers shown in the diagram, see "2.9.6 Registers of Clock Generation Control Block".

■ Block Diagram of Clock Generation Control Block

Figure 2.9-1 Block Diagram of Clock Generation Control Block



2.9.6 Registers of Clock Generation Control Block

This section explains the registers in the clock generation control block.

■ RSRR: Reset Source Register/Watchdog Timer Control Register

The register configuration of the reset source register and watchdog timer control register is shown below.

RSRR	bit	15	14	13	12	11	10	9	8
Address: 000480_H		INIT	Reserved	WDOG	Reserved	SRST	Reserved	WT1	WT0
		(R)	(R)	(R)	(R)	(R)	(R)	(R/W)	(R/W)
Initial value (INITX pin)		1	0	0	0	0	0	0	0
Initial value (INIT)		*	*	*	X	X	*	0	0
Initial value (RST)		X	X	X	X	*	X	0	0

*: Initialized by a source.
R/W: Readable/writable
R: Read only
X: Undefined value

RSRR retains the source of the most recently generated reset, sets the watchdog timer cycle and controls its activation.

When this register is read, the retained reset source is cleared after read. If more than one reset occur before the register is read, reset source flags are accumulated, and as a result, the multiple flags are set.

Writing to this register activates the watchdog timer. After that, the watchdog timer continues to operate until a reset (RST) occurs.

[bit15] INIT: External reset generation flag

This bit indicates whether the INITX pin input has generated a reset (INIT).

Value	Description
0	INITX pin input has not generated INIT.
1	INITX pin input has generated INIT.

•INIT is cleared to "0" immediately after reading.

•It is readable. Writing has no effect on the bit value.

[bit14] Reserved: Reserved bit

This is a reserved bit.

[bit13] WDOG: Watchdog reset generation flag

It indicates whether the watchdog timer has generated a reset (INIT).

Value	Description
0	Watchdog timer has not generated INIT.
1	Watchdog timer has generated INIT.

- This bit is cleared to "0" at a reset (INIT) by the INITX pin input upon power-up or immediately after reading.
- It is readable. Writing has no effect on the bit value.

[bit12] Reserved: Reserved bit

This is a reserved bit.

[bit11] SRST: Software reset generation flag

This bit indicates whether a reset (RST) has been generated by writing to the SRST bit in the STCR register (software reset).

Value	Description
0	Software reset has not generated RST.
1	Software reset has generated RST.

- This bit is cleared to "0" at a reset (INIT) by the INITX pin input upon power-up or immediately after reading.
- It is readable. Writing has no effect on the bit value.

Please refer to the limitations of bit9:SYNCR bit of TBCR (time-base counter control register) when using software reset of the synchronous mode.

[bit10] Reserved: Reserved bit

This is a reserved bit.

[bit9, bit8] WT1, WT0: Watchdog timer interval time selection bits

These bits are used to select the cycle for the watchdog timer.

Based on the value written to the bits, the watchdog timer cycle is selected from the 4 options shown in the following table.

WT1	WT0	Minimum interval for writing to WPR, required to prevent a watchdog reset from being generated	Time from when the last $5A_H$ is written to WPR to when a watchdog reset is generated
0	0	$\phi \times 2^{20}$ (Initial value)	$\phi \times 2^{20}$ to $\phi \times 2^{21}$
0	1	$\phi \times 2^{22}$	$\phi \times 2^{22}$ to $\phi \times 2^{23}$
1	0	$\phi \times 2^{24}$	$\phi \times 2^{24}$ to $\phi \times 2^{25}$
1	1	$\phi \times 2^{26}$	$\phi \times 2^{26}$ to $\phi \times 2^{27}$

(ϕ: Cycle of system base clock)

- These bits are initialized to "00_B" by a reset (RST).
- They are readable. Writing is allowed only once after a reset (RST); succeeding write operations are not valid.

■ STCR: Standby Control Register

The configuration of the standby control register is shown below.

STCR	bit	7	6	5	4	3	2	1	0
Address: 000481_H		STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value (INITX pin)		0	0	1	1	0	0	1	1
Initial value (INIT)		0	0	1	1	X	X	1	1
Initial value (RST)		0	0	X	1	X	X	X	X

R/W: Readable/writable

STCR controls the operating mode of the device.

STCR is used to place the device in one of the 2 standby modes (stop/sleep) and stop the pin operation and oscillation during stop mode as well as to set the oscillation stabilization wait time and issue a software reset.

Note:

To place the device in a standby mode, use the synchronous standby mode, set in SYNCS (bit8) of TBCR (time-base counter control register) and be sure to follow the sequence shown below.

```
(LDI#value_of_standby,R0) ; "value_of_standby" is the data written to STCR.
(LDI#_STCR,R12)           ; "_STCR" is the address of STCR (481H).
STB    R0,@R12            ; Writing to the standby control register (STCR)
LDUB   @R12,R0            ; Reading from STCR for synchronous standby
LDUB   @R12,R0            ; Another dummy read from STCR
NOP                       ; NOP (for timing adjustment) × 5
NOP
NOP
NOP
NOP
```

[bit7] STOP: STOP mode bit

This bit directs the device to enter stop mode. If "1" is written to SLEEP (bit6) and this bit at the same time, the device enters stop mode, as the STOP mode bit has higher priority.

Value	Description
0	Device does not enter stop mode [Initial value].
1	Device enters stop mode.

•This bit is initialized to "0" by a reset (RST) or an event that recovers the device from stop mode.

•This bit is readable and writable.

[bit6] SLEEP: SLEEP mode bit

This bit directs the device to enter sleep mode. If "1" is written to STOP (bit7) and this bit at the same time, the device enters stop mode, as the STOP bit (bit7) has higher priority.

Value	Description
0	Device does not enter sleep mode [Initial value].
1	Device enters sleep mode.

•This bit is initialized to "0" by a reset (RST) or an event that recovers the device from sleep mode.

•This bit is readable and writable.

[bit5] HIZ: Hi-Z mode bit

This bit controls the pin state in stop mode.

Value	Description
0	Retains the pin state before transition to stop mode
1	Sets the pin output to high impedance during stop mode [Initial value].

•This bit is initialized to "1" by a reset (INIT).

•This bit is readable and writable.

[bit4] SRST: Software reset bit

SRST directs the issue of a software reset (RST).

Value	Description
0	Issues a software reset.
1	Does not issue a software reset [Initial value].

•This bit is initialized to "1" by a reset (RST).

•This bit is readable and writable. Reading always returns "1".

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

[bit3, bit2] OS1, OS0: Oscillation stabilization wait time selection bits

These bits set the oscillation stabilization wait time after a reset (INIT) or after the device returns from stop mode.

Based on the value written to the bits, the oscillation stabilization wait time is selected from the 4 options shown in the following table.

OS1	OS0	Oscillation Stabilization Wait Time	Source Oscillation: 16.5 MHz	Sub Clock Oscillation: 32 kHz
0	0	$\phi \times 2^1$ [Initial value]	0.242 µs	125 µs
0	1	$\phi \times 2^{11}$	0.248 ms	128 ms
1	0	$\phi \times 2^{16}$	7.94 ms	4 s
1	1	$\phi \times 2^{22}$	508 ms	262 s

ϕ represents the cycle of the system base clock. Here, it is twice the cycle of the input source oscillation.

- These bits are initialized to "00_B" by a reset (INIT) from the INITX pin input.
- These bits are readable and writable.

[bit1] OSCD2: Sub clock oscillation stop bit

OSCD2 stops the sub clock oscillation in stop mode.

Value	Description
0	Does not stop sub clock oscillation during stop mode.
1	Stops sub clock oscillation during stop mode [Initial value].

- This bit is initialized to "1" by a reset (INIT).
- This bit is readable and writable.

[bit0] OSCD1: Main clock oscillation stop bit

OSCD1 stops the oscillation of the main clock in stop mode.

Value	Description
0	Does not stop the main clock oscillation during stop mode.
1	Stops main clock oscillation during stop mode [Initial value].

- This bit is initialized to "1" by a reset (INIT).
- This bit is readable and writable.

■ TBCR: Time-Base Counter Control Register

The configuration of the time-base counter control register is shown below.

TBCR	bit	15	14	13	12	11	10	9	8
Address: 000482_H		TBIF	TBIE	TBC2	TBC1	TBC0	Reserved	SYNCR	SYNCS
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)		0	0	X	X	X	X	0	0
Initial value (RST)		0	0	X	X	X	X	X	X

R/W: Readable/writable

TBCR controls interrupts such as time-base timer interrupts.

TBCR is used to enable time-base timer interrupts, select the interrupt interval time, and set option function of reset operation.

[bit15] TBIF: Time-base timer interrupt flag

TBIF is a time-base timer interrupt flag.

It indicates that the time-base counter has exceeded the specified interval time (set by bit13 to bit11: TBC2 to TBC0).

A time-base timer interrupt request is generated, if this bit is set to "1" while TBIE (bit14) is enabled to generate an interrupt (TBIE=1).

Clearing source	Writing "0" through instruction
Setting source:	Expiration of the set interval time (Detecting the falling edge of the output of the time-base counter)

- This bit is initialized to "0" by a reset (RST).
- It is readable and writable. For write operation, however, only "0" can be written. Writing "1" does not change the bit value.
- Reading by read-modify-write (RMW) instruction always returns "1".

[bit14] TBIE: Time-base timer interrupt enable bit

TBIE enables the output of a time-base timer interrupt request.

TBIE controls the output of an interrupt request due to the expiration of the interval time of the time-base counter. If TBIF (bit15) is set to "1" when this bit is "1", a time-base timer interrupt request is generated.

Value	Description
0	Disables the output of time-base timer interrupt request. [Initial value]
1	Enables the output of time-base timer interrupt request.

- This bit is initialized to "0" by a reset (RST).
- This bit is readable and writable.

[bit13 to bit11] TBC2, TBC1, TBC0: Time-base timer counter selection bits

These bits set the interval time for the time-base counter used in the time-base timer.

Based on the value written to these bits, the interval time is selected from the 8 options shown on the following table.

TBC2	TBC1	TBC0	Timer interval time	When source oscillation = 16.5 MHz, and PLL = multiply-by-2	Sub clock = 32 kHz
0	0	0	$\phi \times 2^{11}$	62.1 µs	61.4 ms
0	0	1	$\phi \times 2^{12}$	124.1 µs	123 ms
0	1	0	$\phi \times 2^{13}$	248.2 µs	246 ms
0	1	1	$\phi \times 2^{22}$	127 ms	126 s
1	0	0	$\phi \times 2^{23}$	254 ms	256 s
1	0	1	$\phi \times 2^{24}$	508 ms	512 s
1	1	0	$\phi \times 2^{25}$	1017 ms	1024 s
1	1	1	$\phi \times 2^{26}$	2034 ms	2048 s

ϕ: Cycle of system base clock

•The initial value is undefined. Always set a value before enabling an interrupt.

•These bits are readable and writable.

[bit10] Reserved: Reserved bit

This is a reserved bit. The read value is undefined. Writing has no effect on operation.

[bit9] SYNCR: Synchronous reset enable bit

SYNCR enables the synchronous reset operation.

It is used to select which should be performed if an operation initialization reset (RST) request occurs: Normal reset operation in which a reset (RST) immediately performed; or synchronous reset in which an operation initialization reset (RST) is performed after all the bus accesses stop.

Value	Description
0	Normal reset operation [Initial value]
1	Synchronous reset operation

•This bit is initialized to "0" by a reset (INIT).

•This bit is readable and writable.

Note:

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

- Set the interrupt enable flag (I-Flag) to interrupt disabled (I-Flag=0).
- Do not used NMI.

[bit8] SYNCS: Synchronous standby enable bit

SYNCS enables the synchronous standby operation.

It is used to select one of the following operations, which is should be performed if an standby request (either sleep or stop mode request) occurs: Normal standby operation, in which the device enters standby mode only by writing to the control bit in the STCR register; or synchronous standby operation in which the device enters standby mode by reading the STCR register after writing to the control bit in the STCR register.

Value	Description
0	Normal standby operation [Initial value]
1	Synchronous standby operation

•This bit is initialized to "0" by a reset (INIT).

•This bit is readable and writable.

Note:

Please set the synchronous standby operation by setting "1" to this bit at changing to the standby mode.

■ CTBR: Time-Base Counter Clear Register

The configuration of the time-base counter clear register is shown below.

CTBR	bit	7	6	5	4	3	2	1	0
Address: 000483_H		D7	D6	D5	D4	D3	D2	D1	D0
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
	Initial value	X	X	X	X	X	X	X	X

W: Write only

CTBR is used to initialize the time-base counter.

When "$A5_H$" and "$5A_H$" are written to this register consecutively, all the bits of the time-base counter are cleared to "0" immediately after "$5A_H$" is written. There is no time limit between writing "$A5_H$" and "$5A_H$". However, if data other than "$5A_H$" is written after "$A5_H$" is written, clear operation will not be performed even when "$5A_H$" is written, unless "$A5_H$" is written again.

The read value of this register is undefined.

Note:

When this register is used to clear the time-base counter, there will be temporary fluctuations in the oscillation stabilization wait interval, watchdog timer cycle, and time-base timer cycle.

■ CLKR: Clock Source Control Register

The configuration of the clock source control register is shown below.

CLKR	bit	15	14	13	12	11	10	9	8
Address: 000484_H		Reserved	PLL1S2	PLL1S1	PLL1S0	PLL2EN	PLL1EN	CLKS1	CLKS0
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value (INIT)		0	0	0	0	0	0	0	0
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable

CLKR selects the clock source to be used as the system base clock and controls the PLL.

This register is used to select one out of the three available clock sources. It is also used to enable the operation of the dual clock PLL (operation of both main clock and sub clock) and select the multiplication rate.

[bit15] Reserved: Reserved bit

This is a reserved bit. Be sure to set the bit to "0".

[bit14 to bit12] PLL1S2, PLL1S1, PLL1S0: PLL multiplication rate selection bits

These bits are used to select the multiplication rate for the main PLL.

The multiplication rate for the main PLL is selected from 8 options.

Do not rewrite this bit while the main PLL is selected as the clock source.

The maximum operable frequency is 33 MHz. Therefore, do not set any higher frequency. This bit is initialized to "000_B" by a reset (INIT).

PLL1S2	PLL1S1	PLL1S0	Main PLL multiplication rate	Source oscillation: 16.5 [MHz]
0	0	0	× 1 (equal)	ϕ = 60.6 ns (16.5 MHz)
0	0	1	× 2 (multiply-by-2)	ϕ = 30.3 ns (33 MHz)
0	1	0	× 3 (multiply-by-3)	ϕ = 20.2 ns (49.5 MHz)
0	1	1	× 4 (multiply-by-4)	Setting disabled
1	0	0	× 5 (multiply-by-5)	Setting disabled
1	0	1	× 6 (multiply-by-6)	Setting disabled
1	1	0	× 7 (multiply-by-7)	Setting disabled
1	1	1	× 8 (multiply-by-8)	Setting disabled

ϕ: Cycle of system base clock

These bits are readable and writable.

[bit11] PLL2EN: Sub clock selection enable bit

PLL2EN is a selection enable bit for the sub clock.

Do not rewrite this bit while the sub clock is selected as the clock source. Also, do not select the sub clock as the clock source while this bit is set to "0" (due to the settings of bit9 and bit8: CLKS1 and CLKS0).

If OSCD2 (bit1) in STCR is "1", the sub clock will be stopped during stop mode, even when PLL2EN is "1". The operation will be enabled again after return from stop mode.

Value	Description
0	Disables sub clock selection. [Initial value]
1	Enables sub clock selection.

•This bit is initialized to "0" by a reset (INIT).

•This bit is readable and writable.

[bit10] PLL1EN: Main PLL enable bit

PLL1EN is an operation enable bit for the main PLL.

Do not rewrite this bit while the main PLL is selected as the clock source. Also, do not select the main PLL as the clock source while this bit is set to "0" (due to the settings of bit9 and bit8: CLKS1 and CLKS0).

If OSCD1 (bit0) in STCR is "1", the main PLL will be stopped during stop mode, even when PLL1EN is "1". The operation will be enabled again after return from stop mode.

Value	Description
0	Stops main PLL. [Initial value]
1	Enables main PLL operation.

•This bit is initialized to "0" by a reset (INIT).

•This bit is readable and writable.

[bit9, bit8] CLKS1, CLKS0: Clock source selection bits

These bits set the clock source to be used.

Based on the value written to the bits, the clock source is selected from the 3 options shown in the following table.

Note that the value of CLKS0 (bit8) cannot be modified while CLKS1 (bit9) is "1".

Unchangeable combination	Changeable combination
"00_B"→"11_B"	"00_B"→"01_B" or "10_B"
"01_B"→"10_B"	"01_B"→"11_B" or "00_B"
"10_B"→"01_B" or "11_B"	"10_B"→"00_B"
"11_B"→"00_B" or "10_B"	"11_B"→"01_B"

For the above reason, write "01" first, then write "11" to switch from the post-INIT state to the sub clock selection.

CLKS1	CLKS0	Clock source setting
0	0	Source oscillation input from X0/X1 divided by 2 [Initial value]
0	1	Source oscillation input from X0/X1 divided by 2
1	0	Main PLL
1	1	Sub clock

•This bit is initialized to "00_B" by a reset (INIT).

•These bits are readable and writable.

■ WPR: Watchdog Reset Generation Delay Register

The configuration of the watchdog reset generation delay register is shown below.

WPR	bit	7	6	5	4	3	2	1	0
Address: 000485_H		D7	D6	D5	D4	D3	D2	D1	D0
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
	Initial value	X	X	X	X	X	X	X	X

W: Write only

WPR is a register used to delay the generation of a watchdog reset.

When "$A5_H$" and "$5A_H$" are written to this register consecutively, flip-flop for detecting the watchdog timer is cleared immediately after "$5A_H$" is written, to delay the generation of a watchdog reset.

There is no time limit between writing "$A5_H$" and "$5A_H$". However, if data other than "$5A_H$" is written after "$A5_H$" is written, clear operation will not be performed even when "$5A_H$" is written, unless "$A5_H$" is written again.

Table 2.9-1 shows the relationship between the time interval pertaining to the generation of watchdog resets and the RSRR register value.

Unless the writing of both pieces of the data is completed within this interval, a watchdog reset is generated. The time spent until the generation of a watchdog reset and the writing interval required to inhibit the generation vary depending on the state of WT1 (bit9) and WT0 (bit8) in the RSRR register.

Table 2.9-1 Time Interval of Watchdog Reset Generation

WT1	WT0	Minimum interval for writing to WPR, required to prevent a watchdog reset to be generated by RSRR	Time from when the last $5A_H$ is written to WPR to when a watchdog reset is generated
0	0	$\phi \times 2^{20}$ [Initial value]	$\phi \times 2^{20}$ to $\phi \times 2^{21}$
0	1	$\phi \times 2^{22}$	$\phi \times 2^{22}$ to $\phi \times 2^{23}$
1	0	$\phi \times 2^{24}$	$\phi \times 2^{24}$ to $\phi \times 2^{25}$
1	1	$\phi \times 2^{26}$	$\phi \times 2^{26}$ to $\phi \times 2^{27}$

ϕ represents the cycle of system base clock. WT1 and WT0 are bit9 and bit8 of RSRR, used to set the cycle of the watchdog timer.

When the CPU is not in operation, such as during stop mode, sleep mode and DMA transfer, clear operation is performed automatically. Therefore, once such condition occurs, a watchdog reset is delayed automatically. However, a watchdog reset is not postponed when an external bus hold request (BRQ) has been accepted. To hold the external bus for a long time, enter sleep mode and then input a hold request (BRQ).

The read value of this register is undefined.

■ DIVR0: Basic Clock Division Setting Register 0

The configuration of the basic clock division setting register 0 is shown below.

DIVR0	bit	15	14	13	12	11	10	9	8
Address: 000486_H		B3	B2	B1	B0	P3	P2	P1	P0
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value (INIT)		0	0	0	0	0	0	1	1
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable

DIVR0 is a register used to control the division ratio of the base clock for each internal clock.

This register sets the division ratio between the CPU and the clock of the internal bus (CLKB), a peripheral circuit and peripheral bus clock (CLKP).

Note:

The maximum operable frequency is defined for each clock. Operation is not guaranteed if a frequency, in combination with the source clock selection, PLL multiplication rate setting and division ratio setting, is set to exceed the maximum frequency. In particular, take care to follow the correct order in conjunction with modifying the source clock selection setting.

When a setting of this register is changed, the new division ratio becomes valid from the next clock rate.

[bit15 to bit12] B3, B2, B1, B0: CLKB division selection bits

These bits are used to set the division ratio for the CPU clock (CLKB). They set the clock division ratio for the CPU, internal memory and internal bus.

Based on the value written to these bits, the division ratio (clock frequency) of the base clock for the CPU and internal bus is selected from the 16 options shown in the following table.

The maximum operable frequency is 33 MHz. Therefore, do not set any division ratio that will cause this frequency to be exceeded.

B3	B2	B1	B0	Clock division ratio	Clock frequency: When source oscillation is 16.5 MHz, and PLL is multiplied by 2
0	0	0	0	ϕ	33.0 MHz [Initial value]
0	0	0	1	$\phi \times 2$ (divided-by-2)	16.5 MHz
0	0	1	0	$\phi \times 3$ (divided-by-3)	11 MHz
0	0	1	1	$\phi \times 4$ (divided-by-4)	8.25 MHz
0	1	0	0	$\phi \times 5$ (divided-by-5)	6.6 MHz
0	1	0	1	$\phi \times 6$ (divided-by-6)	5.5 MHz
0	1	1	0	$\phi \times 7$ (divided-by-7)	4.71 MHz
0	1	1	1	$\phi \times 8$ (divided-by-8)	4.13 MHz
…	…	…	…	…	…
1	1	1	1	$\phi \times 16$ (divided-by-16)	2.06 MHz

ϕ: Cycle of system base clock

•These bits are initialized to "0000_B" by a reset (INIT).

•These bits are readable and writable.

[bit11 to bit8] P3, P2, P1, P0: CLKP division selection bits

These bits are used to set the division ratio for the peripheral clock (CLKP).

They set the division ratio for the clock used for peripheral circuits and peripheral bus.

Based on the value written to these bits, the division ratio (clock frequency) of the base clock for the peripheral circuit and peripheral bus clocks is selected from the 16 options shown in the following table.

The maximum operable frequency is 33 MHz. Therefore, do not set any division ratio that will cause this frequency to be exceeded.

P3	P2	P1	P0	Clock division ratio	Clock frequency: When source oscillation is 16.5 MHz, and PLL is multiplied by 2
0	0	0	0	ϕ	33.0 MHz
0	0	0	1	$\phi \times 2$ (divided-by-2)	16.5 MHz
0	0	1	0	$\phi \times 3$ (divided-by-3)	11 MHz
0	0	1	1	$\phi \times 4$ (divided-by-4)	8.25 MHz [Initial value]
0	1	0	0	$\phi \times 5$ (divided-by-5)	6.6 MHz
0	1	0	1	$\phi \times 6$ (divided-by-6)	5.5 MHz
0	1	1	0	$\phi \times 7$ (divided-by-7)	4.71 MHz
0	1	1	1	$\phi \times 8$ (divided-by-8)	4.13 MHz
…	…	…	…	…	…
1	1	1	1	$\phi \times 16$ (divided-by-16)	2.06 MHz

ϕ: Cycle of system base clock

•These bits are initialized to "0011_B" by a reset (INIT).

•These bits are readable and writable.

■ DIVR1: Basic Clock Division Setting Register 1

The configuration of the basic clock division setting register 1 is shown below.

DIVR1	bit	7	6	5	4	3	2	1	0
Address: 000487_H		T3	T2	T1	T0	Reserved	Reserved	Reserved	Reserved
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value (INIT)		0	0	0	0	0	0	0	0
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable

DIVR1 is a register used to control the division ratio of the base clock for each internal clock.

This register sets the division ratio of the clock for an external expansion bus interface (CLKT).

Note:

The maximum operable frequency is defined for each clock. Operation is not guaranteed if a frequency, in combination with the source clock selection, PLL multiplication rate setting and division ratio setting, is set to exceed the maximum frequency. In particular, take care to follow the correct order in conjunction with modifying the source clock selection setting.

When a setting of this register is changed, the new division ratio becomes valid from the next clock rate.

[bit7 to bit4] T3, T2, T1, T0: CLKT division selection bits

These bits are used to set the division ratio for the external bus clock (CLKT).

They set the clock division ratio for the external extended bus interface.

Based on the value written to these bits, the division ratio (clock frequency) of the base clock for the external expansion bus interface is selected from the 16 options shown in the following table.

The maximum operable frequency is 16.5 MHz. Therefore, do not set any division ratio that will cause this frequency to be exceeded.

T3	T2	T1	T0	Clock division ratio	Clock frequency: When source oscillation is 16.5 MHz, and PLL is multiplied by 2
0	0	0	0	ϕ	33.0 MHz [Initial value]
0	0	0	1	$\phi \times 2$ (divided-by-2)	16.5 MHz
0	0	1	0	$\phi \times 3$ (divided-by-3)	11 MHz
0	0	1	1	$\phi \times 4$ (divided-by-4)	8.25 MHz
0	1	0	0	$\phi \times 5$ (divided-by-5)	6.6 MHz
0	1	0	1	$\phi \times 6$ (divided-by-6)	5.5 MHz
0	1	1	0	$\phi \times 7$ (divided-by-7)	4.71 MHz
0	1	1	1	$\phi \times 8$ (divided-by-8)	4.13 MHz
…	…	…	…	…	…
1	1	1	1	$\phi \times 16$ (divided-by-16)	2.06 MHz

ϕ: Cycle of system base clock

•These bits are initialized to "0000_B" by a reset (INIT).

•These bits are readable and writable.

[bit3 to bit0] Reserved: Reserved bits

These are reserved bits.

■ OSCCR: Oscillation Control Register

The configuration of the oscillation control register is shown below.

OSCCR	bit	15	14	13	12	11	10	9	8
Address: 00048A$_H$		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	OSCDS1
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value (INIT)		X	X	X	X	X	X	X	0
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable

OSCCR is a register used to control the main oscillation during sub clock operation.

[bit15 to bit9] Reserved: Reserved bits

These are reserved bits.

[bit8] OSCDS1: Main oscillation stop control bit (in sub clock operation mode)

OSCDS1 is used to stop the main oscillation while the sub clock is selected.

Writing "1" to this bit stops the main oscillation when the sub clock is selected as the clock source.

"1" cannot be written to this bit when the main clock is selected.

Do not select the main clock while this bit is "1". Set it to "0" and wait until the main oscillation stabilizes before switching to the main clock. In this case, maintain the oscillation stabilization wait time using the main oscillation stabilization wait timer. Moreover, the main oscillation stabilization wait time is required when the clock source is switched to the main clock by INIT. At this point, the operation after return is not guaranteed unless the settings of OS1 and OS0 (bit3 and bit2) in STCR (standby control register) satisfy the main oscillation stabilization wait time.

In the above case, set OS1 and OS0 in STCR to a value that will satisfy both the sub clock oscillation stabilization wait time and the main oscillation stabilization wait time.

For information about the oscillation stabilization waiting, see "2.9.2 Oscillation Stabilization Wait and PLL Lock Wait Time".

Value	Description
0	Does not stop main oscillation during execution of sub clock [Initial value].
1	Stop main oscillation during execution of sub clock.

•This bit is initialized to "0" by a reset (INIT).

•This bit is readable and writable.

2.9.7 Peripheral Circuits in Clock Control Block

This section explains the peripheral circuit functions contained in the clock control block.

■ Time-Base Counter

The clock control block includes a 26-bit time-base counter which runs on the system base clock.

In addition to measuring the oscillation stabilization wait time (see "2.8.4 Oscillation Stabilization Wait Time"), the time-base counter is used for the following applications.

- Watchdog timer:
  The bit output of the time-base counter is used to measure the watchdog timer for detecting system hang-up.
- Time-base timer:
  The output of the time-base counter is used to generate interval interrupts.

● Watchdog timer

The watchdog timer detects program hang-up, using the output of the time-base counter. When the generation of a watchdog reset is no longer delayed during the set interval by an event such as program hang-up, a setting initialization reset (INIT) request is generated as a watchdog reset.

[Activating the watchdog timer and setting the cycle]

The watchdog timer is activated when RSRR (reset source register/watchdog timer control register) is written to for the first time after a reset (RST).

At this point, WT1 and WT0 (bit9 and bit8) are used to set the interval time for the watchdog timer. For setting the interval time, only the time set in the initial write operation becomes valid. Any succeeding write attempts are ignored.

[Delaying the generation of watchdog reset]

Once the watchdog timer is activated, data must be written periodically to WPR (watchdog reset generation delay register) in the order of "$A5_H$" and "$5A_H$", using the program.

This procedure initializes the flag for generating a watchdog reset.

[Generating watchdog reset]

The flag for generating a watchdog reset is set at the falling edge of the output of the time-base counter in the set interval. If the flag has been set when the second falling edge is detected, a setting initialization reset (INIT) request is generated as a watchdog reset.

[Stopping the watchdog timer]

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

The watchdog timer can be stopped under the following condition in which an operation initialization reset (RST) is generated, and it does not function until reactivated by program operation.

- Operation initialization reset (RST) state
- Setting initialization reset (INIT) state
- Oscillation stabilization wait reset (RST) state

[Suspending the watchdog timer (automatically delayed generation)]

While the CPU's program operation is stopped, the watchdog timer once initializes the flag for generating a watchdog reset to delay the generation of such reset. "Suspended program operation" refers to specific operations listed below.

- Sleep state
- Stop state
- Oscillation stabilization wait RUN state
- Break when emulator debugger and monitor debugger is used
- Period from execution of INTE instruction to execution of RETI instruction
- Step trace trap (Break for each instruction when T flag in PS register is set to "1")
- Data to cache memory at instruction cache control register (ISIZE, ICHCR) or RAM mode

When the time-base counter is cleared, the flag for generating a watchdog reset is also initialized at the same time, and the generation of a watchdog reset is postponed.

Note that a watchdog reset may not be generated if system hang-up results in the above state. In that case, perform a reset (INIT) from the external INITX pin.

● Time-base timer

The time-base timer is a timer that generates interval interrupts using the output of the time-base counter. The timer is suitable for measuring relatively long times, up to {base clock × 2^{27}} cycles such as for the PLL lock wait time and sub clock oscillation stabilization wait time.

A time-base timer interrupt request is generated when the falling edge of the output of the time-base counter corresponding to the set interval is detected.

[Activating the time-base timer and setting the interval]

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

The falling edge of the output of the time-base counter corresponding to the set interval is always detected. Therefore, after setting the interval time, clear TBIF (bit15) first, then set TBIE (bit14) to "1" to enable the output of an interrupt request.

When changing the interval time, disable the output of an interrupt request by setting TBIE (bit14) to "0" beforehand.

The time-base counter always continues to count without being affected by the above settings. To achieve the accurate interval interrupt time, clear the time-base counter before enabling interrupts. Otherwise, an interrupt request may be generated immediately after interrupts are enabled.

[Clearing the time-base counter by program]

When "$A5_H$" and "$5A_H$" are written to the time-base counter clear register (CTBR) in that order, all the bits of the time-base counter are cleared to "0" immediately after "$5A_H$" is written. There is no time limit between writing "$A5_H$" and "$5A_H$". However, if data other than "$5A_H$" is written after "$A5_H$" is written, clear operation will not be performed even when "$5A_H$" is written, unless "$A5_H$" is written again.

When this time-base counter is cleared, the flag for generating a watchdog reset is also initialized simultaneously, and the generation of a watchdog reset is postponed temporarily.

[Clearing the time-base counter by the device state]

When the device enters the following state, all the bits of the time-base counter are cleared to "0".

•Stop state

•Setting initialization reset (INIT) state

Particularly in stop state, a time-base timer interval interrupt may occur unintentionally, as the time-base counter is used to measure the oscillation stabilization wait time.

Therefore, disable time-base timer interrupts and do not use the time-base timer before selecting stop mode.

In any other state, an operation initialization reset (RST) is generated automatically. Consequently, time-base timer interrupts are disabled automatically.

● Main oscillation stabilization wait timer (when sub clock is selected)

This is a 26-bit timer that is synchronized with the main clock to count up, without being affected by the clock source selection or division setting.

The timer is used to measure the main oscillation stabilization wait time during sub clock operation.

The main oscillation can be controlled in sub clock operation by OSCDS1 (bit8) in OSCCR (oscillation control register). This timer is used to measure the oscillation stabilization wait time, when the main oscillation is stopped and then restarted.

Use the following procedure to switch to the main clock operation from the sub clock operation with the main clock being stopped.

1) Clear the main oscillation stabilization wait timer.
2) Set OSCDS1 (bit8) in OSCCR (oscillation control register) to "0" to start the main oscillation.
3) Use the main oscillation stabilization wait timer to wait until the main clock becomes stable.
4) Once the main clock stabilizes, use CLKS1 and CLKS0 (bit9 and bit8) in CLKR (clock source register) to switch from the sub clock to the main clock.

Note:

If the clock is switched to the main clock without waiting until it becomes stable, an unstable clock signal will be supplied; therefore, the resulting operation will not be guaranteed. Always wait until the clock stabilizes before switching to the main clock.

For details about the main oscillation stabilization wait timer, see "9.1 Main Oscillation Stabilization Wait Timer".

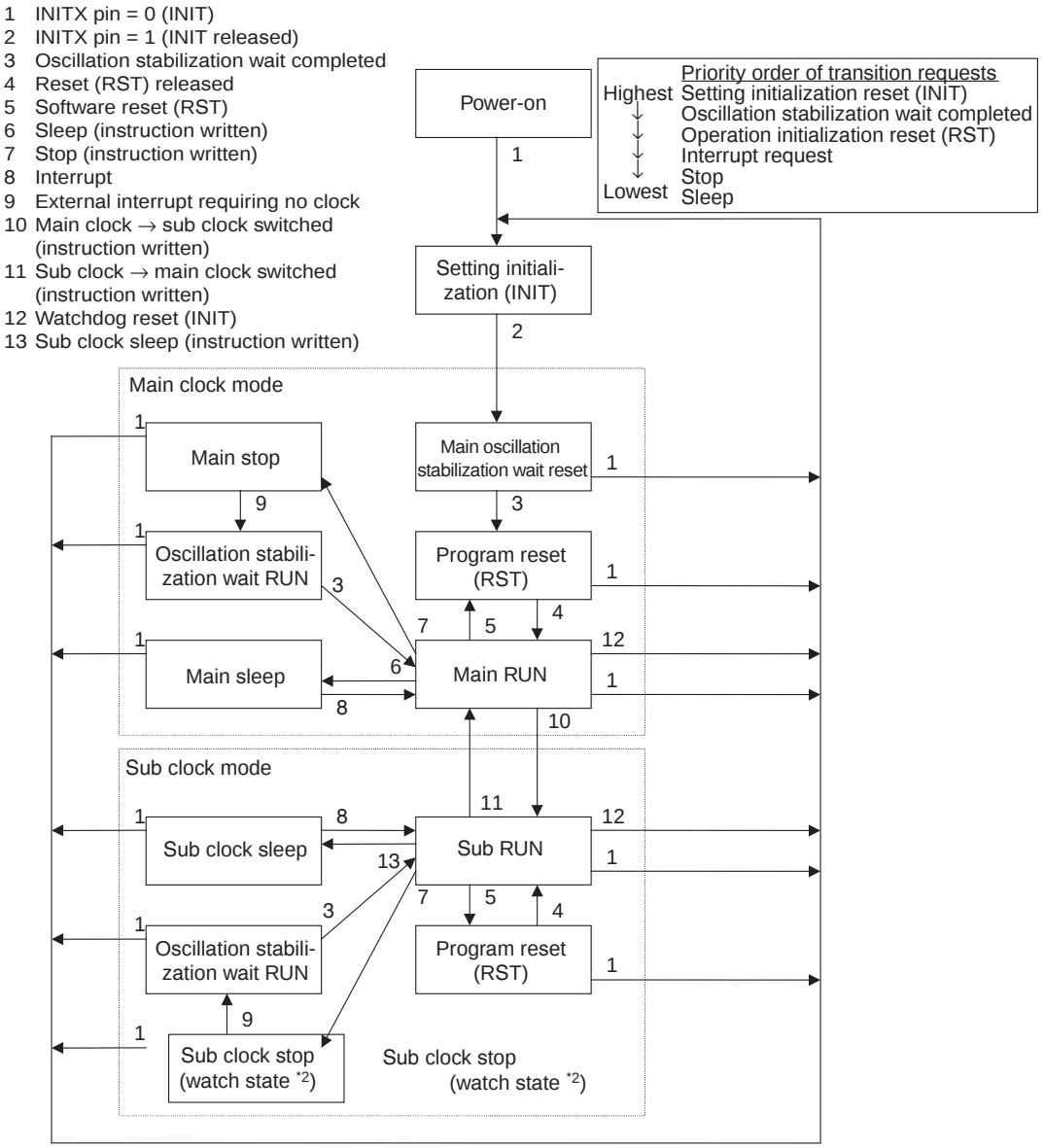
2.10 Device State Control

This section explains various states of this model and how they are controlled.

■ Device States

Figure 2.10-1 shows the device states and each of the transitions of the FR family.

Figure 2.10-1 Device States



*1: To switch the clock source between the main clock and sub clock, switch CLKS1 and CLKS0 (bit1, bit0) in the CLKR (clock source register) while a stable supply of the switched clock is maintained in operation mode.

*2: To stop anything other than the watch timer (watch state), place the device in stop mode while OSCD2 (bit1) in the STCR (standby control register) is set to "0" and OSCD1 (bit0) is set to "1" in the sub clock operation state (they can be written simultaneously).

■ Operational States of Device

This model is provided with the following device operational states:

- RUN state (normal operation)

  In this state, the program is being executed.

  All internal clocks are supplied to keep all circuits operable.

  Only the clock for the 16-bit peripheral bus, however, is stopped unless the bus is accessed.

  In this state, a state transition request is accepted. If synchronous reset mode is selected, however, state transition operations different from normal reset mode are used for some requests.

  For details, see "■ Synchronous Reset Operation" section of "2.8.5 Reset Operation Modes".

- Sleep state

  In this state, the program is stopped. The device is placed in this state by program operation.

  Only the execution of the program by the CPU is stopped, while the peripheral circuits are maintained operable. The built-in memory and internal/external buses remain stopped unless requested by the DMA controller. When a valid interrupt request is generated, this state is released and the device enters the RUN state (normal operation).

  When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state.

  When an operation initialization reset (RST) request is generated, the device enters the operation initialization reset (RST) state.

- Stop state

  In this state, the device is stopped. The device is placed in this state by program operation.

  In the stop state, all the internal circuits are stopped. The internal clocks are all stopped and the oscillation circuit and PLL can be stopped by setting. Also, the external pins can be set to the same high impedance by setting (except some pins).

  The device enters the oscillation stabilization wait RUN state when a specific (non-clock-based) valid interrupt occurs or when a main oscillation stabilization wait timer interrupt request is generated during oscillation operation.

  When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state.

  When an operation initialization reset (RST) request is generated, the device enters the oscillation stabilization wait reset (RST) state.

- Oscillation stabilization wait RUN state

  In this state, the device is stopped. The device enters this state when returning from the stop state.

  All the internal circuits except the clock generation control block (time-base counter and device state control component) are stopped. While all the internal clocks are stopped, the oscillation circuit and the PLL which has been enabled to operate are in operation.

  This state releases the high impedance control of the external pins used in the stop state.

  The device enters the RUN state (normal operation) when the set oscillation stabilization wait time has elapsed.

When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state.

When an operation initialization reset (RST) request is generated, the device enters the oscillation stabilization wait reset (RST) state.

- Oscillation stabilization wait reset (RST) state

In this state, the device is stopped. The device enters this state after returning from the stop state or setting initialization reset (INIT) state.

All the internal circuits except the clock generation control block (time-base counter and device state control component) are stopped. While all the internal clocks are stopped, the oscillation circuit and the PLL which has been enabled to operate are in operation.

This state releases the high impedance control of the external pins used in the stop state.

It outputs an operation initialization reset (RST) to the internal circuits.

The device enters the oscillation stabilization wait reset (RST) state when the set oscillation stabilization wait time has elapsed.

When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state.

- Operation initialization reset (RST) state

In this state, the program is initialized. The device enters this state once the operation initialization reset (RST) request is accepted or the oscillation stabilization wait reset (RST) state ends.

The execution of the program by the CPU is stopped and the program counter is initialized. Most peripheral circuits are initialized. All the internal clocks, the oscillation circuit and the PLL which has been enabled to operate are in operation.

It outputs an operation initialization reset (RST) to the internal circuits.

When the operation initialization reset (RST) request is lost, the device enters the RUN state (normal operation) to execute the operation initialization reset sequence. If the device has just returned from the setting initialization reset (INIT) state, it will execute the setting initialization reset sequence.

When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state.

- Setting initialization reset (INIT) state

In this state, all settings are initialized. The device enters this state once the settings initialization reset (INIT) request is accepted or the hardware standby state ends.

The execution of the program by the CPU is stopped and the program counter is initialized. All the peripheral circuits are initialized. Although the oscillation circuit is in operation, the PLL stops operation. All the internal clocks are stopped while the "L" level is being input to the external INITX pin. In other times, they operate.

A setting initialization reset (INIT) and operation initialization reset (RST) are output to the internal circuits.

When the setting initialization reset (INIT) request is lost, this state is released and the device enters the oscillation stabilization wait reset (RST) state. After that, the device goes through the operation initialization reset (RST) state and executes the setting initialization reset sequence.

● Priority order of state transition requests

In any state, the device follows the priority order of the state transition requests, shown below. Note that some requests can be generated only in a particular state; therefore, they are only valid in that state.

[Highest] Setting initialization reset (INIT) request

↓ Oscillation stabilization wait time completed

(This occurs only in oscillation stabilization wait reset state and oscillation stabilization wait RUN state)

↓ Operation initialization reset (RST) request

↓ Valid interrupt request (Generated only in RUN, sleep and stop state)

↓ Stop mode request (register written) (Generated only in RUN state)

[Lowest] Sleep mode request (register written) (Generated only in RUN state)

2.10.1 Low-power Consumption Mode

Of all the operational states available in this model, this section explains the low-power consumption modes as well as how to use them.
This model is provided with the following low-power consumption modes:

- **Sleep Mode**
  The device is set to sleep mode by writing to a register.
- **Stop Mode**
  The device is set to stop mode by writing to a register.

Each mode is explained below.

■ Sleep Mode

When "1" is written to the SLEEP bit (bit6) in STCR (standby control register), sleep mode is selected and the device enters sleep mode.

After that, the device remains in sleep state until an event allowing the device to recover from sleep state occurs.

When "1" is written to both the STOP bit (bit7) in STCR (standby control register) and this bit, the STOP bit (bit7) has higher priority; therefore, the device enters the stop state.

For information about the sleep state, also see the "■ Operational States of Device" section of "2.10 Device State Control".

[Transition to sleep mode]

To place the device in a sleep mode, use the synchronous standby mode, set in SYNCS (bit8) of TBCR (time-base counter control register) and be sure to follow the sequence shown below.

```
(LDI  #value_of_sleep,R0)    ; "value_of_sleep" is the data written to STCR.
(LDI  #_STCR,R12)            ; "_STCR" is the address of STCR (481H).
STB     R0,@R12              ; Writing to the standby control register (STCR)
LDUB@R12,R0                  ; Reading from STCR for synchronous standby
LDUB@R12,R0                  ; Another dummy read from STCR
NOP                          ; NOP (for timing adjustment) × 5
NOP
NOP
NOP
NOP
```

[Circuits that stop in sleep state]

- Execution of program by CPU
- Bit search module (It however operates during DMA transfer.)
- Various types of built-in memory (It however operates during DMA transfer.)
- Internal/external buses (They however operate during DMA transfer.)

[Circuits that do not stop in sleep state]

- •Oscillation circuit
- •PLL enabled to operate
- •Clock generation control block
- •Interrupt controller
- •Peripheral circuits
- •DMA controller
- •DSU
- •Main oscillation stabilization wait timer

[Events that recover the device from sleep state]

- •Generation of a valid interrupt request

  When an interrupt request holding an interrupt level other than for disabling interrupt ($1F_H$) is generated, the device is released from sleep mode and enters the RUN state (normal state).

  To maintain the device in sleep mode even when an interrupt request is generated, set the relevant ICR to disable interrupts ($1F_H$) for the interrupt level.

- •Generation of a setting initialization reset (INIT) request

  When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state unconditionally.

- •Generation of an operation initialization reset (RST) request

  When an operation initialization reset (RST) request is generated, the device enters the operation initialization reset (RST) state unconditionally.

For information about the priority order of the sources, see "2.10 Device State Control".

[Synchronous standby operation]

If "1" is set for bit8:SYNCS bit of the time base counter control register (TBCR), synchronous standby operation is enabled. In this case, the device does not enter the sleep state only by writing to the SLEEP bit.

After the write operation, the STCR register must also be read from in order to set the device to the sleep state.

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

■ Stop Mode

When "1" is written to the STOP bit (bit7) in STCR (standby control register), stop mode is selected and the device enters the stop state. After that, the device remains in the stop state, until an event allowing the device to recover from the stop state occurs.

When "1" is written to both the SLEEP bit (bit6) in STCR (standby control register) and this bit, the STOP bit (bit7) has higher priority; therefore, the device enters the stop state.

For information about the stop state, also see the "■ Operational States of Device" section of "2.10 Device State Control".

[Stop mode transition]

Use the following sequences sifter using the synchronous standby mode (TBCR: Set by time base counter control register bit8 CYNCS bit) when putting in the stop mode.

```
(LDI  #value_of_stop,R0)     ; "value_of_stop" is the data written to STCR.
(LDI  #_STCR,R12)            ; "_STCR" is the address of STCR (481H).
STB     R0,@R12              ; Writing to the standby control register (STCR)
LDUB @R12,R0                 ; Reading from STCR for synchronous standby
LDUB @R12,R0                 ; Another dummy read from STCR
NOP                          ; NOP (for timing adjustment) × 5
NOP
NOP
NOP
NOP
```

Furthermore, set the I flag, ILM and ICR to ensure that branching into the interrupt handler which is a return source occurs after returning to the standby mode.

[Circuits that halt in stop state]

- Oscillation circuit that has been set to stop:

  The oscillation circuit for the sub clock that is in the stop state is set to halt, when OSCD2 (bit1) in STCR (standby control register) is set to "1". In this case, the watch timer also halts.

  The oscillation circuit for the main clock that is in the stop state is set to halt, when OSCD1 (bit0) in STCR (standby control register) is set to "1". In this case, the main oscillation stabilization wait timer also halts.

- PLL that is connected to the oscillation circuit either disabled to operate or set to halt:

  When OSCD2 (bit1) in STCR (standby control register) is set to "1", the PLL for the sub clock that is in the stop state is set to halt, even if PLL2EN (bit11) in CLKR (clock source control register) is set to "1".

  When OSCD1 (bit0) in STCR (standby control register) is set to "1", the PLL for the main clock that is in the stop state is set to halt, even if PLL1EN (bit10) in CLKR (clock source control register) is set to "1".

- All the internal circuits except the [Circuits that do not halt in stop state]

[Circuits that do not halt in stop state]

•Oscillation circuits that are not set to halt:

The oscillation circuit for the sub clock that is in the stop state does not halt, when OSCD2 (bit1) in STCR (standby control register) is set to "0". In this case, the watch timer also does not halt.

The oscillation circuit for the main clock that is in the stop state does not halt, when OSCD1 (bit0) in STCR (standby control register) is set to "0". In this case, the main oscillation stabilization wait timer also does not halt.

•PLL that is connected to the oscillation circuit which is enabled to operate and not set to halt: When OSCD2 (bit1) in STCR (standby control register) is set to "0", the PLL for the sub clock that is in the stop state does not halt, if PLL2EN (bit11) in CLKR (clock source control register) is set to "1".

When OSCD1 (bit0) in STCR (standby control register) is set to "0", the PLL for the main clock that is in the stop state does not halt, if PLL1EN (bit10) in CLKR (clock source control register) is set to "1".

[High-impedance control of pins in stop state]

The pin outputs in the stop state are set to high impedance, if the HIZ bit (bit5) in STCR (standby control register) is set to "1". For information about the pins subject to this control feature, see "APPENDIX C Pin Status In Each CPU State".

The pin outputs in the stop state retain the value used before the transition to the stop state, if the HIZ bit (bit5) in STCR (standby control register) is set to "0". For details, see "APPENDIX C Pin Status In Each CPU State".

[Events that recover the device from stop state]

•Generation of a specific (non-clock-based) valid interrupt request:

Only the following are valid: external interrupt input pin (INTn pin), main oscillation stabilization wait timer interrupt during main oscillation, and watch interrupt during sub clock oscillation.

When an interrupt request holding an interrupt level other than for disabling interrupt ($1F_H$) is generated, the device is released from stop mode and enters the RUN state (normal state).

To maintain the device in stop mode even when an interrupt request is generated, set the relevant ICR to disable interrupts ($1F_H$) for the interrupt level.

•Main oscillation stabilization wait timer interrupt:

If a main oscillation stabilization wait timer interrupt request is generated, either when OSCDS1 (bit8) in OSCCR (oscillation control register) is set to "0" with the sub clock being selected, or when OSCD1 (bit0) in STCR (standby control register) is set to "0" with the main clock being selected, the device is released from stop mode and enters the RUN state (normal state).

To maintain the device in stop mode even when an interrupt request is generated, stop the main oscillation stabilization wait timer or disable the interrupt enable bit of the main oscillation stabilization wait timer.

•Generation of a setting initialization reset (INIT) request:

When a setting initialization reset (INIT) request is generated, the device enters the setting initialization reset (INIT) state unconditionally.

•Generation of an operation initialization reset (RST) request:

When an operation initialization reset (RST) request is generated, the device enters the operation initialization reset (RST) state unconditionally.

For information about the priority order of the sources, see the "■ Operational States of Device" section of "2.10 Device State Control".

[Clock source selection in stop mode]

Select the main clock divided by 2 as the source clock before setting stop mode. For details, see "2.9 Clock Generation Control", especially "2.9.1 PLL Control" in that section.

Note that the same restrictions as in normal operation apply when setting the division ratio.

MB91313A Series

2.11 Operating Mode

This section explains the operating modes of the FR family.

■ Operating Mode

The operating modes include bus mode and access mode.

● Bus Modes

Bus mode is a mode that controls the operations of the internal ROM and external access function. It is selected based on the content of the mode setting pins (MD2, MD1, and MD0).

● Access Mode

Access mode is a mode that controls the external data bus width. It is selected by the WTH1 and WTH0 bits in the mode register and the DBW0 bit in ACR0 to ACR3 (Area Configuration Register).

■ Bus Mode 0 (Single-chip Mode)

In this mode, the internal I/O, F-bus RAM and F-bus ROM are enabled, but access to other areas is disabled.

The external pins serve as either a peripheral or general-purpose port. They do not function as bus pins.

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

In this mode, the internal I/O, F-bus RAM and F-bus ROM are enabled, and access to an externally accessible area is handled as access to an external space. Some external pins serve as bus pins.

■ Mode Setting

In the FR family, each operating mode is set by the mode pins (MD2, MD1, and MD0) and the mode register (MODR).

● Mode pins

3 pins (MD2, MD1, and MD0) are used for specification related to mode vector fetch.

Table 2.11-1 lists specification pertaining to mode vector fetch.

Table 2.11-1 Mode Vector Fetch Related Specification

Mode pins MD2, MD1, MD0	Mode name	Reset vector access area
000_B	Internal ROM mode vector	Internal

Note that settings other than as specified above are prohibited.

● Mode register (MODR)

The data written to the mode register by mode vector fetch is called mode data. See Section "2.8.3 Reset Sequence" for information about mode vector fetch.

Once the mode register (MODR) is set, the device runs in the operating mode set according to this register.

The mode register is set by any reset source. It cannot be written from the user program.

Reference:

Nothing exists in the address ($0007FF_H$) of the mode register of the conventional FR family.

It can however be rewritten in emulator mode. In this case, use an 8-bit data transfer instruction.

It cannot be written by a 16/32-bit transfer instruction.

The details of the mode register are as follows.

[Register's Detail Explanation]

MODR

bit	7	6	5	4	3	2	1	0	Initial value
Address:$0007FD_H$	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WTH1	WTH0	$XXXXXXXX_B$
							← Operating mode →		

[bit7 to bit2] Reserved bits

Always set them to "000001_B".

Operation is not guaranteed if a value other than "000001_B" is set.

[bit1, bit0] WTH1, WTH0 (Bus width specification bits)

These bits specify the bus width for external bus mode.

In external bus mode, this value is set in the DBW0 bit in ACR0 (CS0 area).

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

CHAPTER 3

EXTERNAL BUS INTERFACE

This chapter explains the features, block diagram, I/O pins, and registers of the external bus interface.

3.1 Overview of External Bus Interface

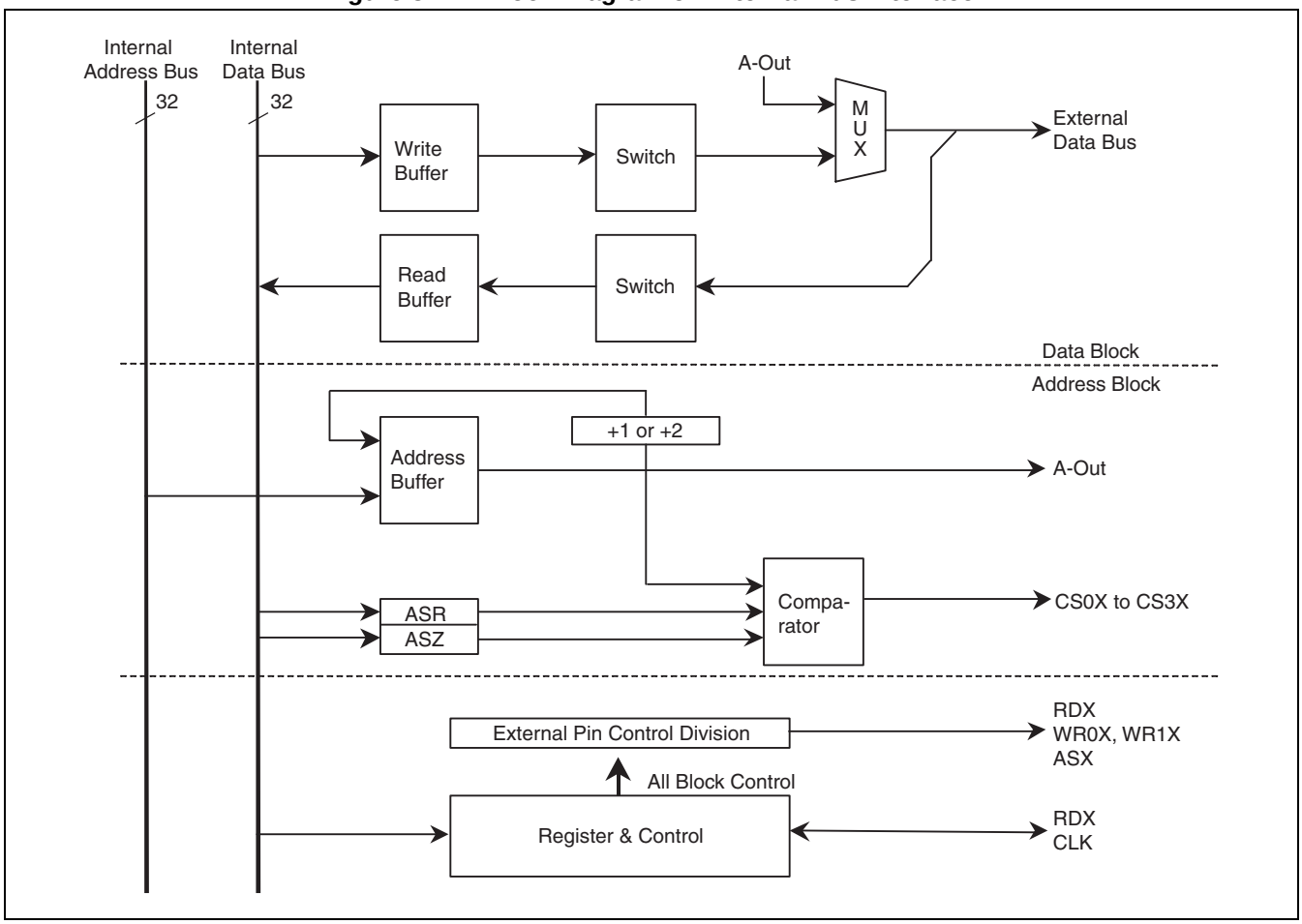
The external bus interface controller controls the interface between the LSI's internal buses and external memory and I/O devices.

■ Features

- Address/data multiplex bus (8-bit/16-bit width)
- Four independent banks (chip select areas) can be set, and chip select for each bank can be output.
  CS0X and CS1X, in units of 64K/128K/256K/512KB, can be set to the space assigned to the external bus areas up to "003FFFFF$_H$"
  CS2X and CS3X, in units of 1M/2M/4M/8MB, can be set to the space between "00400000$_H$" and "00FFFFFF$_H$".
  Boundaries may be limited depending on the size of the area.
- In each chip select area, the following functions can be set independently:
  Enabling and disabling of the chip select area (disabled areas cannot be accessed)
  Setting of the access timing type to support various kinds of memory
  Detailed access timing setting (individual setting of the access type such as the wait cycle)
  Setting of the data bus width (8-bit/16-bit)
- A different detailed timing can be set for each access timing type
  For the same type of access timing, a different setting can be made in each chip select area
  Auto-wait can be set to up to seven cycles (asynchronous SRAM, ROM, FLASH, and I/O area)
  The bus cycle can be extended by external RDY input (asynchronous SRAM, ROM, FLASH, and I/O area)
  Various kinds of idle/recovery cycles and setting delays can be inserted
- Pins that are not used by the external interface can be used as general-purpose I/O ports through settings

■ Block Diagram

Figure 3.1-1 Block Diagram of External Bus Interface



■ I/O Pins

I/O pins are external bus interface pins.

<Multiplex bus interface>

AD15 to AD00

CS0X, CS1X, CS2X, CS3X

ASX, CLK

RDX

WR0X, WR1X

RDY

■ Register List

Figure 3.1-2 Register List of External Bus Interface

Address	bit31 24	23 16	15 8	7 0
000640_H	ASR0		ACR0	
000644_H	ASR1		ACR1	
000648_H	ASR2		ACR2	
$00064C_H$	ASR3		ACR3	
000650_H	Reserved bit		Reserved bit	
000654_H	Reserved bit		Reserved bit	
000658_H	Reserved bit		Reserved bit	
$00065C_H$	Reserved bit		Reserved bit	
000660_H	AWR0		AWR1	
000664_H	AWR2		AWR3	
000668_H	Reserved bit		Reserved bit	
$00066C_H$	Reserved bit		Reserved bit	
000670_H	Reserved bit	Reserved bit	Reserved bit	Reserved bit
000674_H	Reserved bit	Reserved bit	Reserved bit	Reserved bit
000678_H	Reserved bit	Reserved bit	Reserved bit	Reserved bit
$00067C_H$	Reserved bit	Reserved bit	Reserved bit	Reserved bit
000680_H	CSER	Reserved bit	Reserved bit	Reserved bit
000684_H	Reserved bit	Reserved bit	Reserved bit	Reserved bit
000688_H	Reserved bit	Reserved bit	Reserved bit	Reserved bit
$00068C_H$	Reserved bit	Reserved bit	Reserved bit	Reserved bit
$0007F8_H$	Reserved bit	Reserved bit	Reserved bit	Reserved bit
$0007FC_H$	Reserved bit	Reserved bit	Reserved bit	Reserved bit

Reserved: Reserved register. Be sure to set "0" at write.

3.2 Explanation of Registers of External Bus Interface

This section describes the configuration and functions of the registers used for external bus interface.

■ ASR0 to ASR3 (Area Select Register)

Figure 3.2-1 Bit Configuration of ASR (Area Select Register)

ASR0										Initial value		Access
Address	bit	15	…	8	7	6	...	1	0	When INIT	When RST	
000640_H		Reserved	…	Reserved	A23	A22	...	A17	A16	0000_H	0000_H	R/W
ASR1												
Address	bit	15	…	8	7	6		1	0			
000644_H		Reserved	…	Reserved	A23	A22	...	A17	A16	$00XX_H$	$00XX_H$	R/W
ASR2												
Address	bit	15	…	8	7	6		1	0			
000648_H		Reserved	…	Reserved	A23	A22	...	A17	A16	$00XX_H$	$00XX_H$	R/W
ASR3												
Address	bit	15	…	8	7	6		1	0			
$00064C_H$		Reserved	…	Reserved	A23	A22	...	A17	A16	$00XX_H$	$00XX_H$	R/W

R/W: Readable/writable

[bit15 to bit8] Reserved: Reserved bits

Be sure to set these bits to "00_H".

[bit7 to bit0] A23 to A16

ASR0 to ASR3 (Area Select Register 0 to 3) specify the start address of each chip select area in CS0X to CS3X.

The start address can be set in the high-order 8 bits of A[23:16]. Each chip select area starts with the address set in this register and covers the range set by the ASZ[1:0] bit of the ACR0 to ACR3 registers.

The boundary of each chip select area obeys the setting of the ASZ[1:0] bit of the ACR0 to ACR3 registers. For example, if an area of 1M bytes is set by the ASZ[1:0] bit, the low-order four bits of the ASR0 to ASR3 registers are ignored and only A[23:20] bit are valid.

The ASR0 register is initialized to "00_H" by INIT and RST. ASR1 to ASR3 are not initialized by INIT and RST and are therefore undefined. After starting the LSI operation, be sure to set the corresponding ASR register before enabling each chip select area with the CSER register.

■ ACR0 to ACR3 (Area Configuration Register)

Figure 3.2-2 Bit Configuration of ACR (Area Configuration Register)

	bit	15	14	13	12	11	10	9	8	Initial value When INIT	Initial value When RST	Access
ACR0H												
000642_H		Reserved	Reserved	ASZ1	ASZ0	Reserved	DBW0	Reserved	Reserved	$00110X00_B$	$00110X00_B$	R/W
ACR0L												
	bit	7	6	5	4	3	2	1	0			
000643_H		Reserved	Reserved	WREN	Reserved	TYP3	TYP2	TYP1	TYP0	00000000_B	00000000_B	R/W
ACR1H												
	bit	15	14	13	12	11	10	9	8			
000646_H		Reserved	Reserved	ASZ1	ASZ0	Reserved	DBW0	Reserved	Reserved	$00XX0X00_B$	$00XX0X00_B$	R/W
ACR1L												
	bit	7	6	5	4	3	2	1	0			
000647_H		Reserved	Reserved	WREN	Reserved	TYP3	TYP2	Reserved	Reserved	$00X0XXXX_B$	$00X0XXXX_B$	R/W
ACR2H												
	bit	15	14	13	12	11	10	9	8			
$00064A_H$		Reserved	Reserved	ASZ1	ASZ0	Reserved	DBW0	Reserved	Reserved	$00XX0X00_B$	$00XX0X00_B$	R/W
ACR2L												
	bit	7	6	5	4	3	2	1	0			
$00064B_H$		Reserved	Reserved	WREN	Reserved	TYP3	TYP2	TYP1	TYP0	$00X0XXXX_B$	$00X0XXXX_B$	R/W
ACR3H												
	bit	15	14	13	12	11	10	9	8			
$00064E_H$		Reserved	Reserved	ASZ1	ASZ0	Reserved	DBW0	Reserved	Reserved	$00XX0X00_B$	$00XX0X00_B$	R/W
ACR3L												
	bit	7	6	5	4	3	2	1	0			
$00064F_H$		Reserved	Reserved	WREN	Reserved	TYP3	TYP2	TYP1	TYP0	$00X0XXXX_B$	$00X0XXXX_B$	R/W

R/W: Readable/Writable

ACR0 to ACR3 (Area Configuration Register 0 to 3) set the functions of each chip select area.

[bit15, bit14] Reserved: Reserved bits

Be sure to set these bits to "00_B".

[bit13, bit12] ASZ1, ASZ0 = Area Size Bit [1:0]

The size of each chip select area is set as follows:

Register	ASZ1	ASZ0	Size of Each Chip Select Area	Setting
ASR0/ ASR1	0	0	64 KB (00010000_H byte, ASR A[23:16] bit is valid)	Only CS0X and CS1X are valid
	0	1	128 KB (00020000_H byte, ASR A[23:17] bit is valid)	
	1	0	256 KB (00040000_H byte, ASR A[23:18] bit is valid)	
	1	1	512 KB (00080000_H byte, ASR A [23:19] bit is valid)	
ASR2/ ASR3	0	0	1 MB (00100000_H byte, ASR A[23:20] bit is valid)	Only CS2X and CS3X are valid
	0	1	2 MB (00200000_H byte, ASR A[23:21] bit is valid)	
	1	0	4 MB (00400000_H byte, ASR A[23:22] bit is valid)	
	1	1	8 MB (00800000_H byte, ASR A[23] bit is valid)	

ASZ[1:0] are used to set the size of each area by modifying the number of bits for address comparison to a value different from ASR. Thus, an ASR contains bits that are not compared.

The ASZ[1:0] bit of ACR0 are initialized to "11_B" by RST. Despite this setting, however, the CS0 area just after RST is executed is specially set from "00000000_H" to "$00FFFFFF_H$" (setting of entire area). The entire-area setting is reset after the first write to ACR0 and an appropriate size is set as indicated in the table shown above.

[bit11] Reserved: Reserved bit

Be sure to set this bit to "0".

[bit10] DBW0 = Data Bus Width[0]

Data bus width of each chip select area is set as follows:

DBW0	Data Bus Width
0	8 Bit (Byte Access)
1	16 Bit (Half Word Access)

The same values as those of the WTH bits of the mode vector are written automatically to bits DBW0 of ACR0 during the reset sequence.

[bit9 to bit6] Reserved: Reserved bits

Be sure to set these bits to "0000_B".

[bit5] WREN = WRite ENable

This bit sets enabling and disabling of writing to each chip select area.

WREN	Write enable/disable
0	Disable write
1	Enable write

If an area for which write operations are disabled is accessed for a write operation from the internal bus, the access is ignored and no external access at all is performed.

Set the WREN bit of areas for which write operations are required, such as data areas, to "1".

[bit4] Reserved: Reserved bit

Be sure to set this bit to "0".

[bit3 to bit0] TYP3 to TYP0 = TYPe select

Access type of each chip select area is set as follows:

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM, I/O, single ROM/FLASH)
	1	x	x	Address data multiplex access (8/16-bit bus width only)
	x	x	0	Disable WAIT insertion by the RDY pin
		x	1	Enable WAIT insertion by the RDY pin
		0	x	Use the WR0X and WR1X pins as write strobes
		1	x	Setting disabled
1	0	0	0	Setting disabled
			1	Setting disabled
	0	1	0	Setting disabled
	0	1	1	Setting disabled
	1	0	0	Setting disabled
	1	0	1	Setting disabled
	1	1	0	Setting disabled
	1	1	1	Mask area setting (The access type is the same as that of the overlapping area) *1

Set the access type as the combination of all bits.

See the section of operation explanation for details on how each access type works.

*1: CS area mask setting function

If you want to set an area some of whose operation settings are changed for a certain CS area (referred to as the base setting area), you can set ACR:TYP[3:0]=1111 in another CS area so that the area can function as a mask setting area.

If you do not use the mask setting function, disable any overlapping area settings for multiple CS areas.

Access operations to the mask setting area are as follows:

-CSnX corresponding to a mask setting area is not asserted.

-CSnX corresponding to a base setting area is asserted.

-For the following ACR settings, the settings on the mask setting area side are valid:
Bit [10] DBW0 : Bus width setting
Bit [5] WREN : Write-enable setting *
*:For this setting only, a setting that is different from that of the base setting area is not allowed.

-For the following ACR setting, the setting on the base setting area side is valid:
Bit [3:0] TYP[3:0] : Access type setting

-For the AWR settings, the settings on the mask setting area side are valid.

A mask setting area can be set for only part of another CS area (base setting area). You cannot set a mask setting area for an area without a base setting area. Do not overlap multiple mask setting areas. Use care when setting ASR and the ACR:ASZ[1:0] bit.

(Restrictions)

A write-enable setting cannot be implemented by a mask.

Write-enable settings in the base CS area and the mask setting area must be identical.

If write operations to a mask setting area are disabled, the area is not masked and operates as a base CS area.

If write operations to the base CS area are disabled but are enabled to the mask setting area, the area has no base, resulting in malfunctions.

■ AWR0 to AWR3 (Area Wait Register)

Figure 3.2-3 Bit Configuration of AWR (Area Wait Register)

										Initial value		
AWR0H Address	bit	15	14	13	12	11	10	9	8	When INIT	When RST	Access
000660_H		Reserved	W14	W13	W12	Reserved	Reserved	Reserved	Reserved	01110000_B	01110000_B	R/W
AWR0L Address	bit	7	6	5	4	3	2	1	0			
000661_H		Reserved	W06	Reserved	W04	Reserved	W02	W01	W00	01011011_B	01011011_B	R/W
AWR1H Address	bit	15	14	13	12	11	10	9	8			
000662_H		Reserved	W14	W13	W12	Reserved	Reserved	Reserved	Reserved	$0XXX0000_B$	$0XXX0000_B$	R/W
AWR1L Address	bit	7	6	5	4	3	2	1	0			
000663_H		Reserved	W06	Reserved	W04	Reserved	W02	W01	W00	$0X0X1XXX_B$	$0X0X1XXX_B$	R/W
AWR2H Address	bit	15	14	13	12	11	10	9	8			
000664_H		Reserved	W14	W13	W12	Reserved	Reserved	Reserved	Reserved	$0XXX0000_B$	$0XXX0000_B$	R/W
AWR2L Address	bit	7	6	5	4	3	2	1	0			
000665_H		Reserved	W06	Reserved	W04	Reserved	W02	W01	W00	$0X0X1XXX_B$	$0X0X1XXX_B$	R/W
AWR3H Address	bit	15	14	13	12	11	10	9	8			
000666_H		Reserved	W14	W13	W12	Reserved	Reserved	Reserved	Reserved	$0XXX0000_B$	$0XXX0000_B$	R/W
AWR3L Address	bit	7	6	5	4	3	2	1	0			
000667_H		Reserved	W06	Reserved	W04	Reserved	W02	W01	W00	$0X0X1XXX_B$	$0X0X1XXX_B$	R/W

R/W: Readable/writable

AWR0 to AWR3 specify various kinds of wait timing for each chip select area.

The function of each bit changes according to the access type (TYP[3:0] bit) setting of the ACR0 to ACR3 registers.

■ Normal Access or a Address/Data Multiplex Access Operation

A chip select area determined by either of the following settings for the access type (TYP[3:0] bit) of ACR0 to ACR3 registers becomes the area for normal access or a address/data multiplex access operation.

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM, I/O, single ROM/FLASH)
0	1	x	x	Address data multiplex access (8/16-bit bus width only)

The following lists the functions of each AWR0 to AWR3 bit for a normal access or address/data multiplex access area. Since the initial values of registers other than AWR0 are undefined, set them to their initial values before enabling each area with the CSER register.

[bit15] Reserved: Reserved bit

Be sure to set this bit to "0".

[Bit 14 to Bit 12] W14 to W12 = First Access Wait Cycle

These bits set the number of auto-wait cycles to be inserted into the first access cycle of each cycle.

The initial value of the CS0 area is set to 7 (wait). The initial values of other areas are undefined.

W14	W13	W12	First access wait cycle
0	0	0	Auto-wait cycle 0
0	0	1	Auto-wait cycle 1
…			…
1	1	0	Auto-wait cycle 6
1	1	1	Auto-wait cycle 7

[bit11 to bit7] Reserved: Reserved bits

Always set them to "00000_B".

[bit6] W06 = Read → Write Idle Cycle

The read → write idle cycle is set to prevent collision of read data and write data on the data bus when a write cycle follows a read cycle. During an idle cycle, all chip select signals are negated and the data terminals maintain the high impedance state.

If a write cycle follows a read cycle or an access operation to another chip select area occurs after a read cycle, the specified idle cycle is inserted.

W06	Read → write idle cycles
0	0 cycle
1	1 cycle

[bit5] Reserved: Reserved bit

Be sure to set the bit to "0".

[bit4] W04 = Write Recovery Cycle

The write recovery cycle is set to control the access to a device in which the interval of access subsequently after write access is limited. During a write recovery cycle, all chip select signals are negated and the data terminals maintain the high impedance state.

If the write recovery cycle is set to "1" or more, a write recovery cycle is always inserted after write access.

W04	Write recovery cycles
0	0 cycle
1	1 cycle

[bit3] Reserved: Reserved bit

Be sure to set the bit to "1".

[bit2] W02 = Address → CSX Delay

The address → CSX delay setting is made when a certain type of setup is required for the address when CSX falls or CSX edges are needed for successive accesses to the same chip select area.

Set the address and set the delay from ASX output to CS0X to CS3X output.

W02	Address → CSX delay
0	No delay
1	Delay

If no delay is selected by setting "0", assertion of CS0X to CS3X starts at the same timing that ASX is asserted. If successive accesses are made to the same chip select area at this time, assertion of CS0 to CS3X may continue without a change between two access operations.

If delay is specified by selecting "1", assertion of CS0X to CS3X starts when the external memory clock SYSCLK output rises. If successive accesses are made to the same chip select area at this point, CS0X to CS3X are negated at a timing between two access operations.

If CSX delay is selected, one setup cycle is inserted before asserting the read/write strobe after assertion of the delayed CSX (operation is the same as the CSX → RDX/WRX setup setting of W01).

[bit1] W01 = CSnX → RDX/WRX Setup Delay Cycle

The CSnX → RDX/WRX setup extension cycle is set to extend the period before the read/write strobe is asserted after CSnX is asserted. At least one setup extension cycle is inserted before the read/write strobe is asserted after CSnX is asserted.

W01	CSnX → RDX/WRX setup delay cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting "0", RDX/WR0X and WR1X are output at the earliest when the external memory clock SYSCLK output rises just after CSn is asserted. WR0X and WR1X may be delayed 1 cycle or more depending on the internal bus state.

If 1 cycle is selected by setting "1", RDX/WR0X and WR1X are always output 1 cycle or more later.

When successive accesses are made within the same chip select area without negating CSnX, a setup extension cycle is not inserted. If a setup extension cycle for determining the address is required, set the W02 bit and insert the address → CSnX delay. Since CSnX is negated for each access operation, the setup extension cycle is enabled.

If the CSnX delay set by W02 is inserted, this setup cycle is always enabled regardless of the setting of the W01 bit.

[bit0] W00 = RDX/WRX → CSnX Hold Extension Cycle

The RDX/WRX → CSnX hold extension cycle is set to extend the period before negating CSnX after the read/write strobe is negated. One hold extension cycle is inserted before CSnX is negated after the read/write strobe is negated.

W00	RDX/WRX → CSnX hold extension cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting "0", CS0X to CS3X are negated after the hold delay from the rising edge of external memory clock SYSCLK output after RDX/WR0X and WR1X are negated.

If 1 cycle is selected by setting "1", CS0X to CS3X are negated one cycle later.

When making successive accesses within the same chip select area without negating CSnX, the hold extension cycle is not inserted. If a hold extension cycle for determining the address is required, set the W02 bit and insert the address → CSnX delay. Since CSnX is negated for each access operation, the hold extension cycle is enabled.

■ CSER (Chip Select Enable Register)

Figure 3.2-4 Bit Configuration of CSER (Chip Select Enable register)

Address	bit	31	30	29	28	27	26	25	24	Initial value When INIT	Initial value When RST	Access
000680_H		Reserved	Reserved	Reserved	Reserved	CSE3	CSE2	CSE1	CSE0	00000001_B	00000001_B	R/W

The chip select enable register enables and disables each chip select area.

[bit31 to bit28] Reserved: Reserved bits

Be sure to set these bits to "0000_B".

[bit27 to bit24] CSE3 to CSE0 = Chip Select Area Enable (Chip Select Enable 0 to 3)

These bits are the chip select area enable bits for CS0X to CS3X.

The initial value is "0001_B", which enables only the CS0 area.

When "1" is written, a chip select area operates according to the settings of ASR0 to ASR3, ACR0 to ACR3, and AWR0 to AWR3.

Before setting this register, be sure to make all settings required for the corresponding chip select areas.

CSE3 to CSE0	Area control
0	Disable
1	Enable

CSE bit	Corresponding CSX
Bit [24]: CSE0	CS0X
Bit [25]: CSE1	CS1X
Bit [26]: CSE2	CS2X
Bit [27]: CSE3	CS3X

3.3 Chip Select Area

In the external bus interface, a total of four chip select areas can be set.

■ Example of Chip Select Area Setting

The address space for each area is in ASR0 to ASR3 (Area Select Register) and ACR0 to ACR3 (Area Configuration Register). CS0X and CS1X can be set in the space assigned to external bus areas between "00000000H" and "003FFFFFH" in units of 64K/128K/256K/512KB. CS2X and CS3X can be set in the space between "00400000H" to "00FFFFFFH" in units of 1M/2M/4M/8MB.

When bus access is made to an area specified by these registers, the corresponding chip select signals, CS2X and CS3X, are activated ("L" output) during the access cycle.

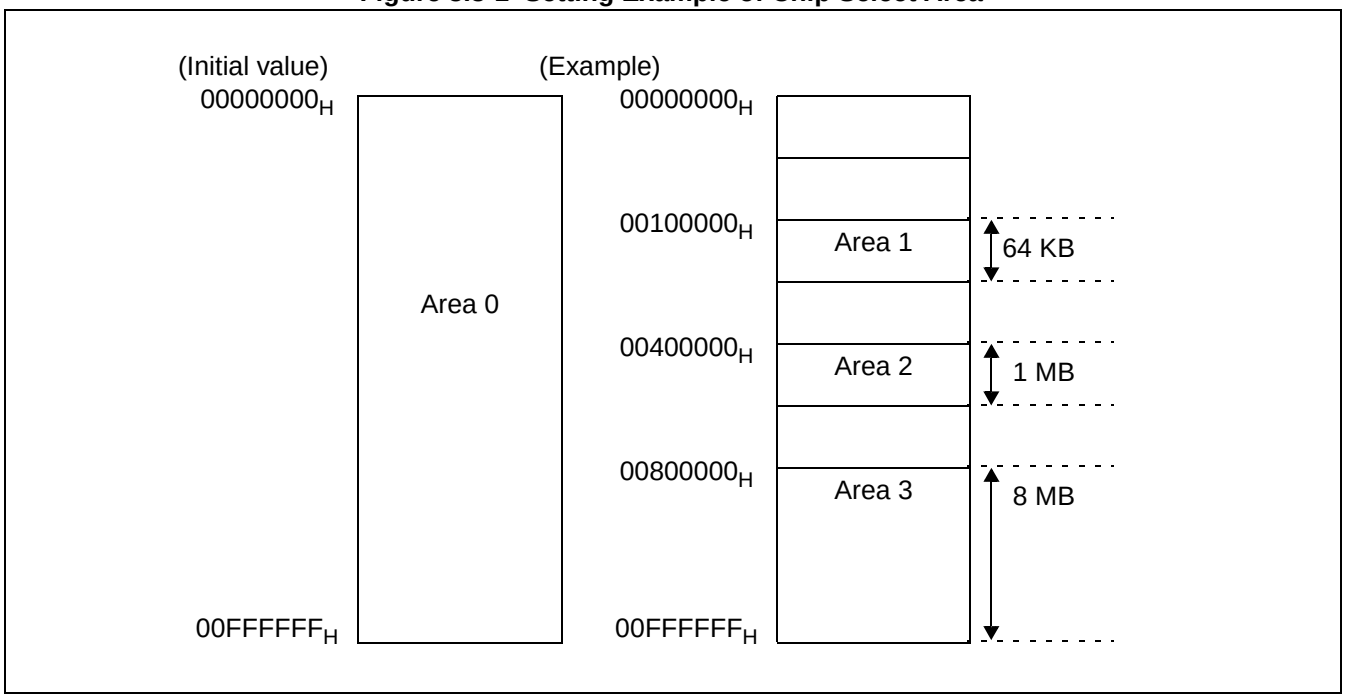
● Example of ASR and ASZ[1:0] settings

1. ASR1=0010_H ACR1 → ASZ[1:0]=00_B Chip select area 1 is assigned to "00100000_H" to "$0010FFFF_H$".
2. ASR2=0040_H ACR2 → ASZ[1:0]=00_B Chip select area 2 is assigned to "00400000_H" to "$004FFFFF_H$".
3. ASR3=0081_H ACR3 → ASZ[1:0]=11_B Chip select area 3 is assigned to "00800000_H" to "$00FFFFFF_H$".
   Since 8 MB is set for ACR → ASZ[1:0] at this point, the unit for boundaries is 8 MB and ASR3[22:16] are ignored.

Before there is any writing to ACR0 after a reset, "00000000_H" to "$00FFFFFF_H$" is assigned to the chip select area 0.

Note: Set the chip select areas so that there is no overlap.

Figure 3.3-1 Setting Example of Chip Select Area

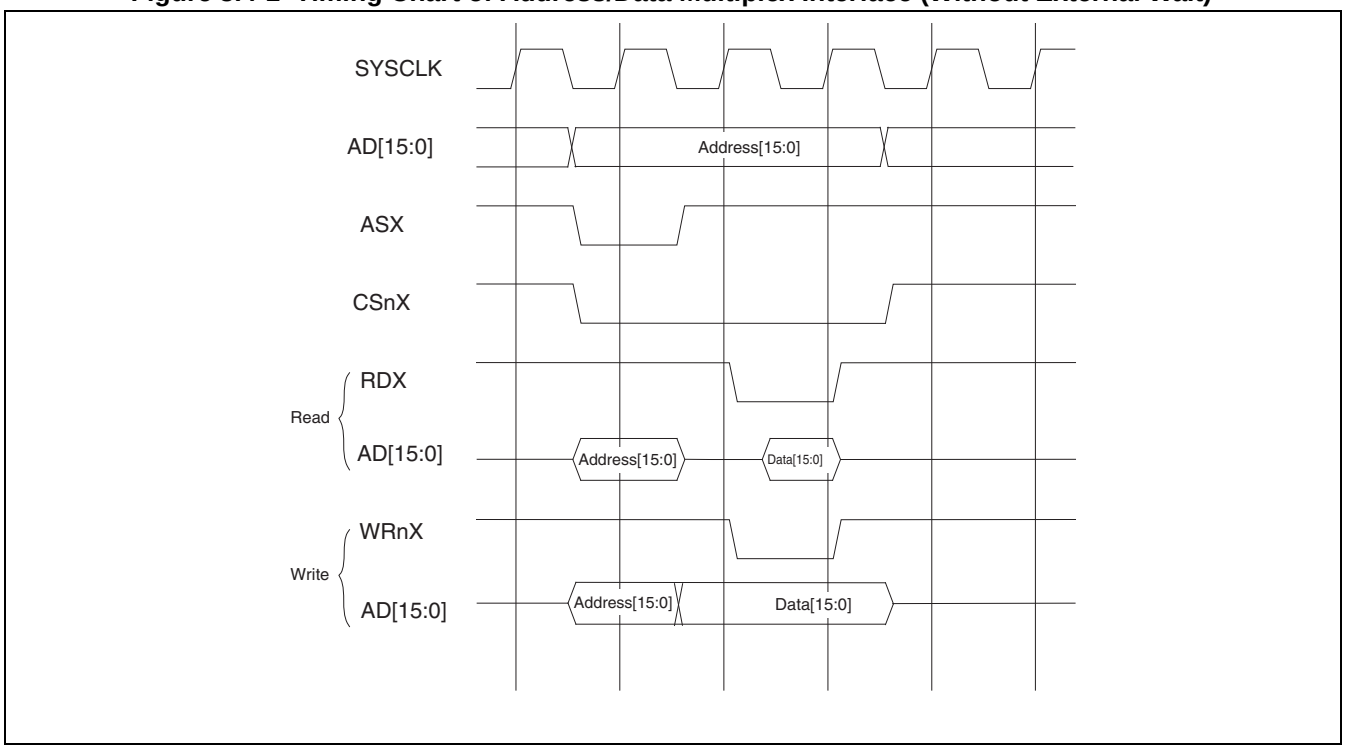


3.4 Address/Data Multiplex Interface

This section indicates the timing of operation of the address/data multiplex interface.

■ Without External Wait (TYP[3:0]=0100_B, AWR=0008_H)

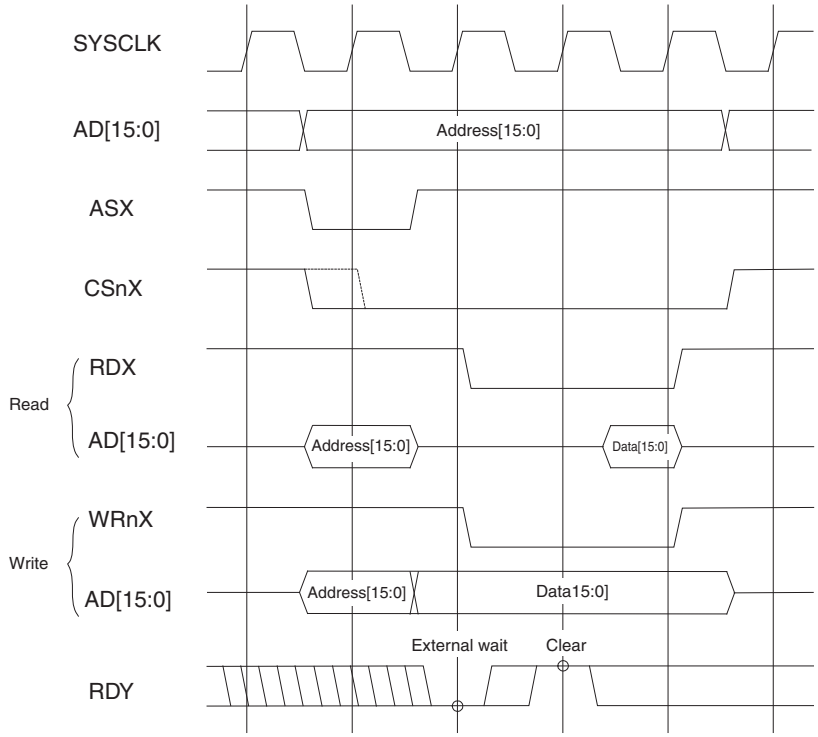
Figure 3.4-1 Timing Chart of Address/Data Multiplex Interface (Without External Wait)



- Making a setting such as TYP[3:0]=$01xx_B$ in the ACR register enables the address/data multiplex interface to be set.
- If the address/data multiplex interface is set, set 8 bits or 16 bits for the data bus width (DBW0 bit).
- In the address/data multiplex interface, the total of 3 cycles of 2 address output cycles + 1 data cycle becomes the basic number of access cycles.
- In the address output cycles, ASX is asserted as the output address latch signal. However, when the CSnX → RDX/WRnX setup delay (AWR:W01) is set to "0", the multiplex address output cycle consists of only one cycle as shown in the figure above. Since the address cannot be directly latched at the rising edge of ASX, fetch the address at the rising edge of MCLK of the cycle in which "L" is asserted for ASX. When the address is directly latched at the rising edge of ASX, see "Setting of CSnX → RDX/WRnX setup".
- As with a normal interface, the address indicating the start of access is output to AD[15:0] during the time division bus cycle. Use this address if you want to use an address more than 8/16 bits in the address/data multiplex interface.
- As with the normal interface, auto-wait (AWR:W14 to W12), read → write idle cycle (AWR:W06), write recovery (AWR:W04), address → CSnX delay (AWR:W02), CSnX → RDX/WRX setup delay (AWR:W01), and RDX/WRX → CSnX hold delay (AWR:W00) can be set.

■ With External Wait (TYP[3:0]=0101_B, AWR=1008_H)

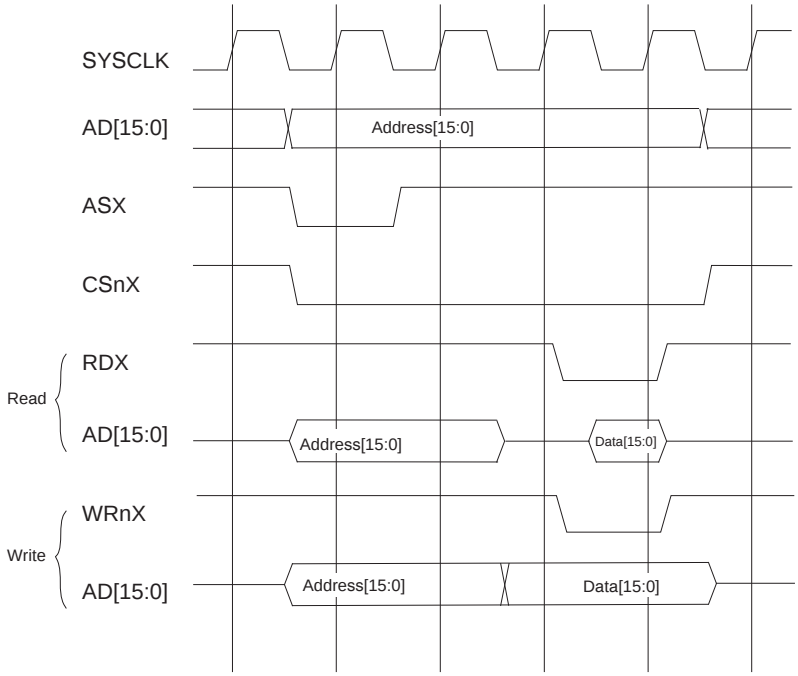
Figure 3.4-2 Timing Chart of Address/Data Multiplex Interface (With External Wait)



Making a setting such as TYP[3:0]=$01x1_B$ in the ACR register enables RDY input in the address/data multiplex interface.

■ Setting of CSX → RDX/WRX Setup (TYP[3:0]=0101_B, AWR=$100B_H$)

Figure 3.4-3 Timing Chart of Address/Data Multiplex Interface (Setting of CSX → RDX/WRX Setup)



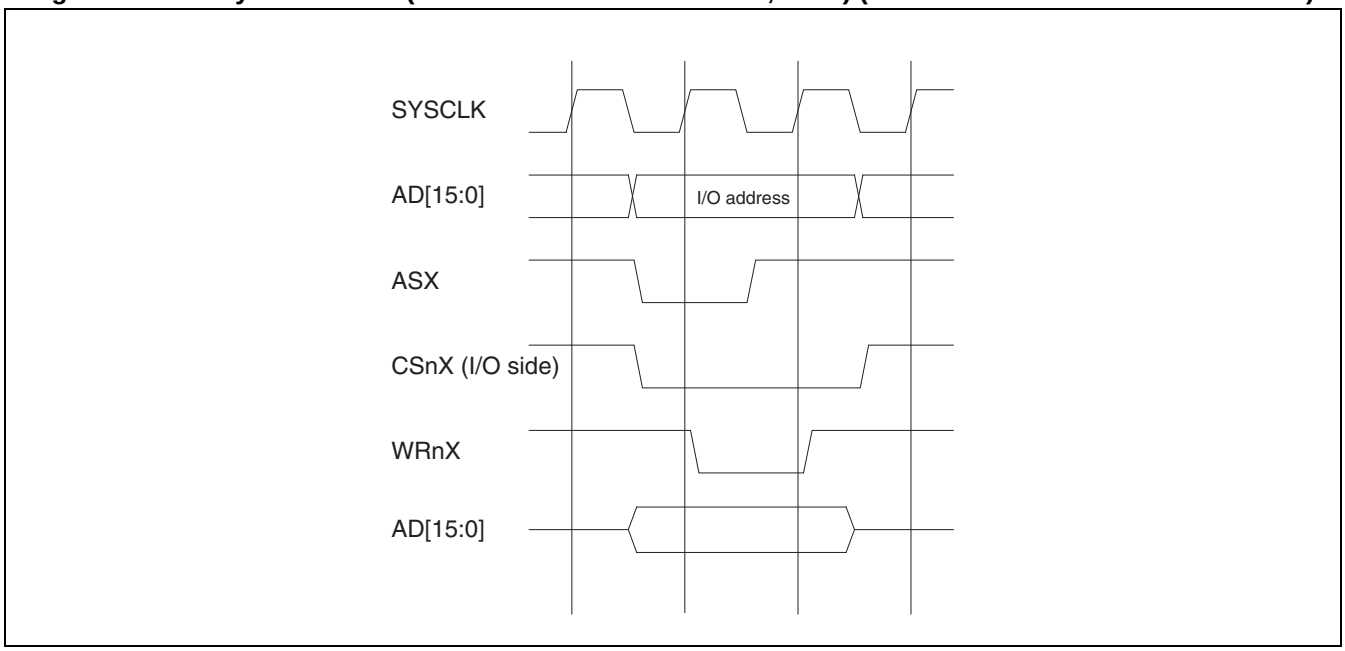
Setting "1" for the CSnX → RDX/WRnX setup delay (AWR:W01) enables the multiplex address output cycle to be extended by 1 cycle as shown in the figure above, allowing the address to be latched directly to the rising edge of ASX. Use this setting if you want to use ASX as an ALE (Address Latch Enable) strobe without using SYSCLK.

3.5 DMA Access

This section indicates the timing of operation of the DMA access.

■ 2-Cycle Transfer (The Timing is the Same as for Internal RAM → External I/O, RAM, External I/O, RAM → Internal RAM.) (TYP[3:0]=0000_B, AWR=0008_H)

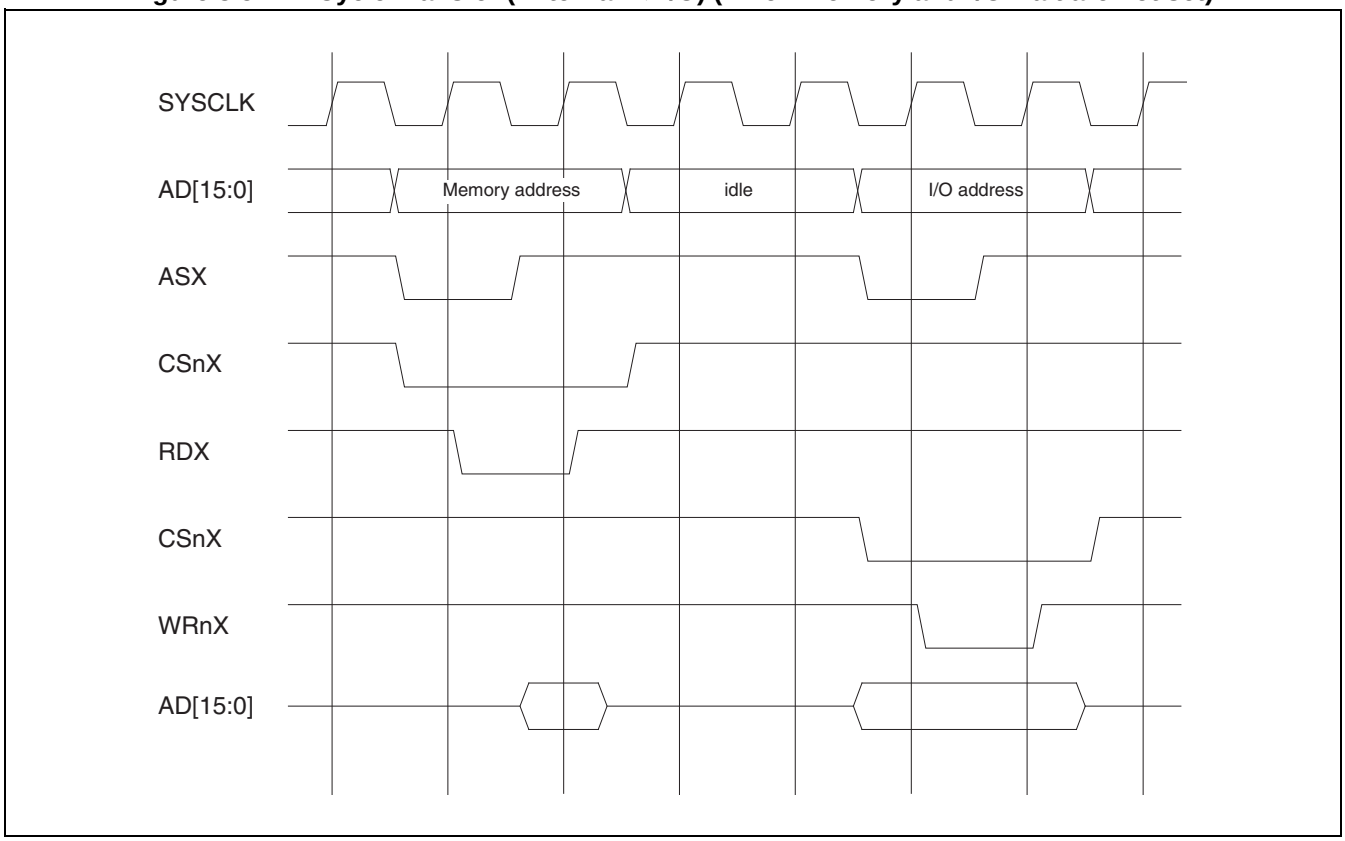
Figure 3.5-1 2-Cycle Transfer (Internal RAM → External I/O, RAM) (When no wait is set on the I/O side)



The bus access is the same as that of the interface for non-DMAC transfer.

■ 2-Cycle Transfer (External → I/O) (TYP[3:0]=0000_B, AWR=0008_H)

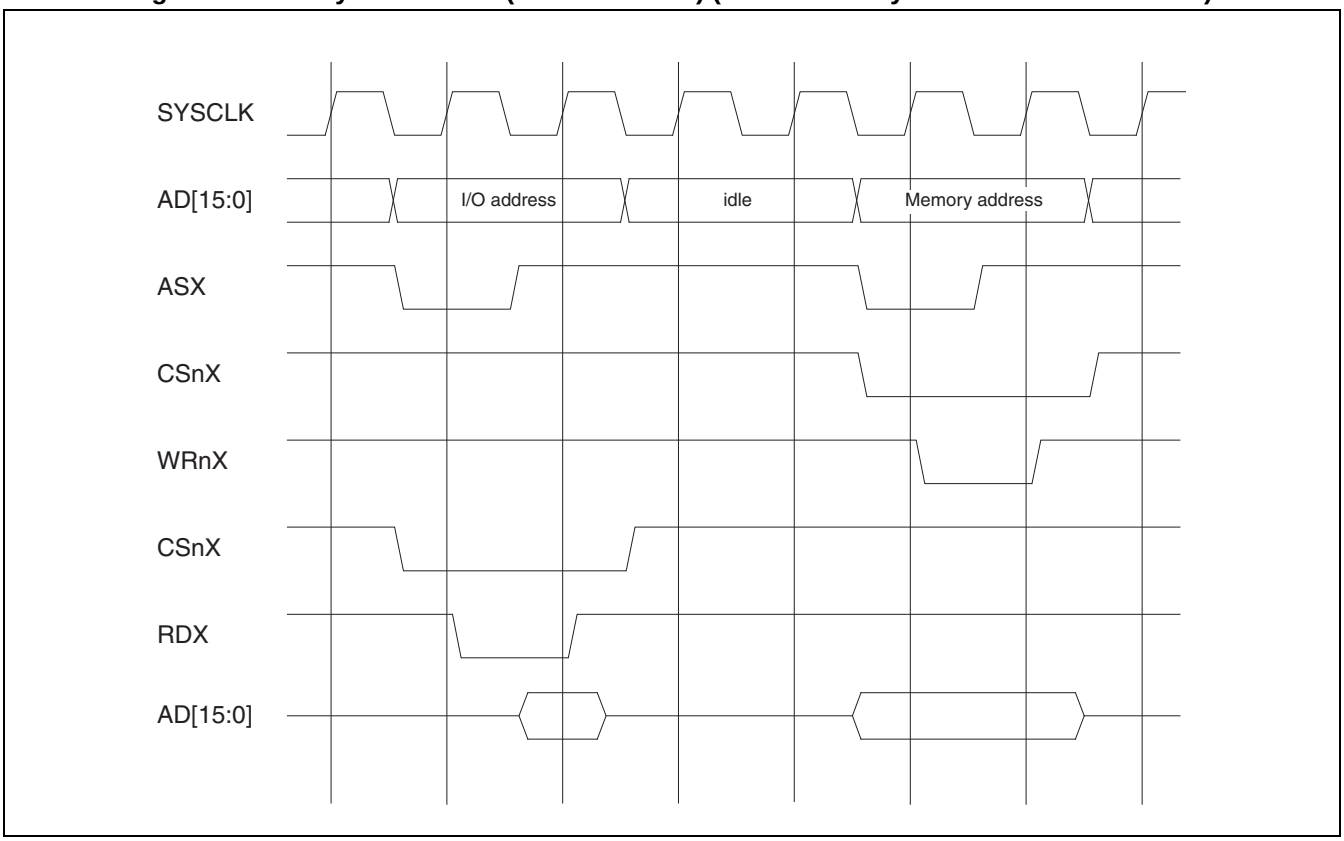
Figure 3.5-2 2-Cycle Transfer (External → I/O) (When memory and I/O wait are not set)



The bus access is the same as that of the interface for non-DMAC transfer.

■ 2-Cycle Transfer (I/O → External) (TYP[3:0]=0000_B, AWR=0008_H)

Figure 3.5-3 2-Cycle Transfer (I/O → External) (When memory and I/O wait are not set)



The bus access is the same as that of the interface for non-DMAC transfer.

MB91313A Series

3.6 Register Setting Procedure

This section explains the procedure of the register used in the external bus interface.

■ Register Setting Procedure

For setting procedure concerning with external bus interface, follow the principle described below.

1. Before rewriting the contents of a register, be sure to set the CSER register so that the corresponding area is set "0".
   If you change the settings while "1" is set, access before and after the change cannot be guaranteed.
2. Use the following procedure to change a register:
   (1) Set "0" for the CSER bit corresponding to the applicable area.
   (2) Set both ASR and ACR at the same time using word access.
   When accessing ASR and ACR using half word, set ACR after setting ASR.
   (3) Set AWR.
   (4) Set the CSER bit corresponding to the applicable area.
3. The CS0X area is enabled after a reset is released. If the area is used as a program area, the register contents need to be rewritten while the CSER bit is "1". In this case, make the settings described in (2) to (3) above in the initial state with a low-speed internal clock. Then, switch the clock to a high-speed clock.

3.7 Note on Using External Bus Interface

This section explains the note on using the external bus interface.

■ Notes on Using External Bus Interface

To mix the area where WR0X and WR1X are used as write strobes (TYP[3:0]=0X0X$_B$) and area where WRnX is used as write strobe (TYP[3:0]=0X1X$_B$), be sure to set the following items for all the areas to be used.

(1) Set one or more read → write idle cycles (AWR:W06=1).

(2) Set one or more write recovery cycles (AWR:W04=1).

These restrictions do not apply if the area where WR0X and WR1X are used as write strobes (TYP[3:0]=0X0X$_B$) is specified but WR0X and WR1X is unavailable (only ROM is connected). The restrictions do not apply if the area where WRX is used as write strobe (TYP[3:0]=0X1X$_B$) is specified and both address → RDX/WRX setup cycle (W01=1) and RDX/WRX → address hold cycle (W00=1) are set.

<Reason for the Restrictions>

In the area where TYP[3:0]=0X1X$_B$ is set and WRnX is used as write strobe, the WR0X/WR1X pins are in form of the byte enable (UBX/LBX) output. In this case, the byte enable output generates the enable signal for each byte location at the same timing as the address and CSX output.

Therefore, if an access is made to an area where the WR0X/WR1X pins are used as asynchronous write strobes before or after the operation, the area may not satisfy the AC standard between CSX and WR0X/WR1X, resulting into an error writing.

If a read → write idle cycle and write recovery cycle are set, CSX is not asserted (the "H" level is maintained) in these cycles, so the AC standard is satisfied.

The restriction does not apply if there is a space for the AC standard (setup and hold) between CSX and WR0X/WR1X in the area where the WR0X/WR1X pins are used as asynchronous write strobes.

CHAPTER 4
I/O PORTS

This chapter gives an overview of I/O ports and describes their register configuration and functions.

4.1 Overview of I/O Ports

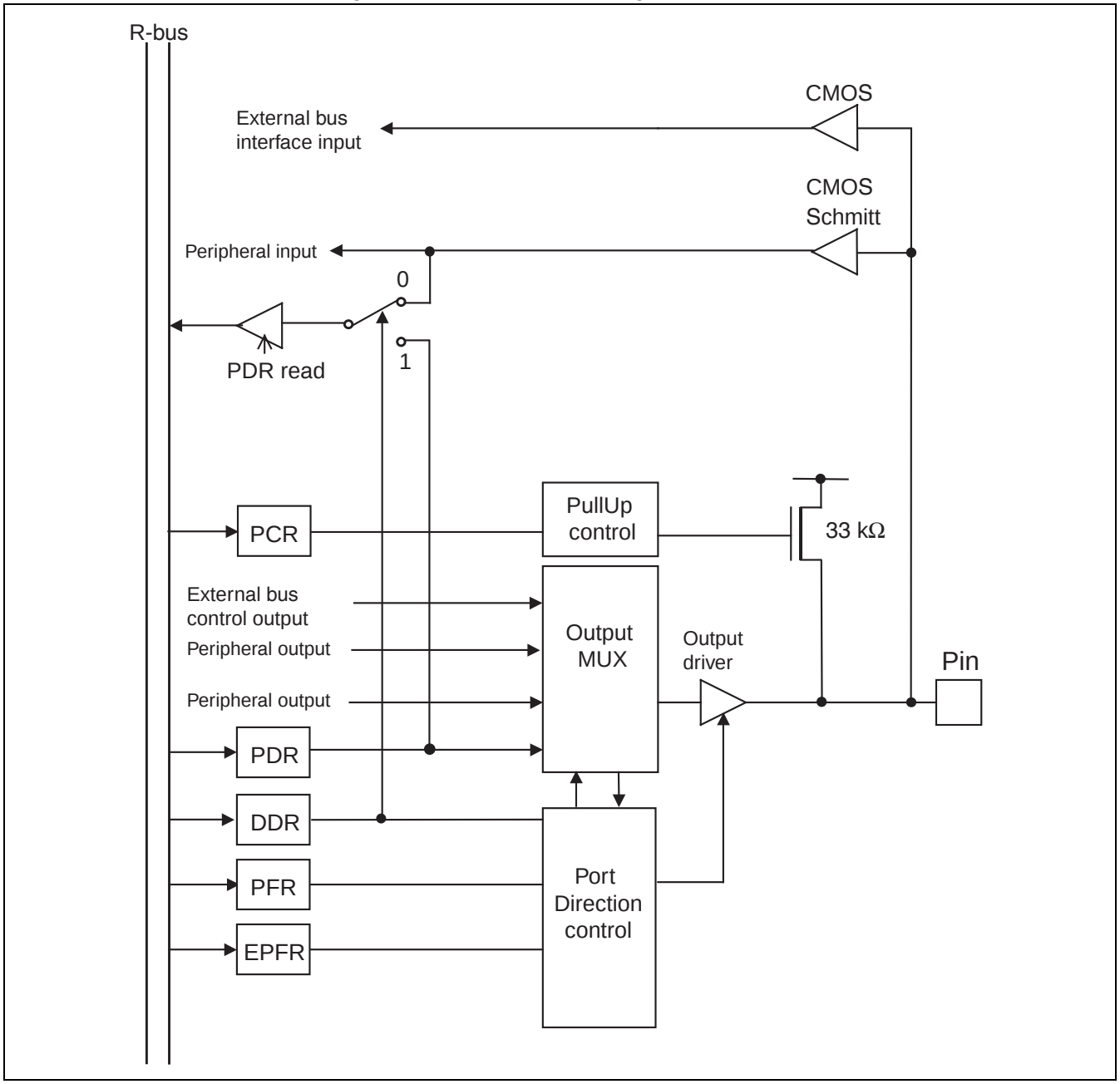
This section gives an overview of the I/O ports of the MB91313A series.

■ Basic Block Diagram of Ports

It can be used as an I/O port if the settings are made so that the external bus interfaces or peripherals corresponding to the pins do not use the pins for input/output.

Figure 4.1-1 shows a basic block diagram of ports.

Figure 4.1-1 Basic Block Diagrams of Ports



■ General Specifications of Ports

- Each port has a port data register (PDR) to store output data. A reset does not initialize the PDR register.
- Each port has a data direction register (DDR) to switch its data direction between input and output. A reset switches the data direction of all ports to input ($DDR=00_H$).
  - Port input mode (PFR=0 & EPFR=0 & DDR=0)

    PDR read :Reads the level at the corresponding external pin.
    PDR write:Writes a set value to the port data register.
  - Port output mode (PFR=0 & EPFR=0 & DDR=1)

    PDR read :Reads a value from the port data register.
    PDR write:Writes a set value to the port data register to output it to the corresponding external pin.
  - Peripheral output mode (other settings than PFR=0 & EPFR=0)

    PDR read :Reads the output value from the peripheral when DDR=0.
    Reads a value from the port data register when DDR=1.
    PDR write:Writes a set value to the port data register.
  - The read-modify-write (RMW) instruction for a port data register reads the register set value irrespective of the state of the corresponding port.
  - Barring some unique circumstance, the input to a peripheral is always connected to a pin. Usually, use the port input mode for input to a peripheral.
- Each port has the pull-up control register, and can set the pull-up of 33 kΩ per pin.
- The ports have the port function register (PFR), and some have the extended port function register (EPFR). They mainly control the peripheral output.
- In the external bus mode, the pin assigned to the external bus interface invalidates the setting of DDR and PFR, and the function of the bus interface is prioritized. When these pins are used as the general-purpose port/peripheral output in the external bus mode, set the EPFR and disable the function of the bus interface.
- In STOP mode, input is fixed to "0". Note, however, that the external interrupt inputs to their respective pins are not fixed and can be used as interrupts when the interrupts are enabled (with the ENIR bit set and the PFR/EPFR/DDR register selecting the input pins).
- Two-way signals of peripherals (the I^2C function SOT and SCK of UART) are enabled by the PFR register. For switching between input and output, see the chapter for the relevant peripheral.

Notes:

- If using the pins as an input resource, make sure that PFR=0 and DDR=0 (port input mode).
- There is no register switched between general-purpose port input and peripheral input. The value input via an external pin is always passed to the general-purpose port and peripheral circuit.
  Even with the DDR output setting, the value output to the outside is always propagated to the general-purpose port and peripheral circuit.
  For use as a peripheral input, use PER, EPER and DDR input and enable each peripheral's input signal.

4.2 Settings of Port Data Registers

This section explains the configuration and functions of the port data register.

Each port has a port data register (PDR) to store output data. A reset does not initialize the PDR register.

■ Configuration of Port Data Register (PDR)

The following figure shows the configuration of the port data register (PDR).

Figure 4.2-1 Configuration of Port Data Register (PDR)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PDR0	000000_H		PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00	$XXXXXXXX_B$
PDR1	000001_H		PDR17	PDR16	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	$XXXXXXXX_B$
PDR2	000002_H		PDR27	PDR26	PDR25	PDR24	PDR23	PDR22	PDR21	PDR20	$XXXXXXXX_B$
PDR3	000003_H		PDR37	PDR36	PDR35	PDR34	PDR33	PDR32	PDR31	PDR30	$XXXXXXXX_B$
PDR4	000004_H		PDR47	PDR46	PDR45	PDR44	PDR43	PDR42	PDR41	PDR40	$XXXXXXXX_B$
PDR5	000005_H		PDR57	PDR56	PDR55	PDR54	PDR53	PDR52	PDR51	PDR50	$XXXXXXXX_B$
PDR6	000006_H		-	-	PDR65	PDR64	PDR63	PDR62	PDR61	PDR60	$--XXXXXX_B$
PDRC	$00000C_H$		PDRC7	PDRC6	PDRC5	PDRC4	PDRC3	PDRC2	PDRC1	PDRC0	$XXXXXXXX_B$
PDRD	$00000D_H$		PDRD7	PDRD6	PDRD5	PDRD4	PDRD3	PDRD2	PDRD1	PDRD0	$XXXXXXXX_B$
PDRE	$00000E_H$		PDRE7	PDRE6	PDRE5	PDRE4	PDRE3	PDRE2	PDRE1	PDRE0	$XXXXXXXX_B$
PDRF	$00000F_H$		PDRF7	PDRF6	PDRF5	PDRF4	PDRF3	PDRF2	PDRF1	PDRF0	$XXXXXXXX_B$
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PDR0 to PDR6, PDRC to PDRF are the I/O data registers for the I/O port.

Input/output is controlled by the corresponding DDR0 to DDR6 and DDRC to DDRF.

The read-modify-write (RMW) instruction for a port data register reads the register set value irrespective of the state of the corresponding port.

4.3 Settings of Data Direction Registers

This section explains the configuration and functions of the data direction register.

■ Configuration of Data Direction Register

The following figure shows the configuration of the data direction register (DDR).

Figure 4.3-1 Configuration of Data Direction Register (DDR)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
DDR0	000400_H		DDR07	DDR06	DDR05	DDR04	DDR03	DDR02	DDR01	DDR00	00000000_B
DDR1	000401_H		DDR17	DDR16	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10	00000000_B
DDR2	000402_H		DDR27	DDR26	DDR25	DDR24	DDR23	DDR22	DDR21	DDR20	00000000_B
DDR3	000403_H		DDR37	DDR36	DDR35	DDR34	DDR33	DDR32	DDR31	DDR30	00000000_B
DDR4	000404_H		DDR47	DDR46	DDR45	DDR44	DDR43	DDR42	DDR41	DDR40	00000000_B
DDR5	000405_H		DDR57	DDR56	DDR55	DDR54	DDR53	DDR52	DDR51	DDR50	00000000_B
DDR6	000406_H		-	-	DDR65	DDR64	DDR63	DDR62	DDR61	DDR60	$--000000_B$
DDRC	$00040C_H$		DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	00000000_B
DDRD	$00040D_H$		DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	00000000_B
DDRE	$00040E_H$		DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	00000000_B
DDRF	$00040F_H$		DDRF7	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0	00000000_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Each port has a data direction register (DDR) to switch its data direction between input and output. A reset switches the data direction of all ports to input ($DDR=00_H$).
  - Port input mode (PFR=0 & EPFR=0 & DDR=0)

    PDR read : Reads the level at the corresponding external pin.
    PDR write: Writes a set value to the port data register.
  - Port output mode (PFR=0 & EPFR=0 & DDR=1)

    PDR read : Reads a value from the port data register.
    PDR write: Writes a set value to the port data register to output it to the corresponding external pin.
  - Peripheral output mode (PFR=1)

    PDR read : Reads the output value from the corresponding peripheral.
    PDR write: Writes a set value to the port data register.
  - Barring some unique circumstance, the input to a peripheral is always connected to a pin. Usually, use the port input mode for input to a peripheral.

4.4 Settings of Extra Port Control Registers

This section describes the functions of port functions registers.

■ Port 0

Port 0 is controlled by PFR0 and EPFR0.

In the external bus 16-bit mode, port 0 is AD[7:0] of the bus interface. In other mode, it is assigned to Multi Function serial interface ch.3 to ch.5. Select the input pin in each resource for selectable input signals.

Figure 4.4-1 Configuration of Extra Port Control Register (Port 0)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR0	000420_H		PFR07	PFR06	PFR05	PFR04	PFR03	PFR02	PFR01	PFR00	00000000_B
EPFR0	000520_H		EPFR07	EPFR06	EPFR05	EPFR04	EPFR03	EPFR02	EPFR01	EPFR00	00000000_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-1 Function of Extra Port Control Register (Port 0)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface5 SOT5 output / SDA5 I/O pin
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface4 SCK4 I/O / SCL4 I/O pin
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface4 SOT4 output / SDA4 I/O pin
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface3 SCK3 I/O / SCL3 I/O pin
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface3 SOT3 output / SDA3 I/O pin
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled

■ Port 1

Port 1 is controlled by PFR1 and EPFR1.

In the external bus 8/16-bit mode, port 1 is AD[15:7] of the bus interface. In other mode, it is assigned to multi function serial interface ch.5 to ch.7. Select the input pin in each resource for selectable input signals.

Figure 4.4-2 Configuration of Extra Port Control Register (Port 1)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR1	000421_H	PFR17	PFR16	PFR15	PFR14	PFR13	PFR12	PFR11	PFR10	00000000_B
EPFR1	000521_H	EPFR17	EPFR16	EPFR15	EPFR14	EPFR13	EPFR12	EPFR11	EPFR10	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-2 Function of Extra Port Control Register (Port 1)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface7 SCK7 I/O / SCL7 I/O pin
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface7 SOT7 output / SDA7 I/O pin
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface6 SCK6 I/O / SCL6 I/O pin
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface6 SOT6 output / SDA6 I/O pin
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface5 SCK5 I/O / SCL5 I/O pin
	1	1	Setting disabled

■ Port 2

Port 2 is controlled by PFR2 and EPFR2.

It is assigned to multi function serial interface ch.0 to ch.2. Select the input pin in each resource for selectable input signals.

Figure 4.4-3 Configuration of Extra Port Control Register (Port 2)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR2	000422_H		PFR27	PFR26	PFR25	PFR24	PFR23	PFR22	PFR21	PFR20	00000000_B
EPFR2	000522_H		EPFR27	EPFR26	EPFR25	EPFR24	EPFR23	EPFR22	EPFR21	EPFR20	11111111_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-3 Function of Extra Port Control Register (Port 2)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface2 SOT2 output / SDA2 I/O pin
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface1 SCK1 I/O SCL1 I/O pin
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface1 SOT1 output / SDA1 I/O pin
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface0 SCK0 I/O SCL0 I/O pin
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface0 SOT0 output / SDA0 I/O pin
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled

When the external bus is not set, it is invalid if it is set to external address output, and the port becomes a general-purpose port.

■ Port 3

Port 3 is controlled by PFR3 and EPFR3.

It is assigned to multi function serial interface ch.2, reload timer 0/1/2 and PWC. Select the input pin in each resource for selectable input signals.

Figure 4.4-4 Configuration of Extra Port Control Register (Port 3)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR3	000423_H	PFR37	PFR36	PFR35	PFR34	PFR33	PFR32	PFR31	PFR30	00000000_B
EPFR3	000523_H	EPFR37	EPFR36	EPFR35	EPFR34	EPFR33	EPFR32	EPFR31	EPFR30	11111111_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-4 Function of Extra Port Control Register (Port 3)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Output of reload timer TOT2
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Output of reload timer TOT1
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Output of reload timer TOT0
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface2 SCK2 I/O SCL2 I/O pin
	1	1	Setting disabled

When the external bus is not set, it is invalid if it is set to external address output, and the port becomes a general-purpose port.

■ Port 4

Port 4 is controlled by PFR4 and EPFR4.

It is assigned to the multifunction timer. Select the input pin in each resource for selectable input signals.

Figure 4.4-5 Configuration of Extra Port Control Register (Port 4)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR4	000424_H	PFR47	PFR46	PFR45	PFR44	PFR43	PFR42	PFR41	PFR40	00000000_B
EPFR4	000524_H	EPFR47	EPFR46	EPFR45	EPFR44	EPFR43	EPFR42	EPFR41	EPFR40	11111111_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-5 Configuration of Extra Port Control Register (Port 4)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface10 SCK10 I/O / SCL10 I/O pin
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface10 SOT10 I/O / SDA10 I/O pin
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TMO3 output
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TMO2 output
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TMO1 output
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TMO0 output
	1	1	Setting disabled

When the external bus is not set, it is invalid if it is set to external address output, and the port becomes a general-purpose port.

■ Port 5

Port 5 is controlled by PFR5 and EPFR5. In the external bus mode, it is control pin for the bus interface (CSnX, ASX, RDX, WRnX). In other mode, it is assigned to PPG0 to PPG3. Select the input pin in each resource for selectable input signals.

Figure 4.4-6 Configuration of Extra Port Control Register (Port 5)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR5	000425_H		PFR57	PFR56	PFR55	PFR54	PFR53	PFR52	PFR51	PFR50	00000000_B
EPFR5	000525_H		EPFR57	EPFR56	EPFR55	EPFR54	EPFR53	EPFR52	EPFR51	EPFR50	11111111_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-6 Function of Extra Port Control Register (Port 5)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Write strobe 1 (WR1X) output (Enabled when the external bus is set)
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Write strobe 0 (WR0X) output (Enabled when the external bus is set)
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Read strobe (RDX) output (Enabled when the external bus is set)
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Address strobe (ASX) output (Enabled when the external bus is set)
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Chip select 3 (CS3X) output (Enabled when the external bus is set)
	1	0	PPG3 output
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Chip select 2 (CS2X) output (Enabled when the external bus is set)
	1	0	PPG2 output
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Chip select 1 (CS1X) output (Enabled when the external bus is set)
	1	0	PPG1 output
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Chip select 0 (CS0X) output (Enabled when the external bus is set)
	1	0	PPG0 output
	1	1	Setting disabled

When the external bus is not set, it is invalid if it is set to external control output, and the port becomes a general-purpose port.

■ Port 6

Port 6 is controlled by PFR6 and EPFR6. In the external bus mode, it is control pin for the bus interface (RDY, CLK). In other mode, it is assigned to reload timer. Select the input pin in each resource for selectable input signals.

Figure 4.4-7 Configuration of Extra Port Control Register (Port 6)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFR6	000426_H	-	-	PFR65	PFR64	PFR63	PFR62	PFR61	PFR60	$--000000_B$
EPFR6	000526_H	-	-	EPFR65	EPFR64	EPFR63	EPFR62	EPFR61	EPFR60	$--001000_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-7 Function of Extra Port Control Register (Port 6)

Bit	PFR	EPFR	Function
bit7	0	0	Ignored
	0	1	Ignored
	1	0	Ignored
	1	1	Ignored
bit6	0	0	Ignored
	0	1	Ignored
	1	0	Ignored
	1	1	Ignored
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	CLK output (Enabled when the external bus is set)
	1	0	Setting disabled
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	RDY input
	1	0	TOT5 output
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TOT4 output
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	TOT3 output
	1	1	Setting disabled

When the external bus is not set, it is invalid if it is set to external control output, and the port becomes a general-purpose port.

■ Port C

Port C is controlled by PFRC and EPFRC. It is also used for PPG and multi function serial interface ch.9.

Figure 4.4-8 Configuration of Extra Port Control Register (Port C)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFRC	$00042C_H$	PFRC7	PFRC6	PFRC5	PFRC4	PFRC3	PFRC2	PFRC1	PFRC0	00000000_B
EPFRC	$00052C_H$	EPFRC7	EPFRC6	EPFRC5	EPFRC4	EPFRC3	EPFRC2	EPFRC1	EPFRC0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-8 Function of Extra Port Control Register (Port C)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPGB (PPG2 & PPG3) output
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPGA (PPG0 & PPG1) output
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface9 SCK9 I/O SCL9 I/O pin
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface9 SOT9 input / SDA9 I/O pin
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled

■ Port D

Port D is controlled by PFRD and EPFRD. It is also used for an analog input of the A/D converter. The port setting is disabled if the corresponding bit is set in the ADER register, and the port becomes an analog input pin. In this case, all the input values to the pin are handled as "0". Select the input pin in each resource for selectable input signals.

Figure 4.4-9 Configuration of Extra Port Control Register (Port D)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PFRD	$00042D_H$		PFRD7	PFRD6	PFRD5	PFRD4	PFRD3	PFRD2	PFRD1	PFRD0	00000000_B
EPFRD	$00052D_H$		EPFRD7	EPFRD6	EPFRD5	EPFRD4	EPFRD3	EPFRD2	EPFRD1	EPFRD0	00000000_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-9 Function of Extra Port Control Register (Port D)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled

Note:

By default, the ADER register is set to A/D analog input. If using other function than analog input, clear the settings of the ADER register.

■ Port E

Port E is controlled by PFRE and EPFRE. It is also used for an analog input and PPG of the multi function serial interface ch.8 and A/D converter. The port setting is disabled if the corresponding bit is set in the ADER register, and the port becomes an analog input pin. In this case, all the input values to the pin are handled as "0". Select the input pin in each resource for selectable input signals.

Figure 4.4-10 Configuration of Extra Port Control Register (Port E)

Address	bit	7	6	5	4	3	2	1	0	Initial value
PFRE	$00042E_H$	PFRE7	PFRE6	PFRE5	PFRE4	PFRE3	PFRE2	PFRE1	PFRE0	00000000_B
EPFRE	$00052E_H$	EPFRE7	EPFRE6	EPFRE5	EPFRE4	EPFRE3	EPFRE2	EPFRE1	EPFRE0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-10 Function of Extra Port Control Register (Port E)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface8 SCK8 I/O SCL8 I/O pin
	1	1	Multi Function Serial Interface8 SCK8 / SOL8 pseudo open drain output
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface8 SOT8 output / SDA8 I/O pin
	1	1	Multi Function Serial Interface8 SOT8 / SDA8 pseudo open drain output
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPG3 output
	1	1	UART PPG3 pseudo open drain output
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPG2 output
	1	1	UART PPG2 pseudo open drain output
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPG1 output
	1	1	UART PPG1 pseudo open drain output
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	PPG0 output
	1	1	UART PPG0 pseudo open drain output
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled

■ Port F

Port E is controlled by PFRF and EPFRF. It is also used for an analog input of the CEC and remote control 0, 1.

Figure 4.4-11 Configuration of Extra Port Control Register (Port F)

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PFRF	$00042F_H$		PFRF7	PFRF6	PFRF5	PFRF4	PFRF3	PFRF2	PFRF1	PFRF0	00000000_B
EPFRF	$00052F_H$		EPFRF7	EPFRF6	EPFRF5	EPFRF4	EPFRF3	EPFRF2	EPFRF1	EPFRF0	00000000_B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 4.4-11 Function of Extra Port Control Register (Port F)

Bit	PFR	EPFR	Function
bit7	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit6	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Multi Function Serial Interface SOT input / SDA I/O pin
	1	1	Setting disabled
bit5	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit4	0	0	General-purpose port
	0	1	Setting disabled
	1	0	Setting disabled
	1	1	Setting disabled
bit3	0	0	General-purpose port
	0	1	Setting disabled
	1	0	HDMI-CEC1 ACK output*
	1	1	Setting disabled
bit2	0	0	General-purpose port
	0	1	Setting disabled
	1	0	HDMI-CEC0 ACK output*
	1	1	Setting disabled
bit1	0	0	General-purpose port
	0	1	Setting disabled
	1	0	HDMI-CEC1 input/output (pseudo open drain)
	1	1	Setting disabled
bit0	0	0	General-purpose port
	0	1	Setting disabled
	1	0	HDMI-CEC0 input/output (pseudo open drain)
	1	1	Setting disabled

*: The ACK output is only available on the MB91F313A. This setting is prohibited on the MB91F313.

Note:

Only set HDMI-CEC input/output mode when using HDMI-CEC mode. Set as a general-purpose port and set as an input in the DDRF register when using remote control reception mode.

4.5 Pull-up Control Register

Pins can be added with a 33 kΩ pull-up resistor. This function can be controlled by software using the bit for each pin.

■ Pull-up Control

The pull-up function is controlled by the port pull-up control register (PCR).

Pin pull-up control is disabled automatically in the following cases:

- Port in the output state
- STOP mode output Hi-Z is selected

■ Port Pull-Up Control Register

Table 4.5-1 shows the settings of port pull-up/pull-down control registers. The setting of each bit is valid only with the corresponding PCR bit set.

Ports P07 to P00, P17 to P10, P57 to P50, and P65 to P60 can be available for the pull-up control. Each port has the corresponding bit.

Table 4.5-1 Settings of Port Pull-up/Pull-down Control Registers

Bit	Port Pull-Up/Pull-Down Control Register	
	0 (Initial Value)	1
PCRxy	No pull-up	Pull-up

Figure 4.5-1 Configuration of Port Pull-Up Control Register

	Address	bit	7	6	5	4	3	2	1	0	Initial value
PCR0	000500_H		PCR07	PCR06	PCR05	PCR04	PCR03	PCR02	PCR01	PCR00	00000000_B
PCR1	000501_H		PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	00000000_B
PCR5	000505_H		PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	00000000_B
PCR6	000506_H		-	-	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60	$--000000_B$

4.6 External Bus, I2C Bridge, ADER Control Register

This function enables the external bus. This function can be controlled by software using the bit for each pin.

■ ADER: External Bus, I2C Bridge, ADER Control Register

Figure 4.6-1 Configuration of External Bus, I2C Bridge, ADER Control Register

Address	bit	15	14	13	12	11	10	9	8	Initial value
000570_H		EXT				I2CBR12	I2CBR01	ADER9	ADER8	00000011_B
	bit	7	6	5	4	3	2	1	0	
		ADER7	ADER6	ADER5	ADER4	ADER3	ADER2	ADER1	ADER0	11111111_B

[bit15 to bit12] EXT: External bus control

Writing "1010_B" to this bit enables the external bus function.

The procedure is as follows:

- Set the necessary functions in the PFR and EPFR register from port 2 to 5.
- Set the external bus interface (640_H to 687_H).
- Writing "1010_B" to this bit enables the external bus.
- The data bus of port 0 and 1 corresponds to the mode register setting and the bus width specified in the external bus interface.
- 8-bit bus width: Port 1 is the data bus.
- 16-bit bus width: Port 1 and 0 are the data bus.
- It serves as a single chip unless this register is set. Therefore, set 8 bit or 16 bit of the data bus width for the mode register.

[bit11,bit10] I2CBR12,I2CBR01: I2C bridge function

Setting this bit (=1) enables the I2C pin to be bridged.

- If set I2CBR12, P24=P27 and P25=P30 are short-circuited inside the chip.
- If set I2CBR01, P21=P24 and P22=P25 are short-circuited inside the chip.
- If set both, P21=P24=P27 and P22=P25=P30 are short-circuited inside the chip.

[bit9 to bit0] ADER9 to ADER0: A/D input enable

The PD0 to PD7, PE0, and PE1 pins are also used for the A/D converter input. Set the corresponding bit if used for the A/D analog input.

Setting this bit (=1) fixes the input gate inside the chip to "0". If using other function than analog input, clear this bit (=0).

With this register, be sure to use half word (16 bit) for access.

4.7 Noise Filter Control Register for I2C

This section explains the noise filter control register for I2C.

■ NSF: Noise Filter Control Register for I2C

Figure 4.7-1 Configuration of Noise Filter Control Register for I2C

Address	bit	15	14	13	12	11	10	9	8	Initial value
000578_H		-	-	-	-	-	NSF10	NSF9	NSF8	$-----000_B$
	bit	7	6	5	4	3	2	1	0	
		NSF7	NSF6	NSF5	NSF4	NSF3	NSF2	NSF1	NSF0	00000000_B

[bit10 to bit0] NSF10 to NSF0: Noise filter enable bit for I2C

NSFn	Operation
0	Noise filter disabled
1	Noise filter enabled

(n = 0 to 10)

Set the value to "1" only if used with 100 kbps or faster environment.

With this register, be sure to use half word (16 bit) for access.

CHAPTER 5

16-BIT RELOAD TIMER

This chapter explains an overview, register configuration/functions and operations of the 16-bit reload timer.

5.1 Overview of 16-Bit Reload Timer

The 16-bit reload timer consists of a 16-bit down counter, a 16-bit reload register, a prescaler for creating internal count clock, and a control register.

■ Overview of 16-Bit Reload Timer

This series has three built-in channels (0 to 2), for the 16-bit reload timer.

ch.0 to ch.2 can activate the DMA transfer via interrupt.

The input clock can be selected from three internal clocks (machine clock divided by 2, 8, and 32) and an external clock.

The output pin (TOUT) provides a toggle output waveform each time an underflow occurs (in reload mode), or a rectangular wave that indicates that counting is in progress (in one shot mode).

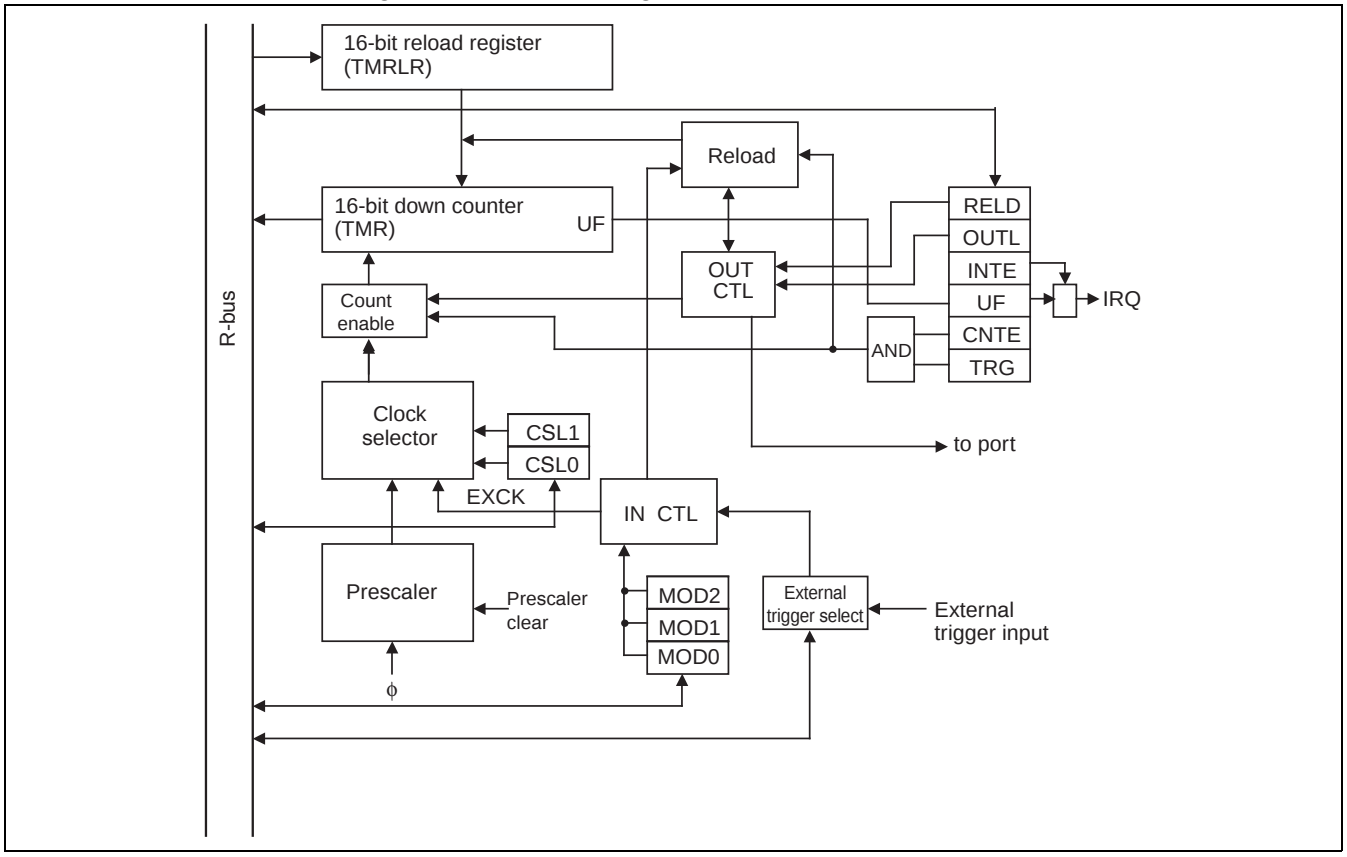
The input pin (TIN) can be used for event input in the external event count mode, and trigger or gate input in the internal clock mode.

The external event count function, when used in the reload mode, can be used as a clock frequency divider for the external clock mode.

■ Block Diagram

Figure 5.1-1 shows a block diagram of the 16-bit reload timer.

Figure 5.1-1 Block Diagram of 16-Bit Reload Timer

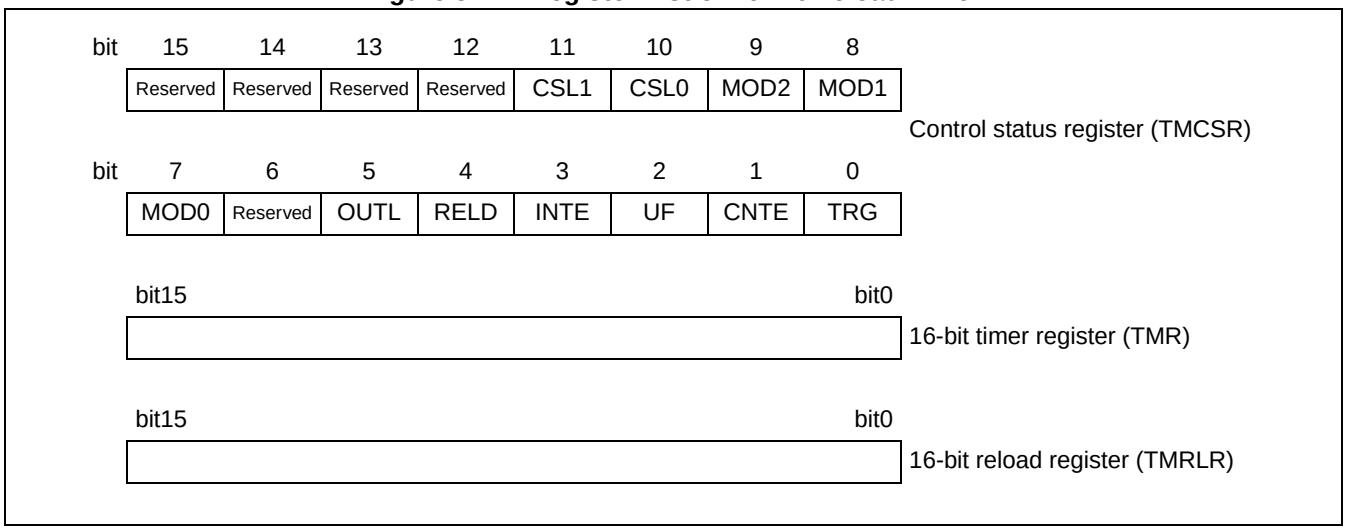


5.2 Registers of 16-Bit Reload Timer

This section explains the configurations and functions of the registers used by the 16bit reload timer.

■ Register List of 16-Bit Reload Timer

Figure 5.2-1 Register List of 16-Bit Reload Timer



5.2.1 Control Status Register (TMCSR)

The control status register (TMCSR) controls the operating modes and interrupts of the 16-bit reload timer.

■ Bit Configuration of Control Status Register (TMCSR)

Figure 5.2-2 Bit Configuration of Control Status Register (TMCSR)

TMCSR

Address	bit	15	14	13	12	11	10	9	8	Initial value
ch.0 $00004E_H$ ch.1 000056_H ch.2 $00005E_H$		Reserved	Reserved	Reserved	Reserved	CSL1	CSL0	MOD2	MOD1	00000000_B
		R	R	R	R	R/W	R/W	R/W	R/W	
	bit	7	6	5	4	3	2	1	0	Initial value
		MOD0	Reserved	OUTL	RELD	INTE	UF	CNTE	TRG	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/Writable
R: Read only

Rewrite the bit other than UF, CNTE, and TRG bits if CNTE=0.

The control status register (TMCSR) supports simultaneous writing.

The following explains bit functions of control status register (TMCSR).

[bit15 to bit12] Reserved: Reserved bits

The read value is always "0000_B".

[bit11, bit10] CSL1, CSL0 (Count clock SeLect)

These bits are the count clock select bits. Table 5.2-1 shows the clock sources that can be selected using these bits. Countable edges used when external event count mode is set are set by the MOD1 and MOD0 bits.

Table 5.2-1 Clock Sources Set Using the CSL Bits

CSL1	CSL0	Clock Source (ϕ: Machine Clock)
0	0	$\phi/2^1$
0	1	$\phi/2^3$
1	0	$\phi/2^5$
1	1	External clock (event)

Note: The minimum pulse width required for an external clock is 2T (T: Peripheral machine clock cycle).

[bit9 to bit7] MOD2, MOD1, MOD0 (MODe)

These bits set the operating modes and functions of the I/O pin.

The MOD2 bit selects the functions of the input pin. If this bit is "0", the input pin becomes the trigger input pin. When a valid edge is input, the contents of the reload register are loaded into the counter and the count operation is continued. Setting this bit to "1" enters the gate count mode. The input pin becomes the gate input, and the count operation is performed only when a valid level is being input.

The MOD1 and MOD0 bits set the pin function for each mode. Table 5.2-2 and Table 5.2-3 show the settings of the MOD2, MOD1, and MOD0 bits.

Table 5.2-2MOD2, MOD1, and MOD0 Setting Method 1 (in Internal Clock Mode (CSL0, CSL1=00_B, 01_B, 10_B))

MOD2	MOD1	MOD0	Input Pin Function	Valid Edge, Level
0	0	0	Trigger disabled	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edges
1	×	0	Gate input	"L" level
1	×	1		"H" level

Table 5.2-3MOD2, MOD1, and MOD0 Setting Method 2 (in Event Count Mode (CSL0, CSL1=11_B))

MOD2	MOD1	MOD0	Input Pin Function	Valid Edge, Level
×	0	0	-	-
	0	1	Event input	Rising edge
	1	0		Falling edge
	1	1		Both edges

Note: × in this table represents any value.

[bit6] Reserved: Reserved bit

The read value is always "0".

[bit5] OUTL

This bit sets the output level of the TOUT pin. The pin level is reversed depending on whether this bit is "0" or "1". This bit, bit4 (RELD bit), and the corresponding bit of the PFR register of the I/O port are combined to specify the output waveform. Table 5.2-4 shows the setting of a combination of these bits.

Table 5.2-4 Setting of PFR, RELD, and OUTL

PFR	OUTL	RELD	Output waveform
0	×	×	General-purpose port
1	0	0	Rectangular wave of "H" during counting
1	1	0	Rectangular wave of "L" during counting
1	0	1	Toggle output of "L" at count start
1	1	1	Toggle output of "H" at count start

Note: PFR is the corresponding bit of the PFR register of the I/O port

[bit4] RELD

This bit is the reload enable bit."1" turns on the reload mode. As soon as the counter value underflows from "0000_H" → "$FFFF_H$", the contents of the reload register are loaded into the counter and the count operation is continued.

If this bit is set to "0", the count operation is stopped when the counter value underflows from "0000_H" → "$FFFF_H$".

[bit3] INTE

This bit is an interrupt request enable bit. If the INTE bit is set to "1," an interrupt request is generated when the UF bit is set to "1."If it is set to "0", no interrupt request is generated.

[bit2] UF

This bit is the timer interrupt request flag. This bit is set to "1" when the counter value underflows from "0000_H" → "$FFFF_H$". Writing "0" clears the flag.

Writing "1" to this bit has no effect on the operations. For a read-modify-write (RMW) instruction, "1" is always read.

[bit1] CNTE

This bit is the count enable bit of the timer. Write "1" to this bit to enter the activation trigger wait state. Writing "0" to this bit stops the count operation.

[bit0] TRG

This bit is the software trigger bit. Writing "1" to this bit generates a software trigger, loads the contents of the reload register into the counter, and starts the count operation.

Writing "0" to this bit has no effect on the operations. Reading value is always "0".

The trigger input to this register is valid only if CNTE=1.

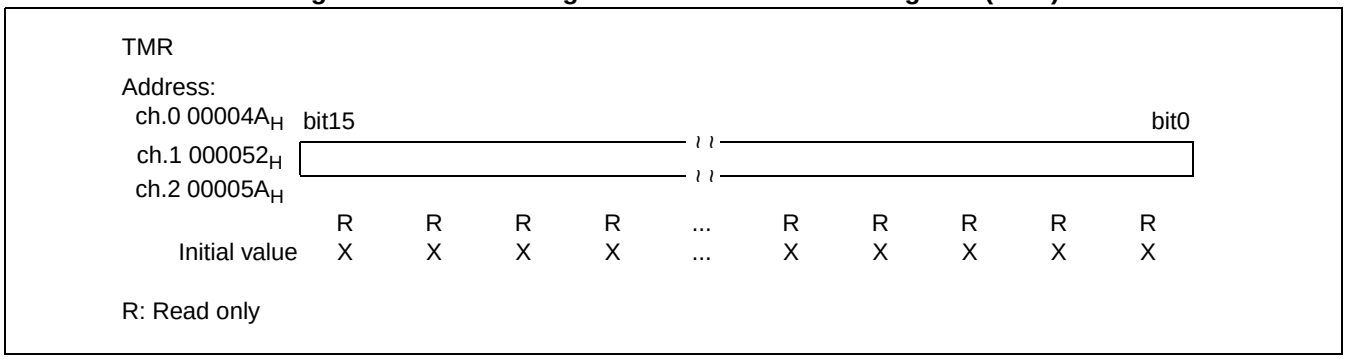
5.2.2 16-Bit Timer Register (TMR)

The 16-bit timer register (TMR) can be used to read the count value of the 16-bit timer.
The initial value is undefined.
Be sure to read this register using a 16-bit data transfer instruction.

■ Bit Configuration of 16-Bit Timer Register (TMR)

Figure 5.2-3 shows the bit configuration of the 16-bit timer register (TMR).

Figure 5.2-3 Bit Configuration of 16-Bit Timer Register (TMR)



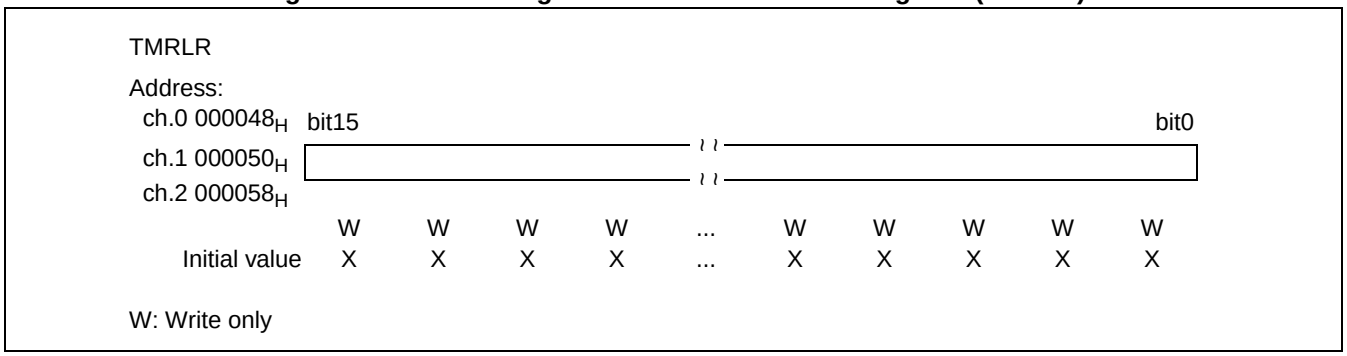
5.2.3 16-Bit Reload Register (TMRLR)

The 16-bit reload register (TMRLR) holds the initial value of a counter. The initial value is undefined.
Be sure to write the value to this register using a 16-bit data transfer instruction.

■ Bit Configuration of 16-Bit Reload Register (TMRLR)

Figure 5.2-4 shows the bit configuration of the 16-bit reload register (TMRLR).

Figure 5.2-4 Bit Configuration of 16-Bit Reload Register (TMRLR)



5.3 Operations of 16-Bit Reload Timer

This section explains the following operations of the 16-bit reload timer:

- **Internal clock operation**
- **Underflow operation**
- **Input pin function operation**
- **Output pin function operation**

■ Internal Clock Operation

If the timer operates with a division clock of the internal clock, one of the clocks created by dividing the machine clock by 2, 8, or 32 can be selected as the clock source.

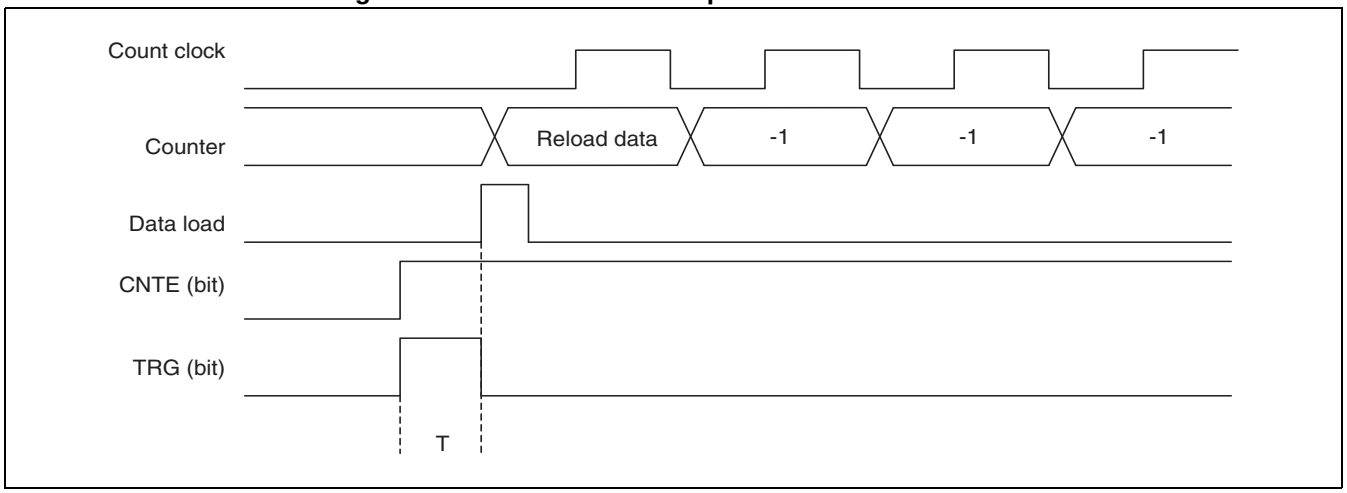
The external input pin can be used for the trigger input or gate input depending on the register setting.

To start the count operation as soon as counting is enabled, write "1" to the CNTE and TRG bits of the control status register. Trigger input due to the TRG bit is always valid regardless of the operating mode, when the timer is running (CNTE=1).

Figure 5.3-1 shows the activation and operations of the counter.

Time as long as T (T: peripheral clock machine cycle) is required after the counter activation trigger is input and before the data of the reload register is actually loaded into the counter.

Figure 5.3-1 Activation and Operations of the Counter



■ Underflow Operation

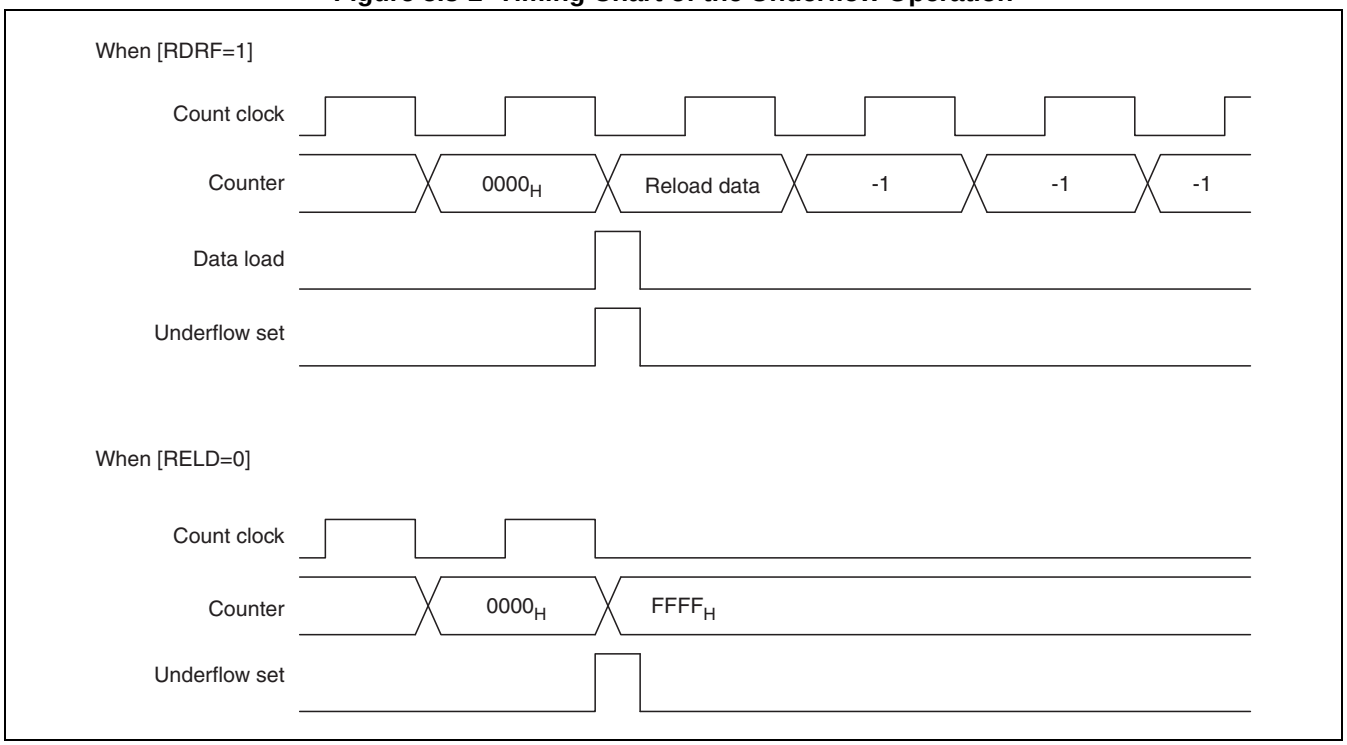
Underflow is an event in which the counter value changes from "0000_H" to "$FFFF_H$". Thus, an underflow occurs at the count of [Reload register setting value +1].

If the RELD bit of the control status register (TMCSR) is set to "1" when an underflow occurs, the contents of the 16-bit reload register (TMRLR) are loaded into the counter and the count operation is continued. If the RELD bit is set to "0", the counter stops at "$FFFF_H$".

The underflow sets the UF bit of the control status register (TMCSR). If the INTE bit is set to "1," an interrupt request is generated.

Figure 5.3-2 shows the timing chart of the underflow operation.

Figure 5.3-2 Timing Chart of the Underflow Operation



■ Operation of Input Pin Function (in Internal Clock Mode)

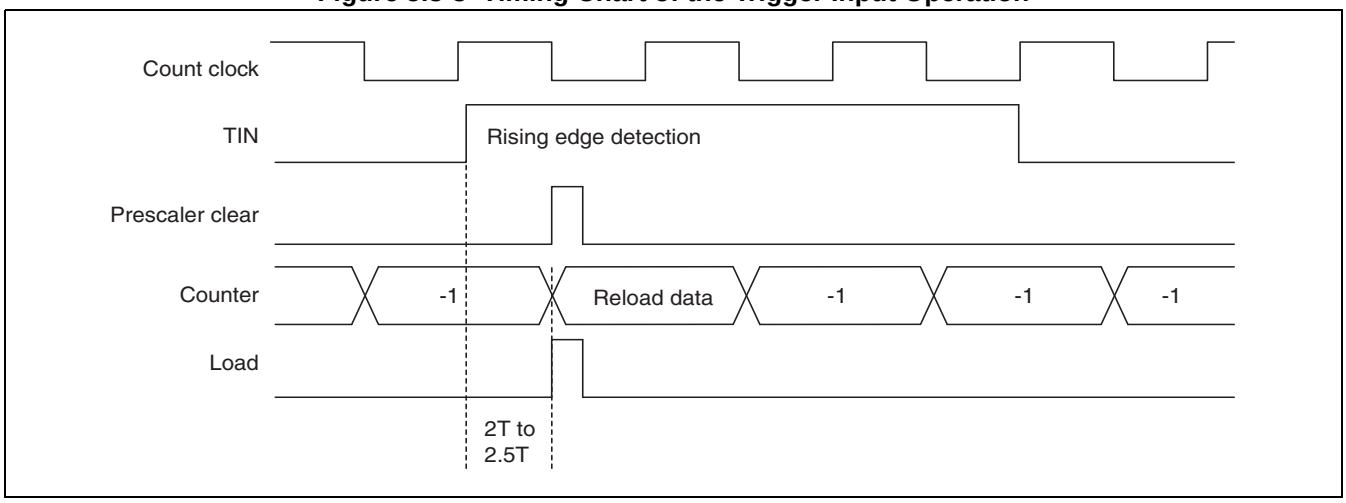
If an internal clock is selected as the clock source, the TIN pin can be used as the trigger input or gate input.

● Operation of Trigger Input

If the pin is used as the trigger input when a valid edge is input, the contents of the 16-bit reload register (TMRLR) are loaded into the counter, the internal prescaler is cleared, and the count operation is started. Input the pulse with 2T (T is peripheral clock machine cycle) or more for TIN.

Figure 5.3-3 shows the timing chart of the trigger input operation.

Figure 5.3-3 Timing Chart of the Trigger Input Operation

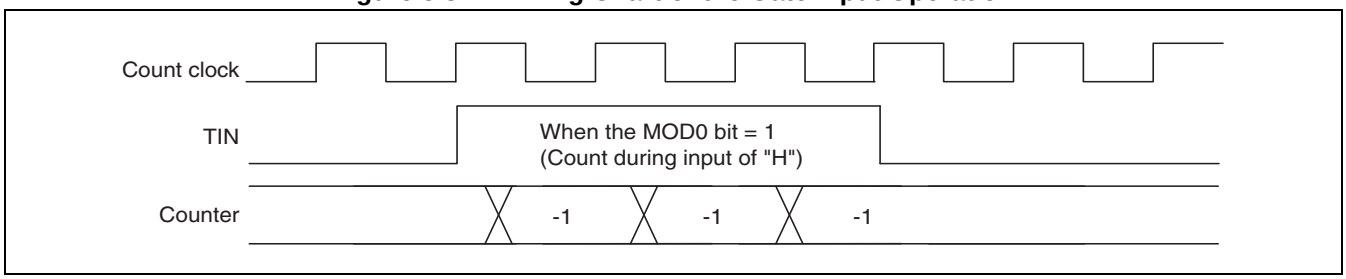


● Operation of Gate Input

If the pin is used as the gate input, the count operation is performed only while the valid level, set by the MOD0 bit of the control status register (TMCSR), is input from the TIN pin. In this case, the count clock keeps operating. In the gate mode, the software trigger is available, regardless of the gate level. The pulse width for the TIN pin should be 2T (T is peripheral clock machine cycle) or more.

Figure 5.3-4 shows the timing chart of the gate input operation.

Figure 5.3-4 Timing Chart of the Gate Input Operation



■ External Event Count Operation

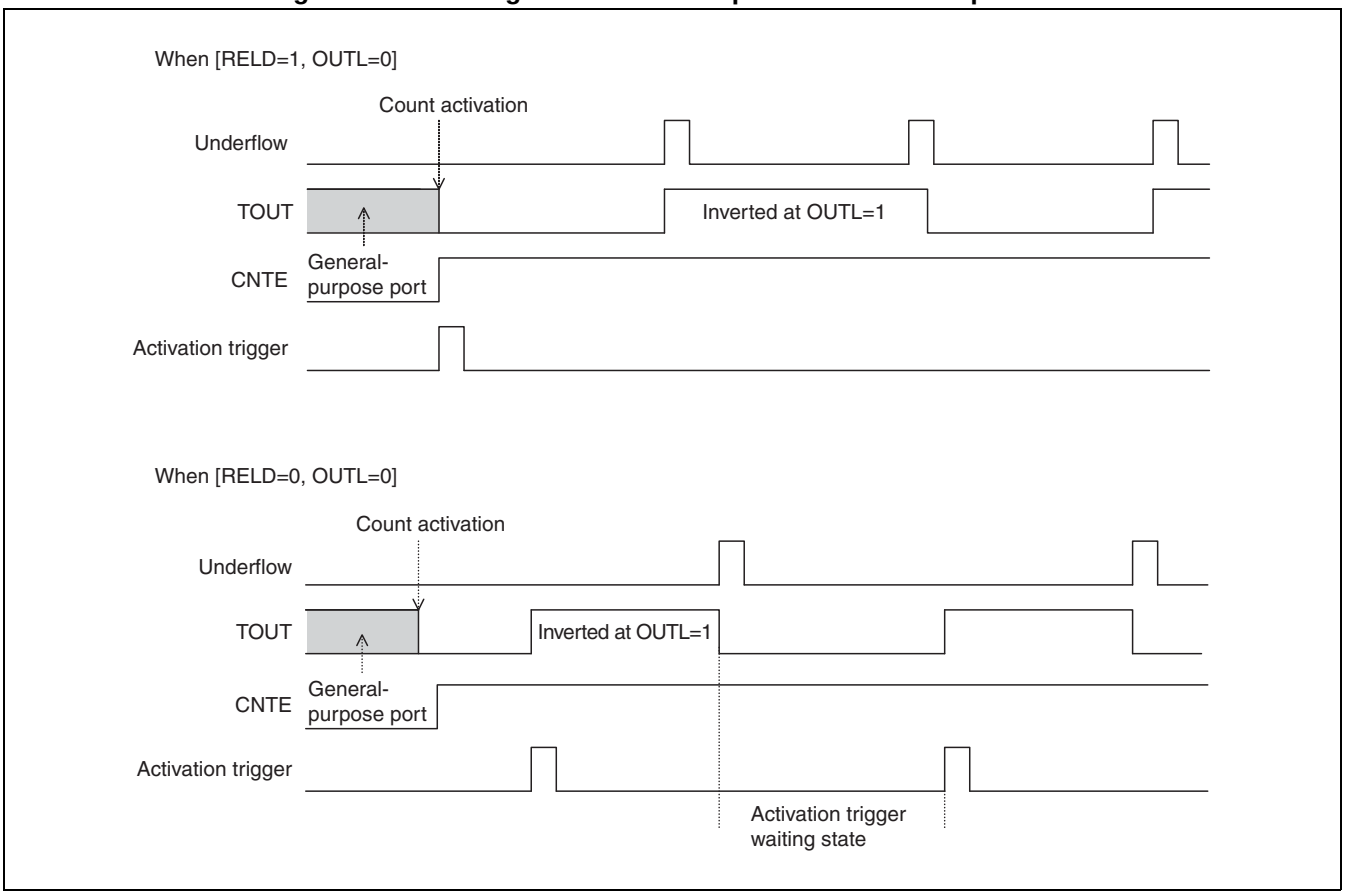
If an external clock is selected, the TIN pin becomes the external event input pin, and the valid edge set by the register is counted. The pulse width for the TIN pin should be 2T (T is peripheral clock machine cycle) or more.

■ Output Pin Function Operation

The TOT pin provides a toggle output that is inverted by an underflow (in reload mode), or a pulse output that indicates that counting is in progress (in one shot mode). The output polarity can be set using the OUTL bit of the control status register (TMCSR). If OUTL=0, toggle output is "0" for the initial value, and the one-shot pulse output is "1" while the count operation is in progress. If OUTL=1, the output waveform is reversed.

Figure 5.3-5 shows the timing chart of the output pin function operation.

Figure 5.3-5 Timing Chart of the Output Pin Function Operation



■ Other Operations

Ch.0 to ch.2 of the 16-bit reload timer can be used to activate the DMA transfer with its interrupt request signal.

The DMA controller clears the interrupt flag of the reload timer at the time of the reception of a transfer request.

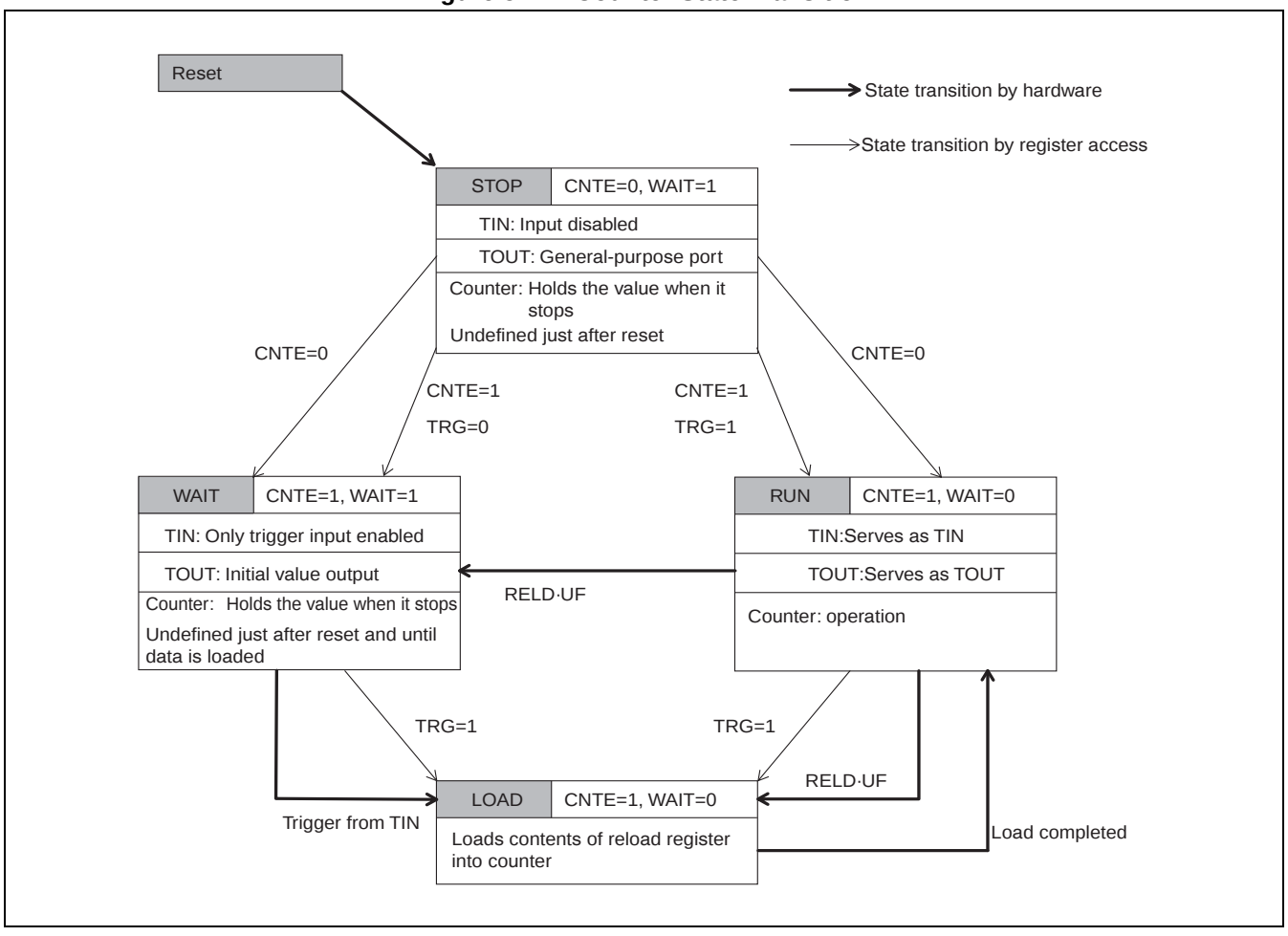
5.4 Operating Status of Counter

The counter state is determined by the CNTE bit of the control status register (TMCSR) and the WAIT signal, which is an internal signal. The states that can be set include the stop state, when CNTE=0 and WAIT=1 (STOP state); the activation trigger wait state, when CNTE=1 and WAIT=1 (WAIT state); and the operation state, when CNTE=1 and WAIT=0 (RUN state).

■ Operating Status of Counter

Figure 5.4-1 shows the transition of each state.

Figure 5.4-1 Counter State Transition



5.5 Notes on Using 16-Bit Reload Timer

This section explains the precautions when using the 16-bit reload timer.

■ Notes on Using 16-Bit Reload Timer

- Internal prescaler

  The internal prescaler is enabled if a trigger (software or external trigger) is applied while bit1 (timer enable: CNTE) of the control status register (TMCSR) is set to "1".

  Even if used only with the gate count mode, be sure to assign a trigger before inputting the valid gate level.

  It is recommended that "1" is written to bit0 (TRG) of the TMCSR register when setting CNTE.

- Timing of setting and clearing the interrupt request flag

  If the device attempts to set and clear the interrupt request flag at the same time, the flag is set and the clear operation becomes ineffective.

- 16-bit timer register (TMR) / 16-bit reload register (TMRLR)

  If the device attempts to write to the 16-bit timer register and reload the data into the 16-bit reload register at the same time, old data is loaded into the counter. New data is loaded into the counter only in the next reload timing.

- 16-bit timer register (TMR)

  If the device attempts to load and count the 16-bit timer register at the same time, the load (reload) operation takes precedence.

CHAPTER 6

PPG (PROGRAMMABLE PULSE GENERATOR)

This chapter gives an outline of the PPG (Programmable Pulse Generator) timer and explains the register configuration and functions and the timer operations.

6.1 Outline of the PPG Timer

The PPG timer can efficiently output highly accurate PWM waveforms.
The MB91313A series has four channels of the PPG timer.

■ Characteristics of PPG Timer

- Each channel consists of a 16-bit down counter, 16-bit data register with a cycle setting buffer, 16-bit compare register with a duty setting buffer, and pin control block.
- One of the four count clocks can be selected for the 16-bit down counter:
  Peripheral clocks: ϕ, ϕ/4, ϕ/16, and ϕ/64
- A reset or counter borrow can initialize the counter value to "$FFFF_H$".
- Each channel has PPG output (PPG0 to PPG3).
- Registers

  Cycle setting register : Data register for reload with buffer
  Data is transferred from the buffer when an activation trigger signal is detected and a counter borrow occurs.
  The PPG output is inverted when a counter borrow occurs.

  Duty setting register : Compare register with buffer
  PPG output is inverted when the value of this register and the counter value match.
- Pin control

  Set to "1" when the duty matches (priority).

  Reset to "0" when a counter borrow occurs.

  Output-value fixed mode is available to facilitate output of all-L (or H).

  The polarity can be specified.
- An interrupt request can be generated as one of the following combinations:

  Activation of PPG timer (software trigger or trigger input)

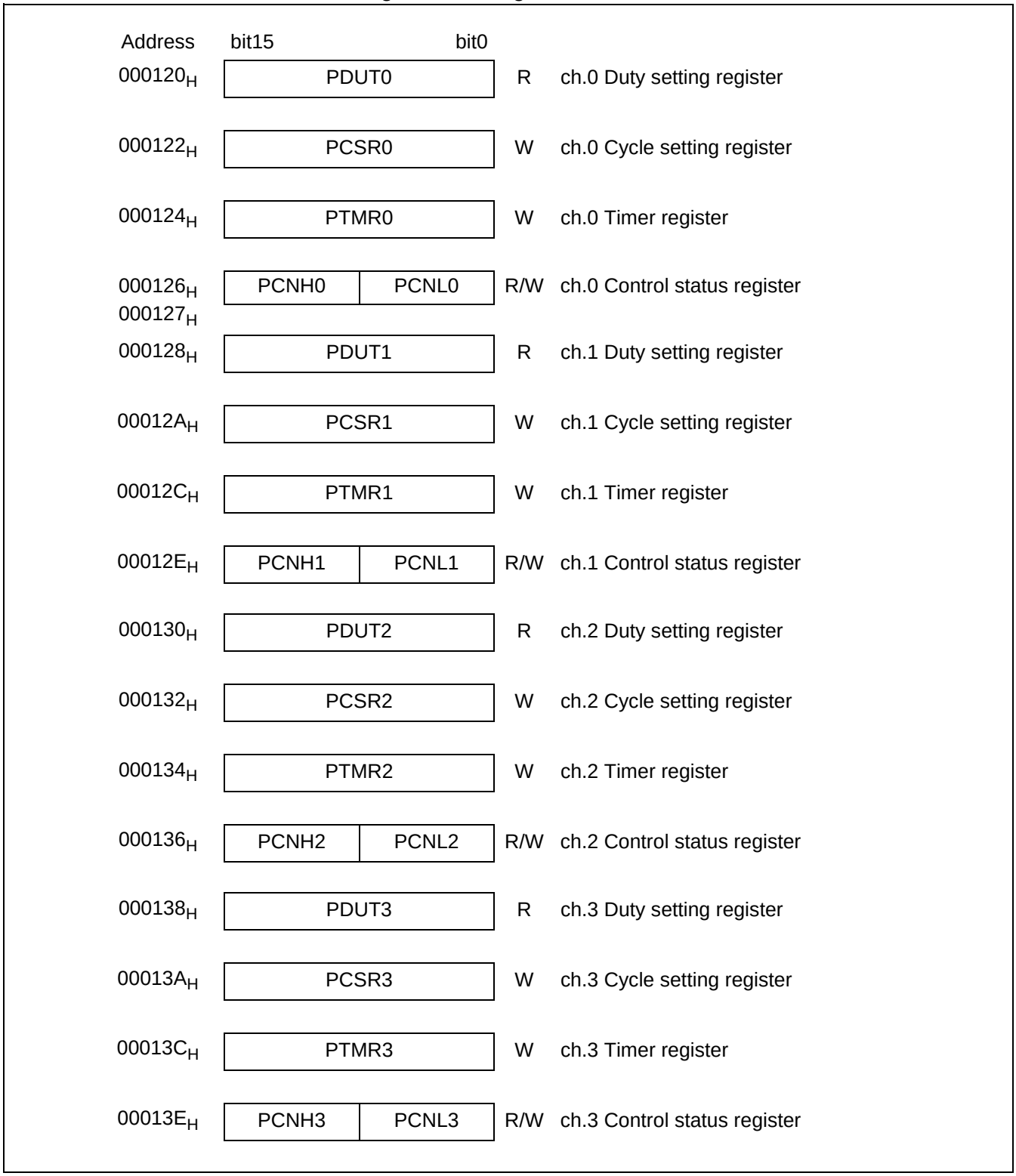
  Generation of counter borrow (cycle match)

  Generation of duty match

  Generation of counter borrow (cycle match) or duty match
- Restart during operation can be set.
- Remote control sending is supported.

■ Registers

Figure 6.1-1 Registers of PPG



■ Block Diagram

Figure 6.1-2 Overall Block Diagram

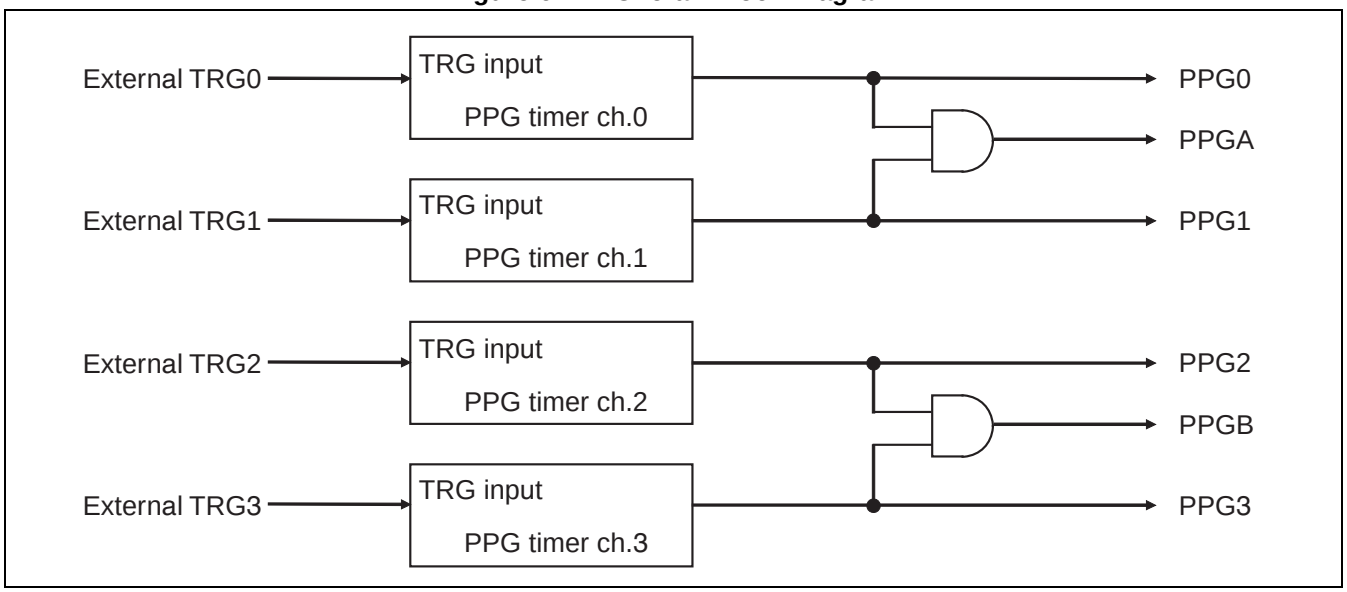
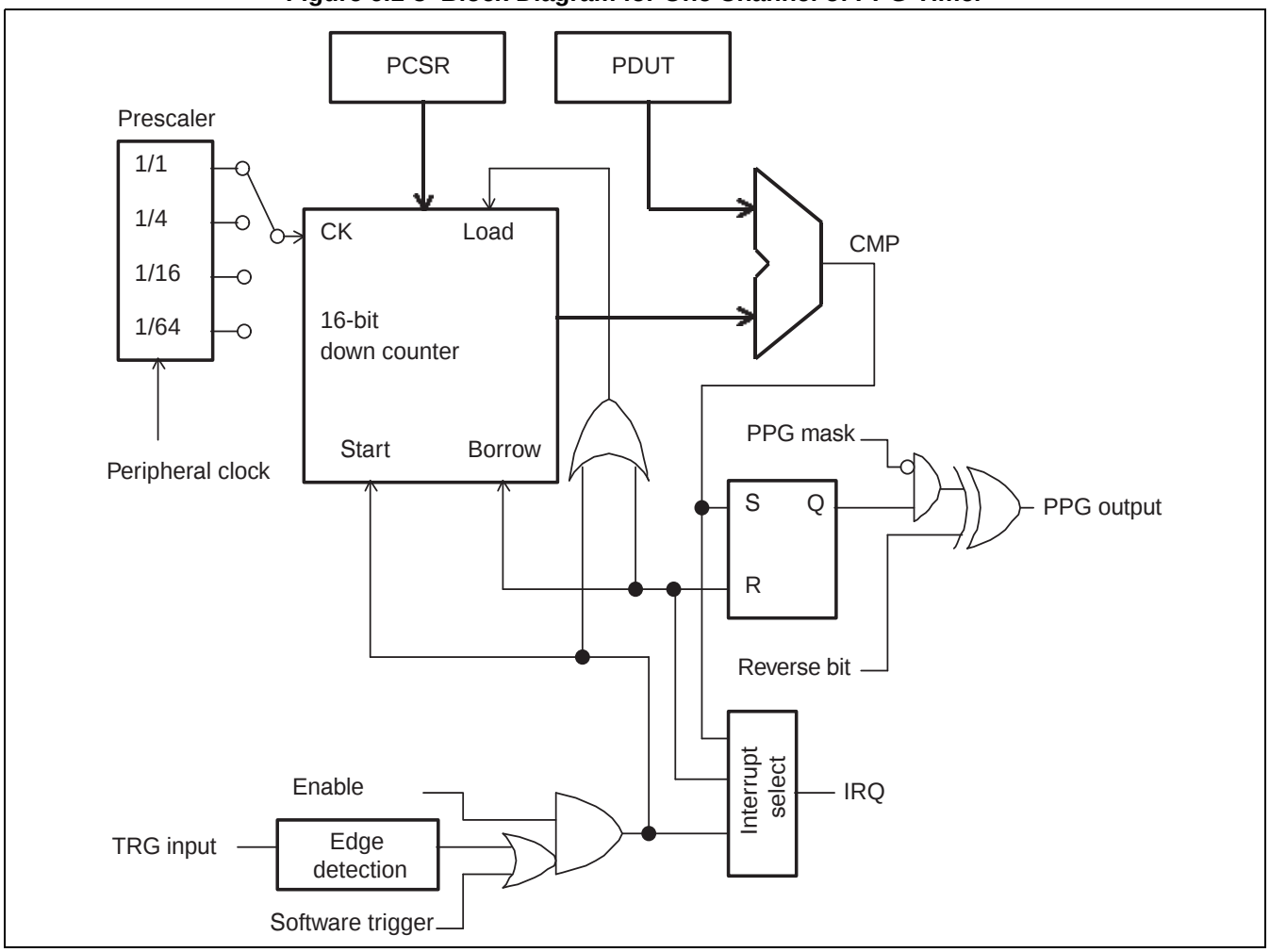


Figure 6.1-3 Block Diagram for One Channel of PPG Timer



■ Control Status Register

Figure 6.1-4 Bit Configuration of Control Status Register

PCNH

	bit	15	14	13	12	11	10	9	8	
Address: ch.0 000126_H ch.1 $00012E_H$ ch.2 000136_H ch.3 $00013E_H$		CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	-	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	← Attribute
		0	0	0	0	0	0	0	-	← Initial value
		O	O	×	×	×	×	O	-	← Rewrite during operation

PCNL

	bit	7	6	5	4	3	2	1	0	
Address: ch.0 000127_H ch.1 $00012F_H$ ch.2 000137_H ch.3 $00013F_H$		EGS1	EGS0	IREN	IRQF	IRS1	IRS0	-	OSEL	
		R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	← Attribute
		0	0	0	0	0	0	-	0	← Initial value
		×	×	O	O	×	×	-	×	← Rewrite during operation

R/W: Readable/writable
-: Unused

[bit15] CNTE: Timer enable bit

This bit enables operation of the 16-bit down counter.

0	Disabled (initial value)
1	Enabled

[bit14] STGR: Software trigger bit

Writing 1 into this bit applies software trigger.

The read value is always 0.

[bit13] MDSE: Mode select bit

This bit is used to select either the PWM mode in which continuous pulses are output or the one-shot mode in which a single pulse is output.

0	PWM mode (initial value)
1	One-shot mode

[bit12] RTRG: Restart select bit

This bit enables a restart resulting from a software trigger or trigger input.

0	Restart disabled (initial value)
1	Restart enabled

[bit11, bit10] CKS1, CKS0: Count clock select bit

These bits are used to select the count clock of the 16-bit down counter.

CKS1	CKS0	Cycle
0	0	ϕ (initial value)
0	1	ϕ/4
1	0	ϕ/16
1	1	ϕ/64

ϕ: Peripheral machine clock

[bit9] PGMS: PPG Output mask select bit

Writing 1 into this bit allows PPG output to be masked to 0 or 1, regardless of mode, cycle, and duty settings.

PPG output when write "1" to PGMS

Polarity	PPG output
Ordinary polarity	L output
Reverse polarity	H output

For all-H output in ordinary polarity mode and all-L output in reverse polarity mode, specify the same value in the cycle setting register and duty setting register in order to output the above mask value with the polarity reversed.

[bit8] This bit is undefined.

[bit7, bit6] EGS1, EGS0: Trigger input edge select bit

These bits are used to select an effective edge for PPG input.

Regardless of the mode that is selected, writing "1" to the bit of a software trigger enables the software trigger.

EGS1	EGS0	Edge selection
0	0	Not effective (initial value)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

[[bit5] IREN: Interrupt request enable bit

0	Disabled (initial value)
1	Enabled

[bit4] IRQF: Interrupt request flag bit

If bit5:IREN, is enabled and an interrupt source selected in bit3, bit2:IRS1 and IRS0, occurs then this bit is set and an interrupt request is generated and issued to the CPU.

This bit is cleared if "0" is written to it.

This bit remains unchanged if "1" is written to it.

The read value by a read-modify-write (RMW) instruction is always "1", regardless of the bit value.

[bit3, bit2] IRS1, IRS0: Interrupt resource select bit

These bits are used to select a source that sets bit4:IRQF.

IRS1	IRS0	Interrupt resource
0	0	Software trigger or trigger input (initial value)
0	1	Occurrence of a counter borrow (cycle match)
1	0	Occurrence of a duty match
1	1	Occurrence of a counter borrow (cycle match) or duty match

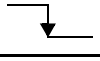
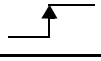
[bit1] Unused bit

[bit0] OSEL: PPG output polarity specification bit

This bit sets the polarity of the PPG output.

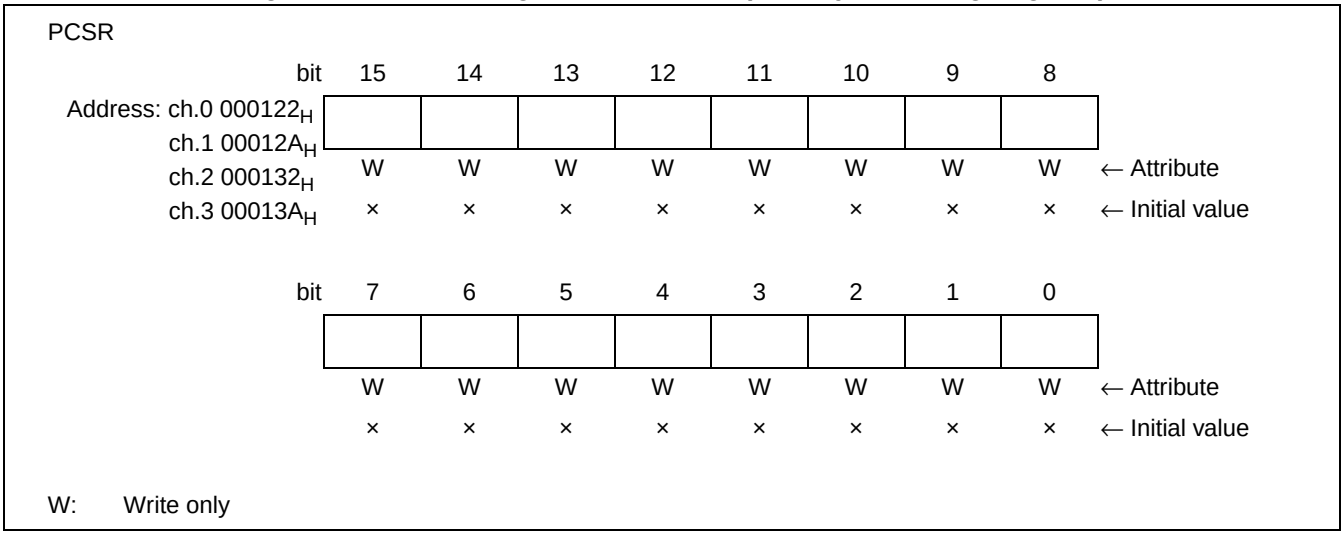
The followings show the combination results for this bit and bit9:PGMS.

PGMS	OSEL	PPG output
0	0	Ordinary polarity (initial value)
0	1	Reverse polarity
1	0	Output fixed to "L"
1	1	Output fixed to "H"

Polarity	After reset	Duty match	Counter borrow
Ordinary polarity	"L" output		
Reverse polarity	"H" output		

■ PCSR (PPG Cycle Setting Register)

Figure 6.1-5 Bit Configuration of PCSR (PPG Cycle Setting Register)



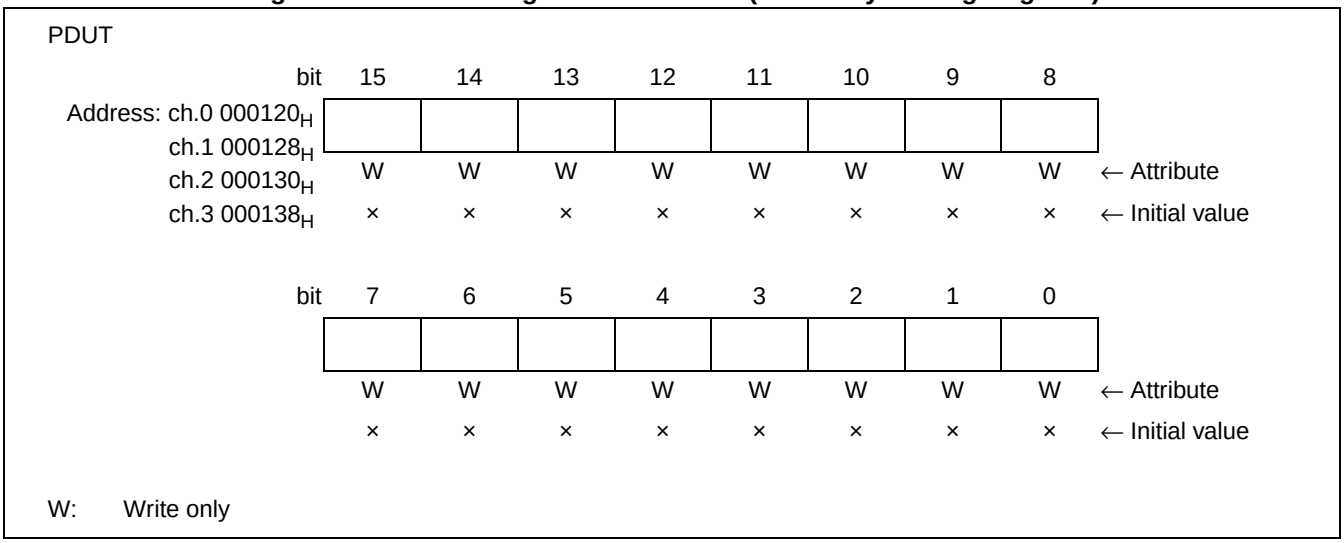
The PPG cycle setting register (PCSR) is a register with a buffer for setting a cycle. Transfers from the buffer are performed with counter borrow.

When initializing or rewriting the cycle setting register, be sure to write to the duty setting register after the writing of the cycle setting register.

This register must be accessed using 16-bit data.

■ PDUT (PPG Duty Setting Register)

Figure 6.1-6 Bit Configuration of PDUT (PPG Duty Setting Register)



The PPG duty setting register (PDUT) is a register with buffer for setting a duty. Transfers from the buffer are performed with counter borrow.

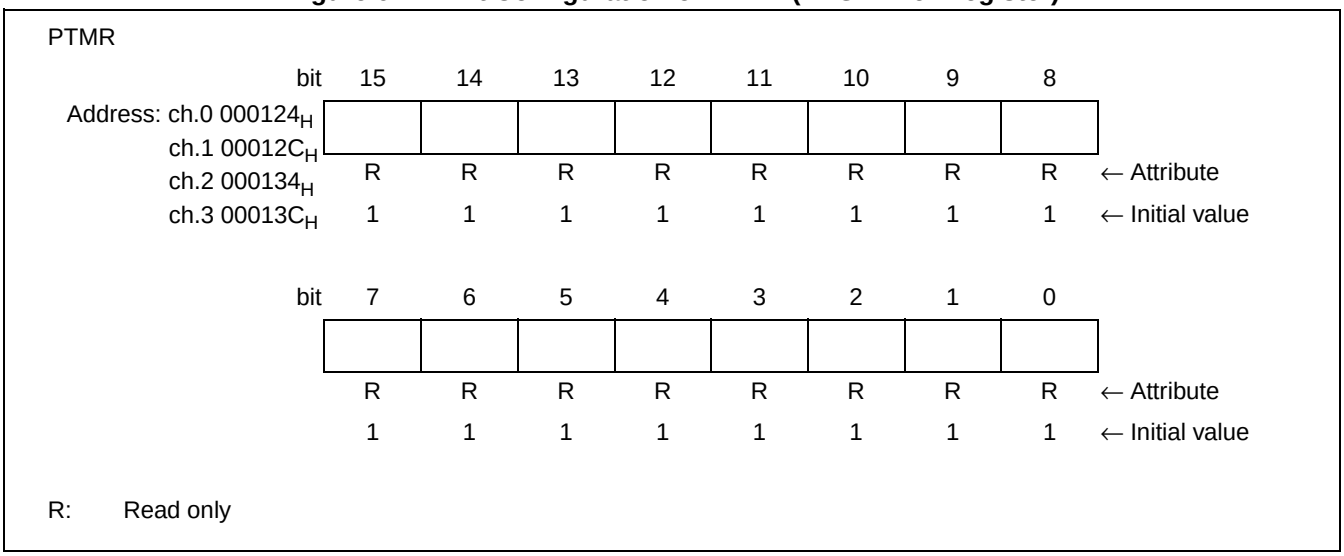
When the same value is set in the cycle setting register and the duty setting register, all-H is output in ordinary polarity mode and all-L is output in reverse polarity mode.

Do not specify a smaller value in PCSR than that in PDUT. Otherwise, PPG output becomes undefined.

This register must be accessed using 16-bit data.

■ PTMR (PPG Timer Register)

Figure 6.1-7 Bit Configuration of PTMR (PPG Timer Register)



The PPG timer register (PTMR) is a register used to read the value of the 16-bit down counter.

This register must be accessed using 16-bit data.

6.2 Operation of the PPG Timer

This section describes the PPG timer operation.

■ Timing Charts for PWM Operation

Figure 6.2-1 PWM Operation Timing Chart (When reactivation is disabled)

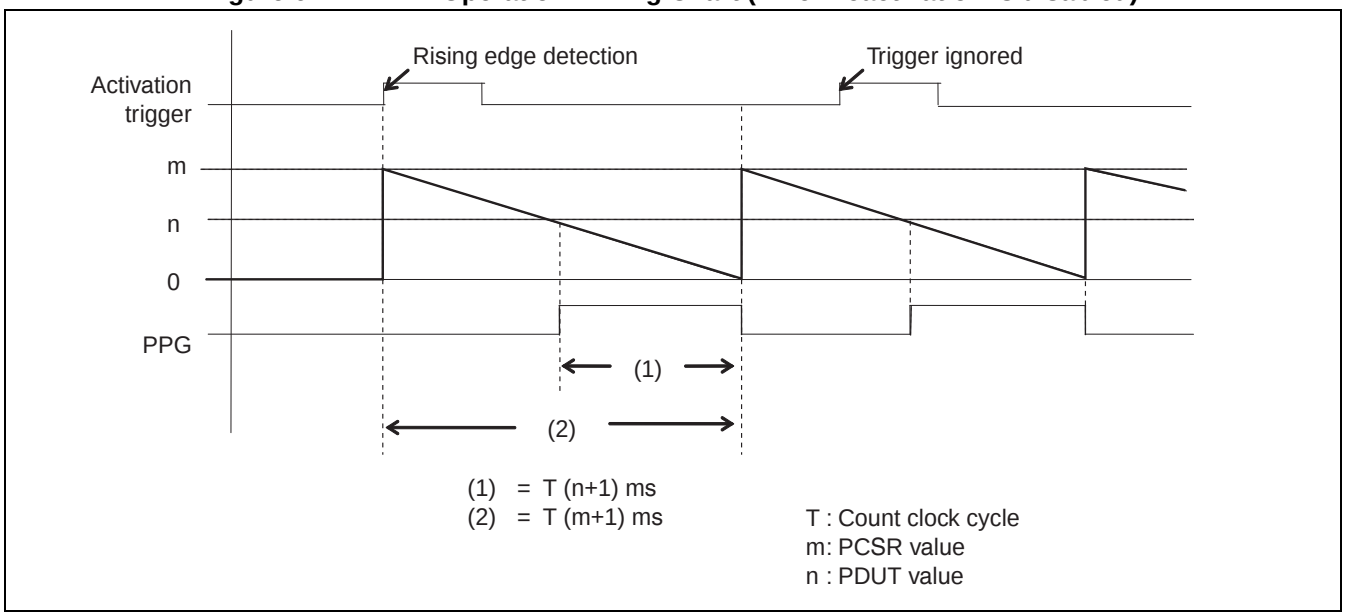
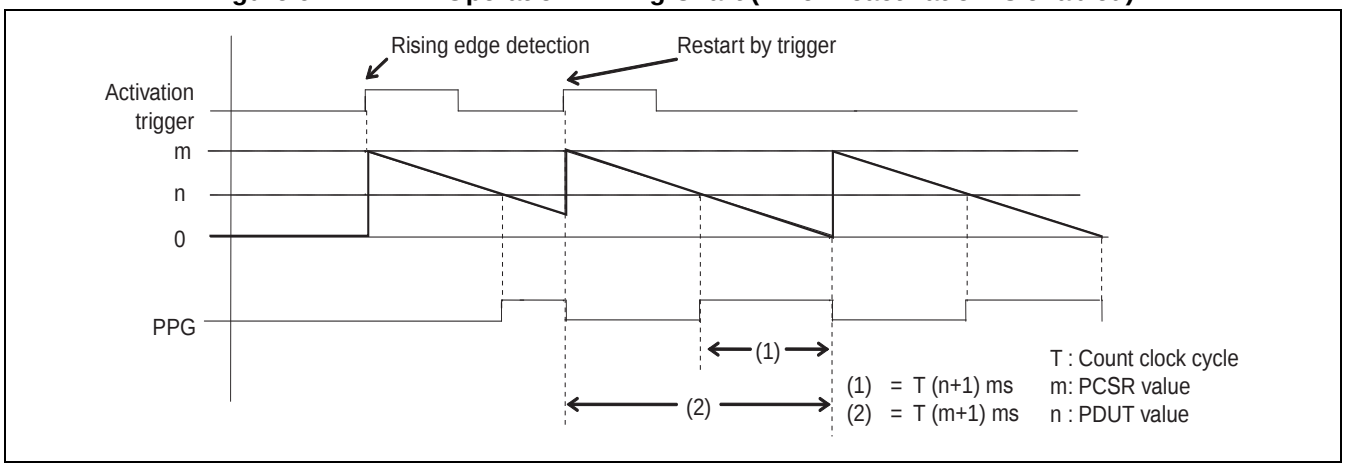


Figure 6.2-2 PWM Operation Timing Chart (When reactivation is enabled)



● PWM mode

In PWM mode, the PPG timer can output pulses continuously after an activation trigger signal is detected.

The output pulse cycle can be controlled with the PCSR value, and the duty ratio can be controlled with the PDUT value.

Note: After data is written to PCSR, be sure to write data to PDUT.

■ Timing Charts for One-Shot Operation

Figure 6.2-3 Timing Charts for One-Shot Operation (When restart is disabled)

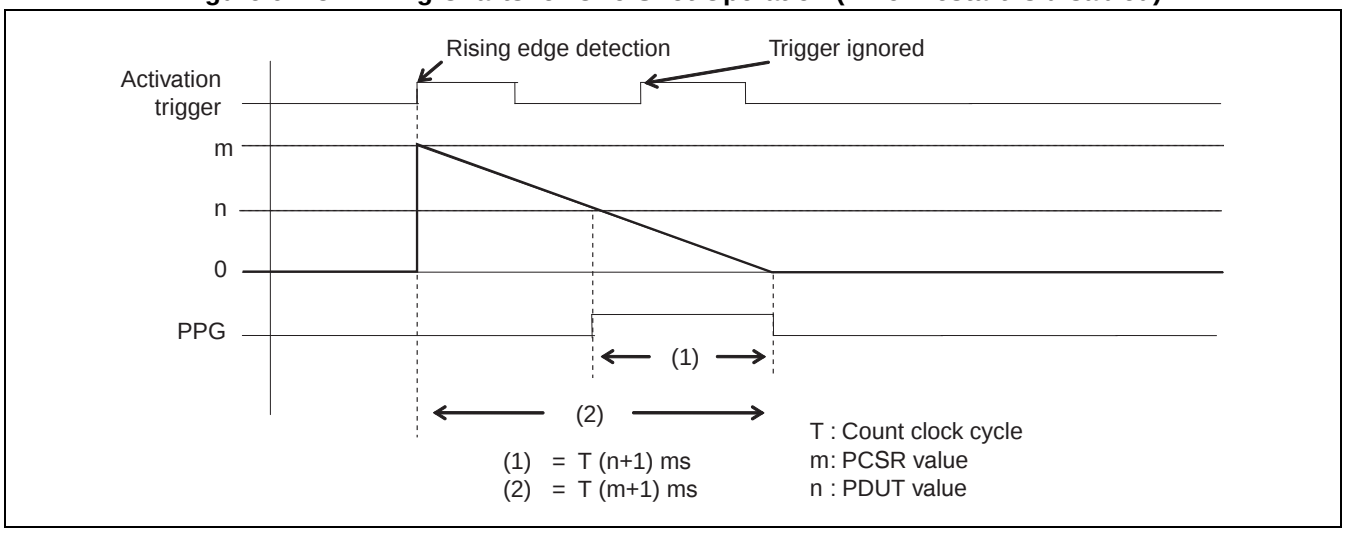
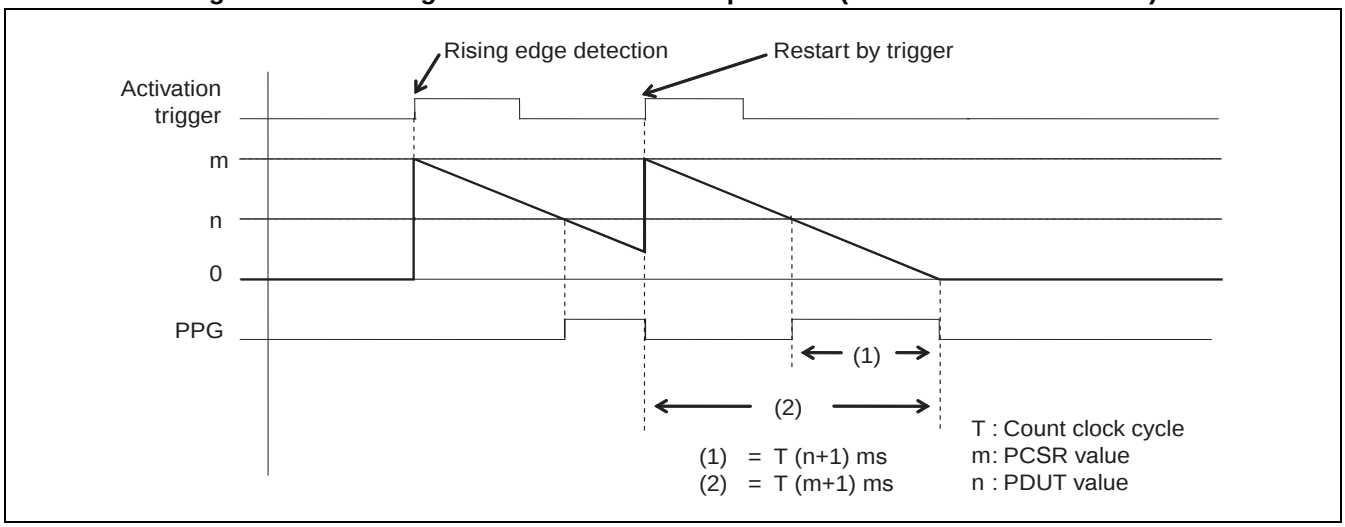


Figure 6.2-4 Timing Charts for One-Shot Operation (When restart is enabled)



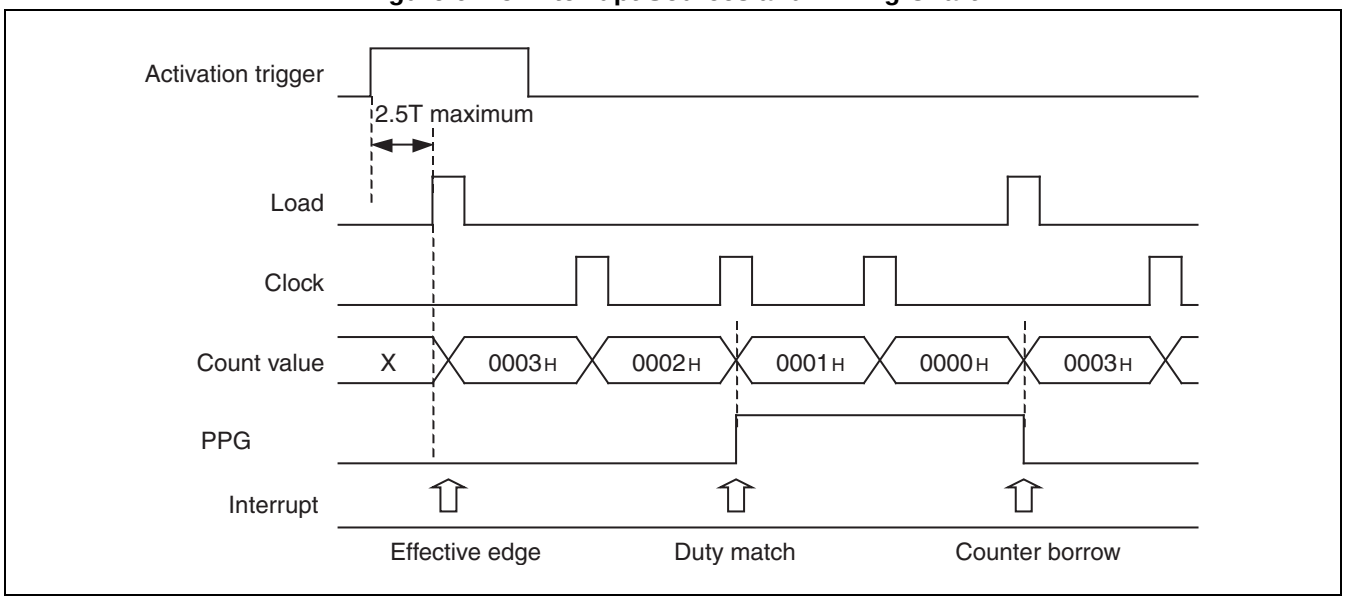
● One-shot mode

In one-shot mode, the PPG timer can output a single pulse of an arbitrary width when triggered.

When reactivation is enabled, the PPG timer reloads the counter value after an edge is detected during operation.

■ Interrupt Sources and Timing Chart (with PPG Output Set for Ordinary Polarity)

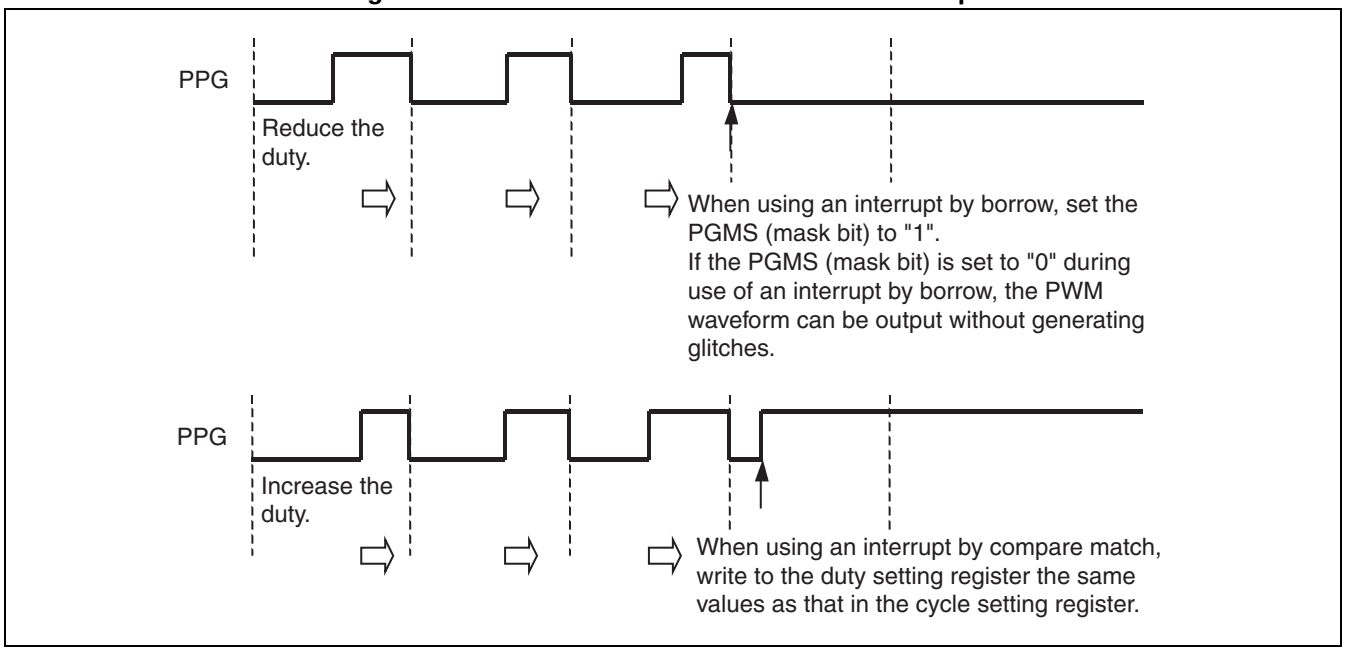
Figure 6.2-5 Interrupt Sources and Timing Chart



It takes at most 2.5T (T stands for the count clock cycle) to load a count value after an activation trigger is input.

■ Examples of Methods of All-L and All-H PPG Output

Figure 6.2-6 Methods of All-L and All-H PPG Output



6.3 Precautions on Using the PPG Timer

This section gives notes on using the PPG timer.

■ Precautions on Using the PPG Timer

- If the device attempts to set and clear the interrupt request flag at the same time, the flag is set and the clear operation becomes ineffective.
- The settings of bit11 and bit10 (count clock select bits CKS1 and CKS0) of the PPG control register are reflected immediately after data is written to the bits. Change the settings of the bits when counting stops.
- If the device attempts to load and count the PPG down counter (PPGC: 16-bit down counter) at the same time, the load operation takes precedence.

CHAPTER 7

16-BIT PULSE WIDTH COUNTER

This chapter gives an overview of the 16-bit pulse width counter and explains the register configuration and functions and the counter operation.

7.1 Overview of the PWC Timer

The 16-bit pulse width counter uses a 16-bit up counter to measure the pulse width of externally input signals.
The 16-bit pulse width counter consists of a 16-bit up counter, three 8-bit control registers, a PWC data register, PWC upper data register, and a low-pass filter (LPF).

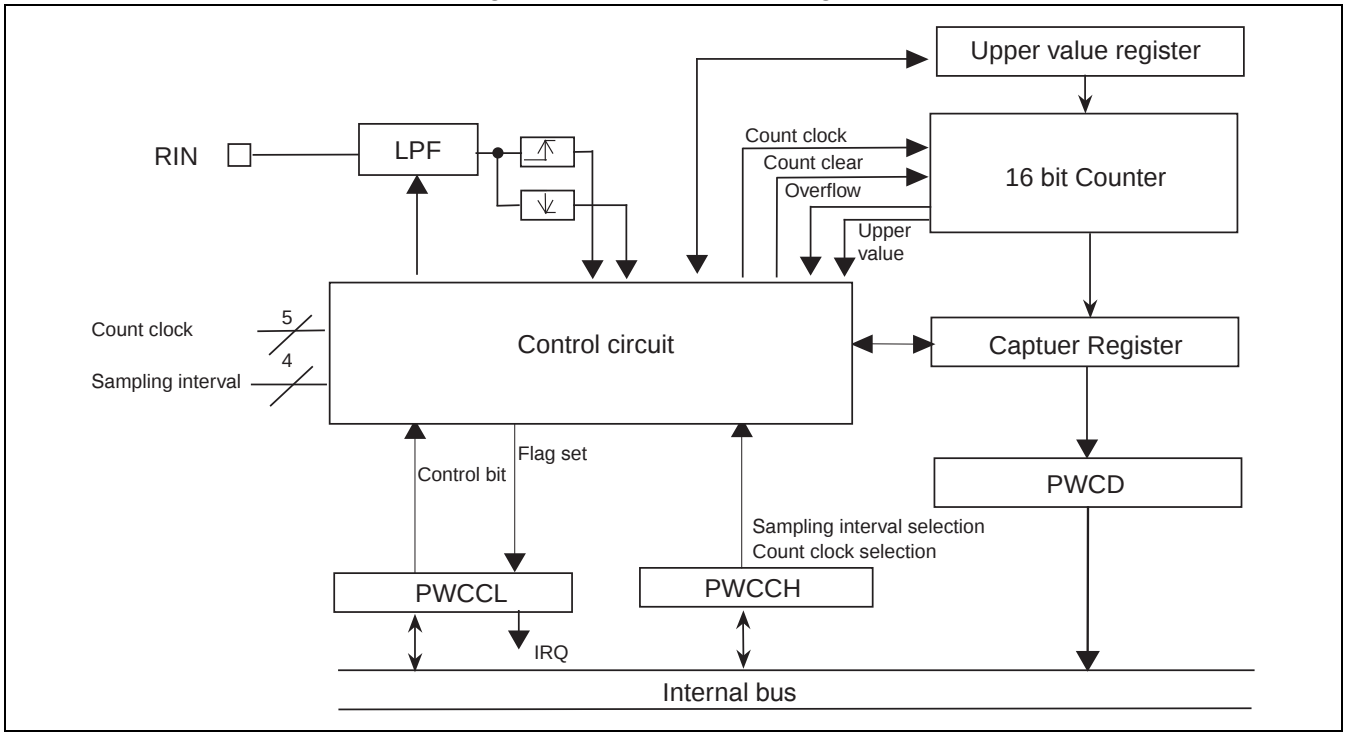
- **Interrupt request generation during data register transfer**

■ Registers of the 16-Bit Pulse Width Counter

Address	bit15	bit8 bit7	bit0	
$0000D0_H$	PWCCL		PWCCH	PWC control register
$0000D4_H$	PWCD			PWC data register
$0000D8_H$	PWCC2		Reserved	PWC control register
$0000DC_H$	PWCUD			PWC upper value setting register

■ Block Diagram

Figure 7.1-1 PWC Block Diagram



■ PWC Control Register (PWCCL)

Figure 7.1-2 Bit Configuration of PWC Control Register (PWCCL)

PWCCL

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000D0_H$		INT	INTE	OVFL	OVFLE	-	-	Reserved	ST	$0000\ --00_B$
		R/W	R/W	R/W	R/W	-	-	-	R/W	

R/W: Readable/writable
-: Unused

[bit7] INT

This bit is a flag that indicates that capture data has been transferred to the PWC data register. When a capture data transfer interrupt request is enabled (bit6: INTE = 1) and this bit is set, an interrupt request is generated.

The read modify write (RMW) instruction is read as "1".

0	Interrupt source is cleared.
1	Capture data is available.

[bit6] INTE

This bit is the capture data transfer request interrupt enable bit.

0	Interrupt request is disabled.
1	Interrupt request is enabled.

[bit5] OVFL

This bit is a flag that indicates that the 16-bit up counter has overflowed from $FFFF_H$ to 0000_H. When an overflow interrupt request is enabled (bit4: OVFLE = 1) and this bit is set, an interrupt request is generated.

The read modify write (RMW) instruction is read as "1".

0	Interrupt source is cleared.
1	An overflow occurs.

[bit4] OVFLE

This bit is the overflow interrupt request enable bit.

0	Interrupt request is disabled.
1	Interrupt request is enabled.

[bit3, bit2] Unused bits

[bit1] Reserved: Reserved bit

This bit is a reserved bit. Be sure to write "0" at writing.

[bit0] ST

This bit is the PWC start bit.

0	PWC stops.
1	PWC operates.

■ PWC Control Register (PWCCH)

Figure 7.1-3 Bit Configuration of PWC Control Register (PWCCH)

PWCCH

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000D0_H$		TEST1	TEST0	-	CSLF1	CSLF0	CS2	CS1	CS0	$00\text{-}00000_B$
		R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable
-: Unused

[bit7, bit6] TEST1, TEST0

These bits are test bits.

Be sure to write "0" at writing.

[bit5] Unused bit

[bit4, bit3] CSLF1, CSLF0

These bits are used to select the LPF sampling interval from the followings.

CSLF1	CSLF0	Sampling Interval
0	0	$\phi \times 2^6$
0	1	$\phi \times 2^8$
1	0	$\phi \times 2^{10}$
1	1	$\phi \times 2^{12}$

(ϕ is the cycle of the system base clock.)

[bit2, bit1, bit0] CS2, CS1, CS0

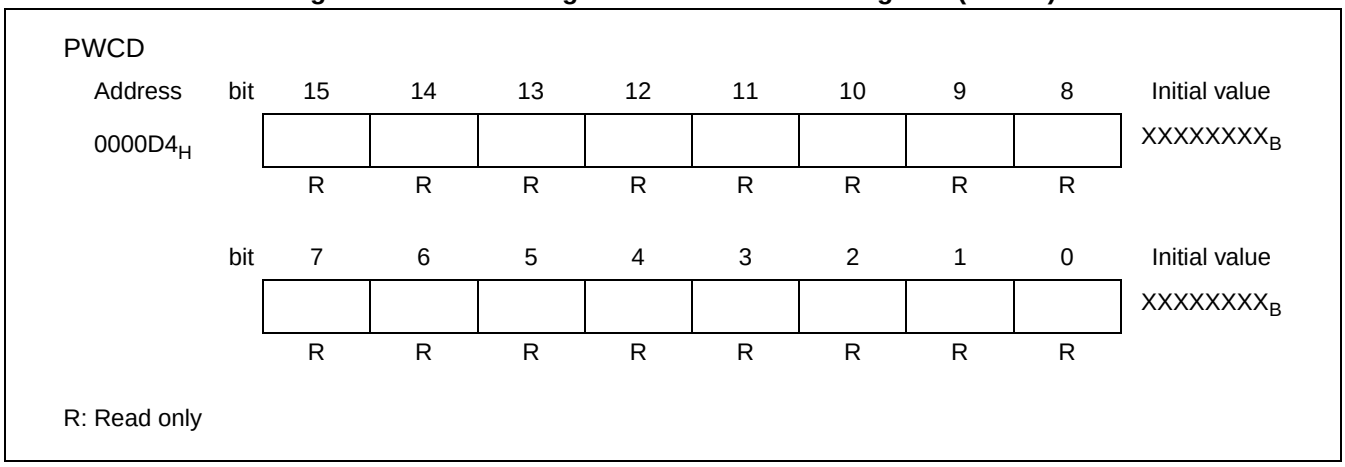
These bits are used to select the internal count clock as shown below.

CS2	CS1	CS0	Count clock selection
0	0	0	ϕ
0	0	1	$\phi \times 2^6$
0	1	0	$\phi \times 2^8$
0	1	1	$\phi \times 2^{10}$
1	0	0	$\phi \times 2^{12}$

(ϕ is the cycle of the system base clock.)

■ PWC Data Register (PWCD)

Figure 7.1-4 Bit Configuration of PWC Data Register (PWCD)



The PWC data register (PWCD) stores the measured value of the pulse width. Only the edge of input signal is captured the capture value.

When the overflow is performed and the upper value is exceeded, this register does not capture.

■ PWC Control Register 2 (PWCC2)

Figure 7.1-5 Bit Configuration of PWC Control Register 2 (PWCC2)

PWCC2

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000D8_H$		UPINT	UPINTE	LOW	-	-	-	-	-	$000-----_B$
		R/W	R/W	R/W	-	-	-	-	-	

R/W: Readable/writable
-: Undefined

[bit7] UPINT

This bit is a flag that indicates that the setting value of upper register has counted. When the upper value interrupt request is enabled (bit6: UPINTE=1) and this bit is set, an interrupt request is generated.

The read modify write (RMW) instruction is read as "1".

0	Interrupt source is cleared. (Initial value)
1	Upper value over count is available.

[bit6] UPINTE

This bit is an upper value interrupt request enable bit. Set to this bit to "1" and compare the counter value and the upper setting register.

0	Interrupt request is disabled. (Initial value)
1	Interrupt request is enabled.

[bit5] LOW

The bit represents that the capture value in the data register is indicated "L" width.

0	"H" width measurement is completed (Initial value)
1	"L" width measurement is completed.

[bit4 to bit0] Unused bit

■ PWC Upper Value Setting Register (PWCUD)

Figure 7.1-6 Bit Configuration of PWC Upper Value Setting Register (PWCUD)

PWCUD

Address	bit	15	14	13	12	11	10	9	8	Initial value
$0000DC_H$										$XXXXXXXX_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	bit	7	6	5	4	3	2	1	0	Initial value
										$XXXXXXXX_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

This register stores the upper value of a pulse width measurement.

This register is corresponding to each width regardless of H and L width, the pulse which exceeds the upper value is measured and the UPINT bit of PWCC2 register is set. When this register exceeds the counter value, the count is continued and is not stopped. Therefore the initial value of this register is undefined, writing "1" to the UPINTE bit of PWCC2 register, and write the upper value before compare it.

7.2 Operation of the PWC Timer

The 16-bit pulse width counter consists of a 16-bit up counter, three 8-bit control registers, a PWC data register, PWC upper value setting register, and an LPF. This counter measures the pulse width. One of five count clocks can be selected. The basic operation is described below.

■ Operation Overview

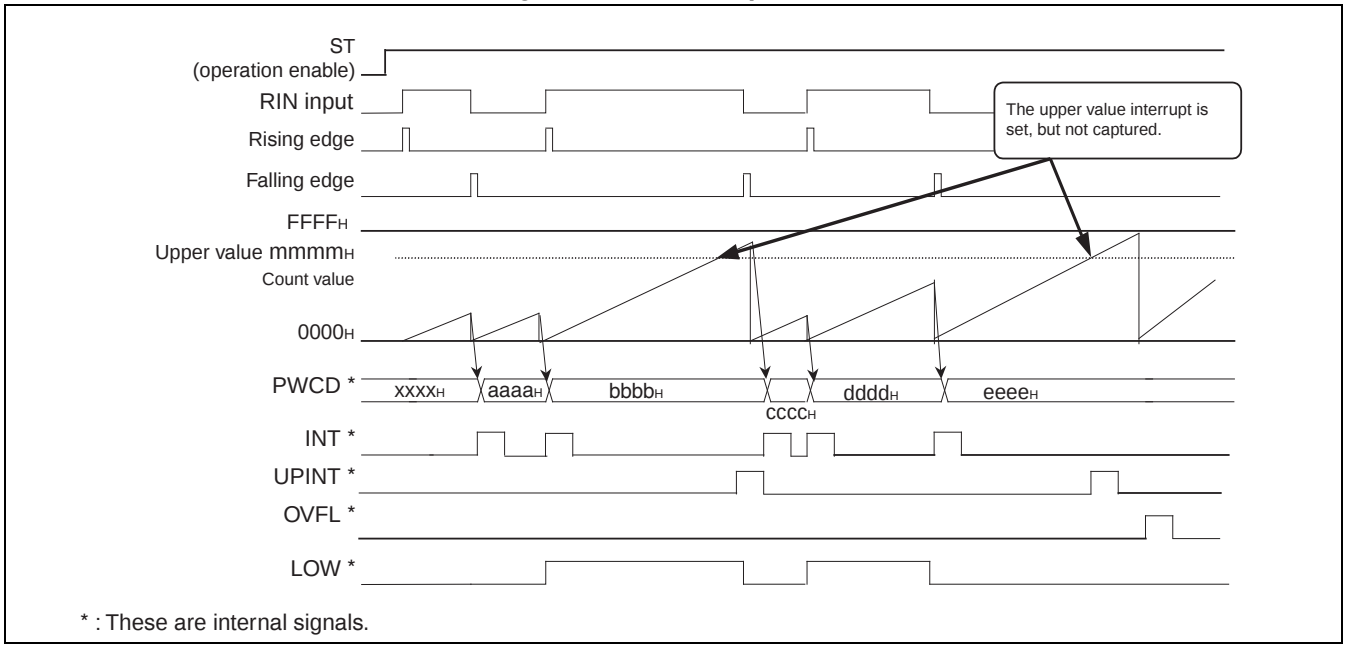
● Pulse width count operation

The PWC captures the counter value and clears the counter at the rising and falling edge of the RIN signal. The cleared counter continues counting unchanged. When the count value is captured, the PWC generates an interrupt.

When the counter value changes from $FFFF_H$ to 0000_H, the PWC generates an overflow interrupt.

Figure 7.2-1 shows the operation of the 16-bit pulse width counter.

Figure 7.2-1 PWC Operation



Note: The first edge (ST=1) is not captured after the operation enables.

■ Count Clock Selection

One of five count clocks can be selected.

Selectable count clock is shown as follow.

CS2	CS1	CS0	Count clock selection	PLL frequency multiply by 2 (33 MHz)	PLL off (Source oscillation 16.5 MHz)
0	0	0	CLKP	30 ns	61 ns
0	0	1	$\phi \times 2^{6}$ *	1.9 µs	3.8 µs
0	1	0	$\phi \times 2^{8}$ *	7.8 µs	15.5 µs
0	1	1	$\phi \times 2^{10}$ *	31.0 µs	62.1 µs
1	0	0	$\phi \times 2^{12}$ *	124.1 µs	248.2 µs

(CLKP is the peripheral clock. ϕ is the cycle of the system base clock.)

■ LPF Sampling Intervals

The LPF sampling intervals can be selected as follow.

CSLF1	CSLF0	Sampling interval	PLL frequency multiply by 2 (33 MHz)	PLL off (Source oscillation 16.5 MHz)
0	0	$\phi \times 2^{6}$ *	1.9 µs	3.8 µs
0	1	$\phi \times 2^{8}$ *	7.8 µs	15.5 µs
1	0	$\phi \times 2^{10}$ *	31.0 µs	62.1 µs
1	1	$\phi \times 2^{12}$ *	124.1 µs	248.2 µs

(ϕ is the cycle of the system base clock.)

*: Caution of setting

The PWC operation clock is CLKP. The count clock and the LFP sampling clock operate using ϕ.

Therefore, it does not operate correctly when the PWC operation clock is not faster than the count clock and the LFP sampling clock.

Cycle: PWC operation clock $\times$ 4 < count clock

PWC operation clock $\times$ 4 < keep the LFP sampling clock condition.

Example: at CLKP: 33 MHz $\rightarrow$ 30 ns $\times$ 4 < count clock ($\phi \times 2^{6}$: 1.9 µs) No problem

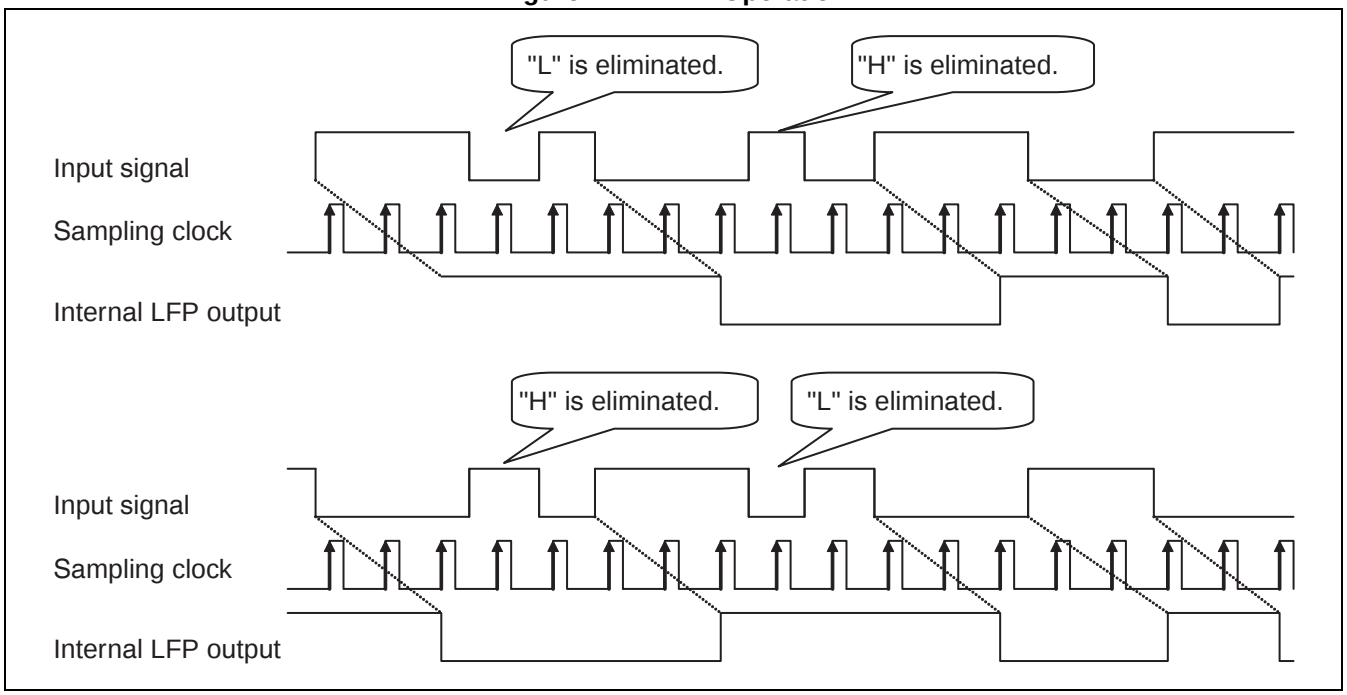
at CLKP: 16.5 MHz $\rightarrow$ 61 ns $\times$ 4 < count clock ($\phi \times 2^{6}$: 1.9 µs) No problem

at CLKP: 8.25 MHz $\rightarrow$ 121 ns $\times$ 4 < count clock ($\phi \times 2^{6}$: 1.9 µs) No problem

at CLKP: 4.13 MHz $\rightarrow$ 242 ns $\times$ 4 < count clock ($\phi \times 2^{6}$: 1.9 µs) No problem

at CLKP: 0.26 MHz $\rightarrow$ 3879 ns $\times$ 4 < count clock ($\phi \times 2^{6}$: 1.9 µs) Setting prohibited

Figure 7.2-2 LFP Operation



■ Interrupt Request Generation

The 16-bit pulse width counter can generate the following three interrupt requests:

- Capture data transfer interrupt request

  When capture data is transferred to the PWC data register, the interrupt flag is set. When interrupt requests are enabled, an interrupt request is generated.

- Counter overflow interrupt request

  When the counter value overflows from $FFFF_H$ to 0000_H during measurement, the overflow flag is set. When interrupt requests are enabled, an interrupt request is generated.
  Capture is not performed in overflow.

- Interrupt request which counts exceeding the value of upper register during counting
  When the counter value is larger than the upper setting register during measurement, the flag is set. When interrupt requests are enabled, an interrupt request is generated.

CHAPTER 8

MULTIFUNCTION TIMER

This chapter gives an overview of the multifunction timer and explains the register configuration and functions and the timer operation.

8.1 Overview of the Multifunction Timer

The multifunction timer consists of four channels for a 16-bit up counter. The multifunction timer has the following features:

- **A low-pass filter reduces noise that is below the amplitude of the set clock.**
- **The pulse width can be measured according to settings using seven types of clock signals.**
- **An event count from pin input is available.**
- **An interval timer that uses seven types of clocks and external input clocks is available.**
- **An HSYNC counter is available.**

■ Registers

Figure 8.1-1 Registers List of Multifunction Timer

Address			
$0000F0_H$	T0LPCR	T0CCR	(R/W)
$0000F2_H$	T0TCR	T0R	(R/W)
$0000F4_H$	T0DRR		(R/W)
$0000F6_H$	T0CRR		(R/W)
$0000F8_H$	T1LPCR	T1CCR	(R/W)
$0000FA_H$	T1TRR	T1R	(R/W)
$0000FC_H$	T1DRR		(R/W)
$0000FE_H$	T1CRR		(R/W)
000100_H	T2LPCR	T2CCR	(R/W)
000102_H	T2TRR	T2R	(R/W)
000104_H	T2DRR		(R/W)
000106_H	T2CRR		(R/W)
000108_H	T3LPCR	T3CCR	(R/W)
$00010A_H$	T3TRR	T3R	(R/W)
$00010C_H$	T3DRR		(R/W)
$00010E_H$	T3CRR		(R/W)
000110_H	TMODE		(R/W)

■ Block Diagram

Figure 8.1-2 Block Diagram of the Multifunction Timer (Simple)

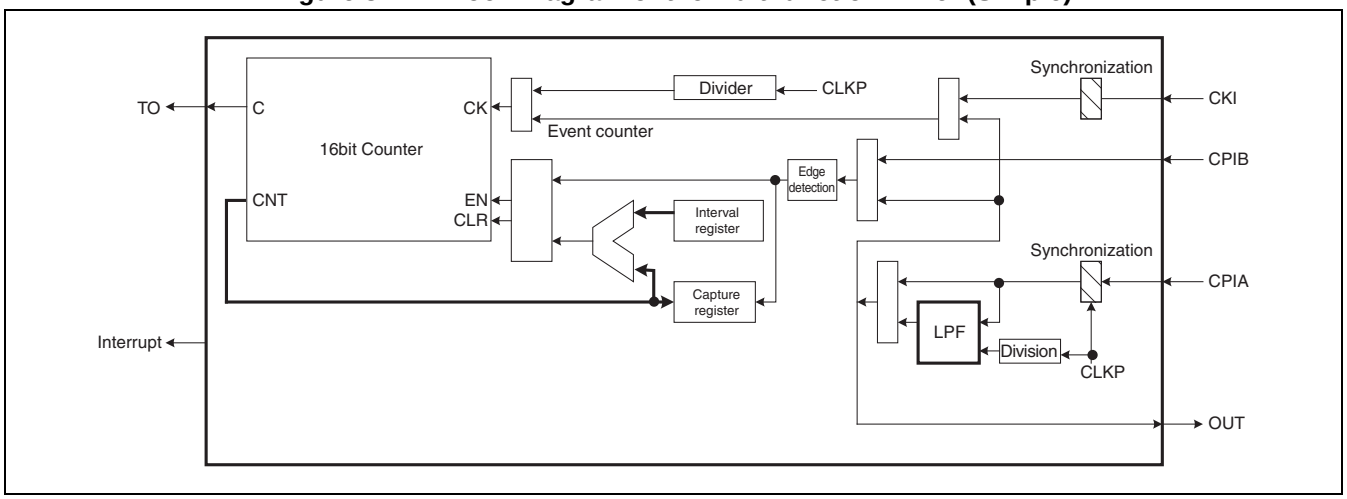
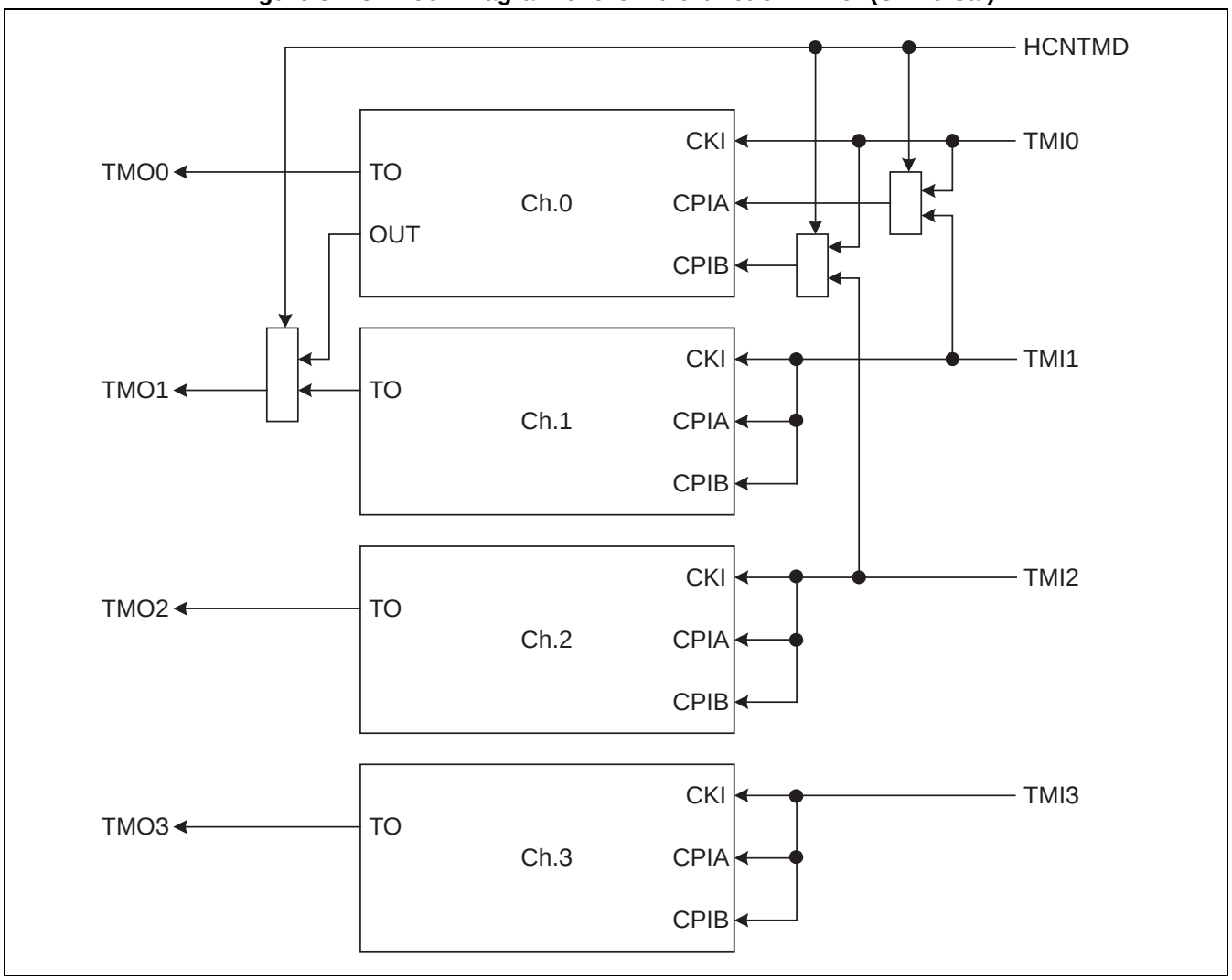


Figure 8.1-3 Block Diagram of the Multifunction Timer (Universal)



8.2 Registers of the Multifunction Timer

■ TxLPCR (Low-Pass Filter Control Register)

Figure 8.2-1 Bit Configuration of TxLPCR (Low-Pass Filter Control Register)

TxLPCR

Address	bit	15	14	13	12	11	10	9	8	Initial value
ch.0 $0000F0_H$		-	-	-	-	-	FCx1	FCx0	FxEN	$-----000_B$
ch.1 $0000F8_H$		-	-	-	-	-	R/W	R/W	R/W	
ch.2 000100_H										
ch.3 000108_H										

R/W: Readable/writable
-: Unused

The low-pass filter control register (TxLPCR) sets the low-pass filter for input pins. The low-pass filter control register (TxLPCR) can be 8-bit accessed. Because this filter reduces noise logically, the delay between the output waveform and the input waveform is the noise reduction width plus two cycles.

[bit15 to bit11] Unused bits

[bit10, bit9] FCx1, FCx0 (filter clock select flag)

These bits are used to select the operating clock for the LPF.

Table 8.2-1 Operating Clock Selection

FCx1	FCx0	Clock cycle	Noise reduction width (@33MHz)
0	0	$\phi \times 2^2$	0.12 µs [Initial value]
0	1	$\phi \times 2^3$	0.24 µs
1	0	$\phi \times 2^4$	0.48 µs
1	1	$\phi \times 2^5$	0.97 µs

[bit8] FxEN (filter enable flag)

This bit specifies whether the filter is used.

0	The filter is not used [initial value].
1	The filter is used.

■ TxCCR (Capture Control Register)

Figure 8.2-2 Bit Configuration of TxCCR (Capture Control Register)

TxCCR

Address	bit	7	6	5	4	3	2	1	0	Initial value
ch.0 0000F1$_H$ ch.1 0000F9$_H$ ch.2 000101$_H$ ch.3 000109$_H$		CPF	Reserved	CPST	CPED	CPIE	CPOV	CPMD	CPIS	0-000000$_B$
		R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable
-: Unused

The capture control register (TxCCR) sets the count, edge, and interrupt in capture mode.

The capture control register can be 8-bit accessed. If this register is written to during operation (entire register ST = 1), the timer operation is unpredictable. Be sure to rewrite this register when it is stopped (ST = 0).

[bit7] CPF (capture edge detection flag)

This bit indicates that the capture end edge has been detected.

0	No capture edge [initial value]
1	Capture edge

Writing "1" to this bit has no effect.

Note: If data is written to this flag from the hardware and the CPU at the same time, writing from the hardware has priority.

[bit6] Reserved: Reserved bit

[bit5] CPST (capture start edge select flag)

This bit sets the polarity of the capture start edge.

0	Rising edge [initial value]
1	Falling edge

Note: When you specify the same edge as the polarity of capture end edge for a value of this bit, capture is restarted after the next edge from the end edge.

[bit4] CPED (capture end edge select flag)

This bit sets the polarity of the capture end edge.

0	Rising edge [initial value]
1	Falling edge

[bit3] CPIE (capture interrupt enable flag)

This bit enables capture interrupt at capture end.

0	Capture interrupts are disabled [initial value]
1	Capture interrupts are enabled

When this bit and CPF are both set to "1", an interrupt is sent to the CPU.

[bit2] CPOV (capture overflow detection flag)

This bit indicates that the counter has detected an overflow from $FFFF_H$ to 0000_H in the free-run mode of capture mode.

0	No capture overflow [initial value]
1	Capture overflow

Notes:

- Writing "1" to this bit has no effect.
- If data is written to this bit from the hardware and the CPU at the same time, writing from the hardware has priority.

[bit1] CPMD (capture count mode flag)

This bit sets the count mode of the capture counter.

0	Free-run mode [initial value]
1	Upper-limit compare mode

[bit0] CPIS (capture input select flag)

This bit is used to select the input signal for capture.

0	CPIA input is used [initial value].
1	CPIB input is used.

■ TxTCR (Timer Setting Register)

Figure 8.2-3 Bit Configuration of TxTCR (Timer Setting Register)

TxTCR

Address	bit	15	14	13	12	11	10	9	8	Initial value
ch.0 0000F2$_H$ ch.1 0000FA$_H$ ch.2 000102$_H$ ch.3 00010A$_H$		TCF	TSES	TCC	TIE	CINV	TCS2	TCS1	TCS0	00000000$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

The timer setting register (TxTCR) controls the timer operation.

The timer setting register (TxTCR) can be 8-bit accessed.

If this register is rewritten during operation (entire register ST = 1), the timer operation is unpredictable. Be sure to rewrite this register when it is stopped (ST = 0).

[bit15] TCF (timer compare match detection flag)

This bit indicates that a timer compare match has been detected.

0	No compare match [initial value]
1	Compare match

Notes:

- Writing "1" to this bit has no effect.
- If data is written to this bit from the hardware and the CPU at the same time, writing data from the hardware has priority.

[bit14] TSES (timer start edge select flag)

This bit sets the start edge of the timer.

0	Rising edge [initial value]
1	Falling edge

[bit13] TCC (timer count clear setting flag)

This bit specifies that the counter is cleared when a timer compare match is detected.

0	Count clear [initial value]
1	No count clear

[bit12] TIE (timer interrupt enable flag)

This bit enables timer interrupts.

0	Timer interrupts are disabled [initial value].
1	Timer interrupts are enabled.

When this bit and TCF are both set to "1", an interrupt is sent to the CPU.

[bit11] CINV (timer clock invert flag)

This bit inverts the timer input clock signal from the external pin.

0	The count increments at the rising edge of the clock [initial value].
1	The count increments at the falling edge of the clock.

[bit10 to bit8] TCS2 to TCS0 (timer clock select flag)

These bits are used to select the timer clock.

Note: To use the event count mode, set these bits to "111_B".

TCS bit			Clock and source to be selected	
TCS2	TCS1	TCS0	Division ratio	Cycle (@33MHz)
0	0	0	$\phi \times 2^3$	0.24 µs
0	0	1	$\phi \times 2^5$	0.96 µs
0	1	0	$\phi \times 2^7$	3.88 µs
0	1	1	$\phi \times 2^9$	15.5 µs
1	0	0	$\phi \times 2^{10}$	31.0 µs
1	0	1	$\phi \times 2^{12}$	124.1 µs
1	1	0	$\phi \times 2^{14}$	496.5 µs
1	1	1	External clock	

■ TxR (Entire Timer Control Register)

Figure 8.2-4 Bit Configuration of TxR (Entire Timer Control Register)

TxR

Address	bit	7	6	5	4	3	2	1	0	Initial value
ch.0 0000F3$_H$ ch.1 0000FB$_H$ ch.2 000103$_H$ ch.3 00010B$_H$		-	-	-	TST2	TST1	MD1	MD0	ST	---00000$_B$
		-	-	-	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable
-: Unused

The entire timer control register (TxR) controls the entire timer operation.

The entire timer control register (TxR) can be 8-bit accessed.

[bit7 to bit5] Unused bits

[bit4, bit3] TST2, TST1 (test bits)

Always write 0 to these bits.

[bit2, bit1] MD1, MD0 (timer select flag)

These bits are used to select the timer operation.

MD1	MD0	Selection mode
0	0	Interval timer [initial value]
0	1	Event count
1	0	Capture
1	1	Setting prohibited

[bit0] ST (timer operation start flag)

0	Timer operation is disabled [initial value].
1	Timer operation is enabled.

Set CPIE or TIE to "0" before ST=0.

When ST=0 and the interrupt factor occurs at the same time, even though ST=0, the interrupt occurs.

■ TxDRR (Timer Compare Data Register)

Figure 8.2-5 Bit Configuration of TxDRR (Timer Compare Data Register)

TxDRR

Address	bit	15	14	13	12	11	10	9	8	Initial value
ch.0 0000F4$_H$ ch.1 0000FC$_H$ ch.2 000104$_H$ ch.3 00010C$_H$		D15	D14	D13	D12	D11	D10	D9	D8	XXXXXXXX$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

bit	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

The timer compare data register (TxDRR) stores timer compare data. The timer compare data register (TxDRR) compares data in this register and the value of the timer counter and then indicates whether there is a compare match.

To use this register, set the interval time in the timer mode and the event count in the external event mode. Enter the upper count limit in capture mode. This register cannot be 8-bit accessed. Setting "0" in this register results in 2^{16} counts.

■ TxCRR (Capture Data Register)

Figure 8.2-6 Bit Configuration of TxCRR (Capture Data Register)

TxCRR

Address	bit	15	14	13	12	11	10	9	8	Initial value
ch.0 0000F6$_H$ ch.1 0000FE$_H$ ch.2 000106$_H$ ch.3 00010E$_H$		D15	D14	D13	D12	D11	D10	D9	D8	XXXXXXXX$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

bit	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

The capture data register (TxCRR) is used to read the captured value.

This register can be written to enter the initial value.

This register cannot be 8-bit accessed.

■ TMODE

Figure 8.2-7 Bit Configuration of TMODE

TMODE

Address	bit	15	14	13	12	11	10	9	8	Initial value
000110_H		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	bit	7	6	5	4	3	2	1	0	Initial value
		Reserved	Reserved	Reserved	Reserved	Reserved	HCNTMD	Reserved	Reserved	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

TMODE is a register to set the HSYNC counter mode.

This register is allowed an access with 16-bit.

[bit15 to bit3] Reserved: Reserved bits

These bits are reserved. Write "0" to these bits.

[bit2] HCNTMD:

This bit sets the HSYNC counter mode.

0: Normal mode

1: HSYNC counter mode

The counter ch.0 is used in the HSYNC counter mode.

When input HSYNC to TMI0 and VSYNC to TMI1, the counter is used to set to the capture mode.

Be sure to specify different edge for capture start and end edges.

[bit1, bit0] Reserved: Reserved bits

These bits are reserved. Write "0" to these bits.

8.3 Multifunction Timer Operation

The multifunction timer has the following operating modes:

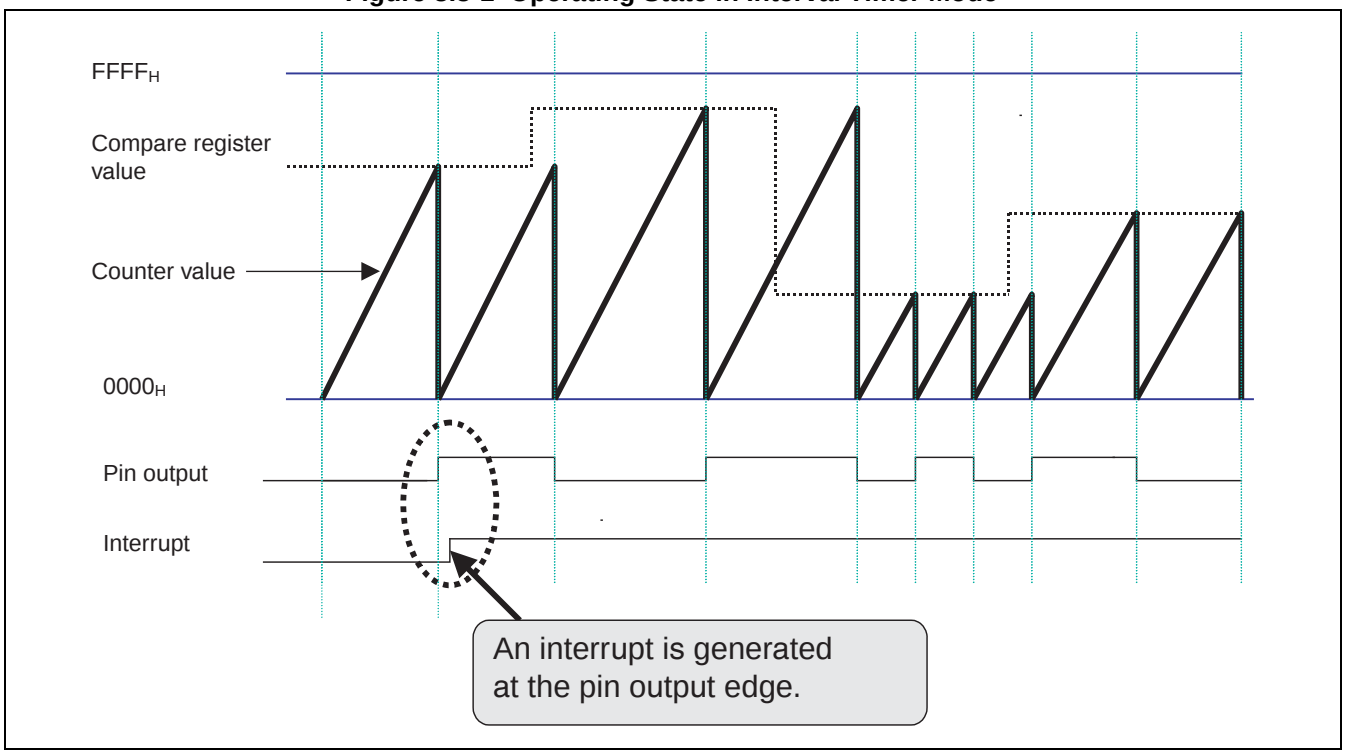
- **Interval timer**
- **Event count**
- **Capture mode**

This section gives an overview of operation in each mode. The initial value of the toggle output of this module is 0 in all modes.

■ Interval Timer Mode

In the interval timer mode, the multifunction timer has functions that use the clock selected from the seven types of clock sources for the timer count and toggle output and generate an interrupt if the counter value and the compare register value match. The following figure shows the multifunction timer operating state in interval timer mode.

Figure 8.3-1 Operating State in Interval Timer Mode

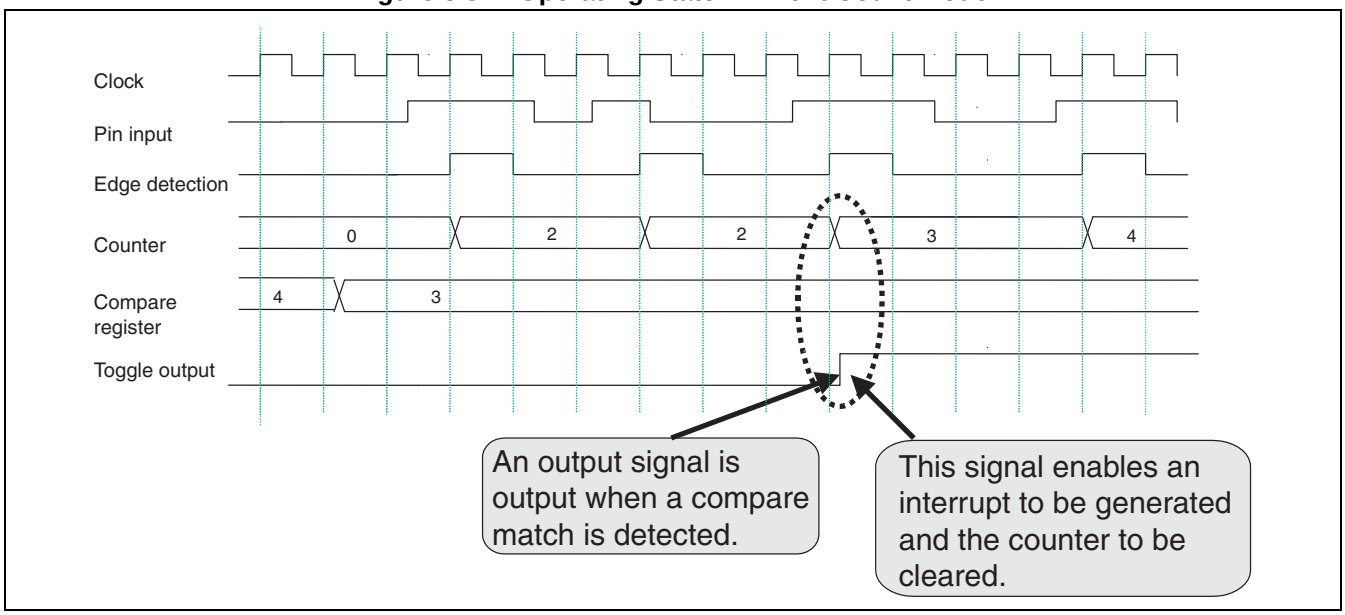


■ Event Count Mode

In the event count mode, the multifunction timer detects the pin input edge and counts the edges the specified number of times.

When the counter value and the compare register value match, TCF is set to "1". If TIE is set to "1" at this time, an interrupt is generated. When a compare match is detected, the counter can be cleared.

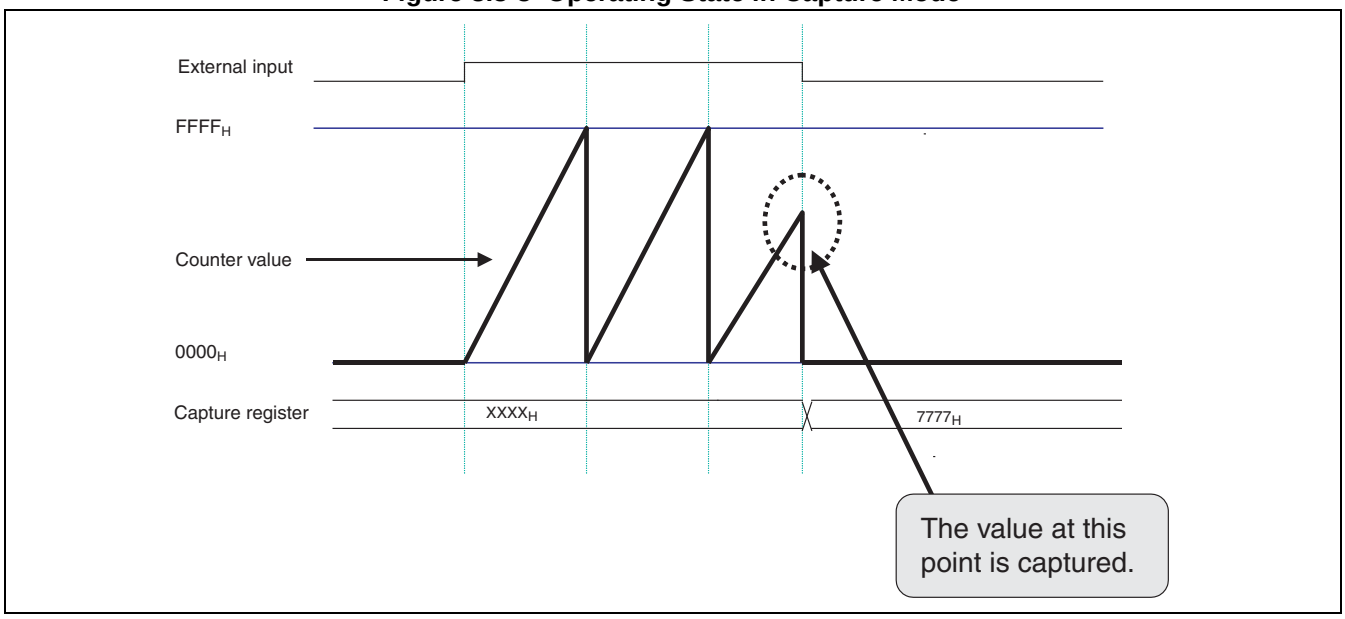
Figure 8.3-2 Operating State in Event Count Mode



■ Capture Mode

In the capture mode, the width between the rising or falling edges of an external pin input can be measured. The clock for measurement can be selected from the seven types of clock sources. The start and end edges can be selected from either the rising or falling edge. In free-run mode, the count value is captured when the end edge is reached. In the upper-limit compare mode, an upper limit is input if the count value and the upper-limit compare value match before the end edge is reached. Otherwise, the captured value at the end edge is input. The following figure shows an example of starting the count at the rising edge and ending it at the falling edge in free-run mode.

Figure 8.3-3 Operating State in Capture Mode

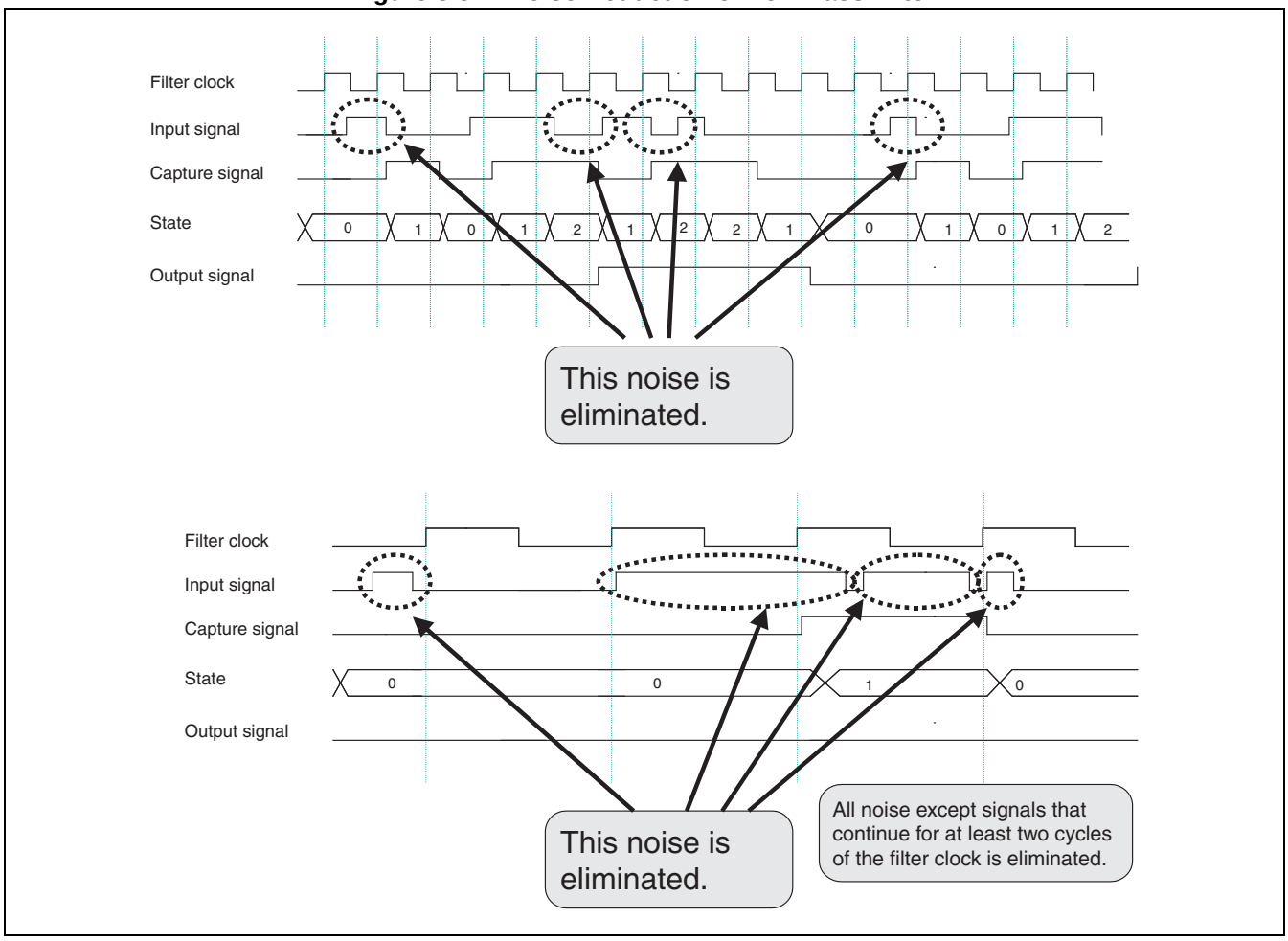


■ Low-Pass Filter

This module contains a low-pass filter for each external pin input.

This filter enables logical reduction of noise in four types of widths.

Figure 8.3-4 Noise Reduction of Low-Pass Filter



CHAPTER 9
OTHER TIMERS

This chapter explains the main oscillation stabilization wait timer, interval timer, and watch timer.

9.1 Main Oscillation Stabilization Wait Timer

The main oscillation stabilization wait timer is a 23-bit counter that is synchronized with the main clock to count up. It includes an interval timer function that continues to generate interrupts in regular time intervals.
This timer uses the main clock to secure the oscillation stabilization wait time, when the main oscillation is temporarily suspended during sub clock operation and restarted using OSCDS1 (bit8) in OSCCR (oscillation control register).

■ Interval Times of the Main Oscillation Stabilization Wait Timer

Table 9.1-1 shows types of the interval times. The following 3 interval times are available for selection.

Table 9.1-1 Interval Times of Main Oscillation Stabilization Wait Timer

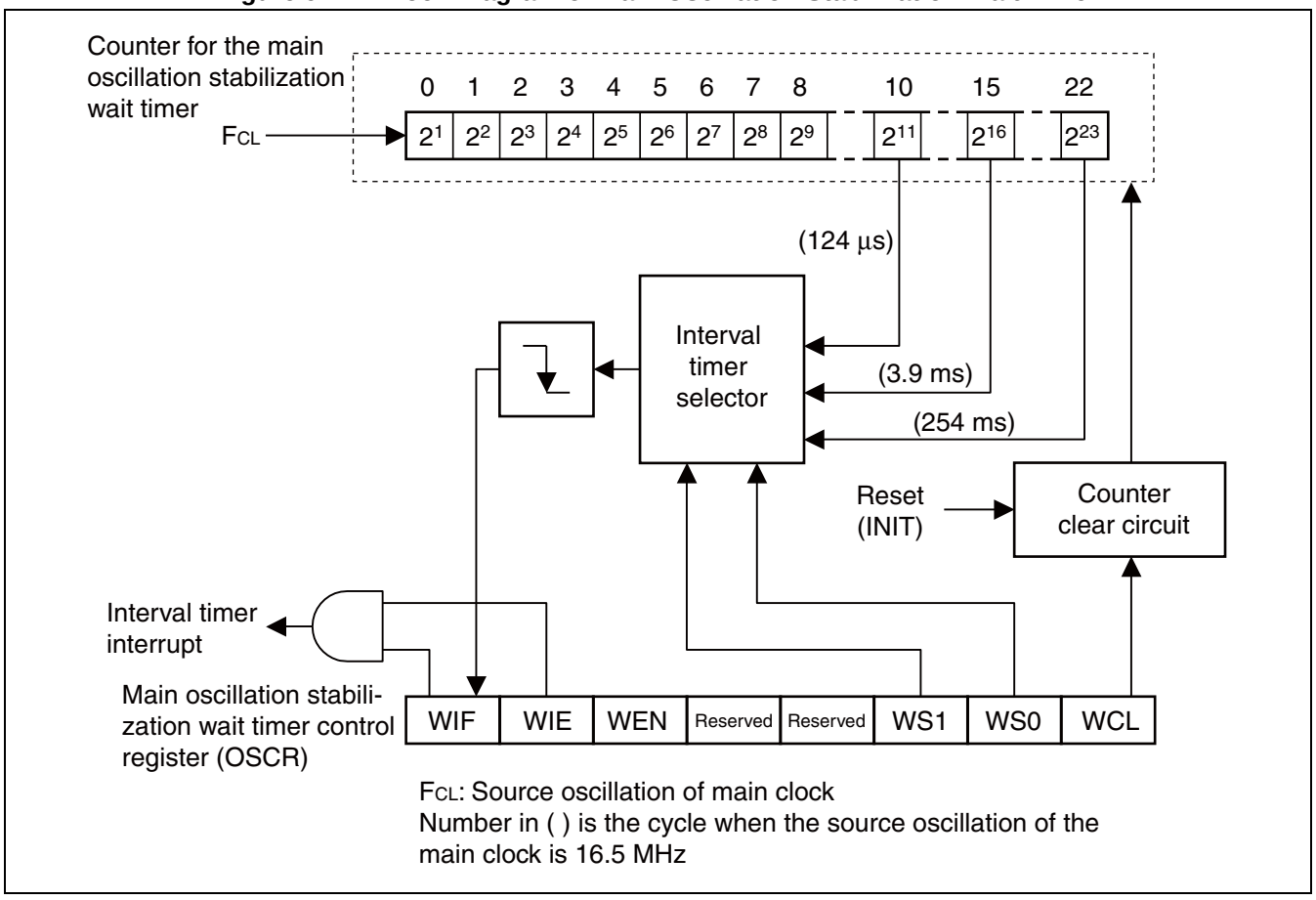
Main clock cycle	Interval time
$1/F_{CL}$ (approx. 60 ns)	$2^{11}/F_{CL}$ (124 µs)
	$2^{16}/F_{CL}$ (3.9 ms)
	$2^{22}/F_{CL}$ (254 ms)

Note: F_{CL} represents the main clock oscillation frequency (at F_{CL} = 16.5 MHz).

■ Block Diagram of the Main Oscillation Stabilization Wait Timer

Figure 9.1-1 shows a block diagram of the main oscillation stabilization wait timer.

Figure 9.1-1 Block Diagram of Main Oscillation Stabilization Wait Timer



● Main oscillation stabilization wait timer

This timer is a 23-bit up-counter that uses the source oscillation of the main clock for its count clock.

● Counter clear circuit

This circuit clears the counter at a reset (INIT), other than the OSCR register setting (WCL=0).

● Interval timer selector

Out of 3 different division outputs of the counter for the main oscillation stabilization wait timer, this circuit selects one to be used for the interval timer. The falling edge of the selected division output is used as the interrupt source.

● Main oscillation stabilization wait register (OSCR)

This register selects the interval time, clears the counter, controls interrupts and checks the interrupt state.

■ Explanation of the Main Oscillation Stabilization Wait Timer Register

The configuration of the main oscillation stabilization wait timer register is shown below.

Figure 9.1-2 Bit Configuration of Main Oscillation Stabilization Wait Control Register

OSCR

Address	bit	15	14	13	12	11	10	9	8
000490_H		WIF	WIE	WEN	Reserved	Reserved	WS1	WS0	WCL
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)		0	0	0	0	0	0	0	0
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable

[bit15] WIF: Timer interrupt flag

WIF is the flag for main oscillation stabilization wait interrupt requests.

It is set to "1" at the falling edge of the selected division output for the interval timer.

A main oscillation stabilization interrupt request is output, when this bit and the interrupt request enable bit are set to "1".

Value	Description
0	No request for main oscillation stabilization interrupt [Initial value]
1	Request for main oscillation stabilization interrupt

- This bit is initialized to "0" by a reset (INIT).
- This bit is readable and writable. For write operation, however, only "0" can be written. Writing "1" does not change the bit value.
- Reading by read-modify-write (RMW) instruction always returns "1".

[bit14] WIE: Timer interrupt enable bit

WIE enables/disables the output of an interrupt request to the CPU. A main oscillation stabilization interrupt request is output, when this bit and the main oscillation stabilization interrupt request flag bit are set to "1".

Value	Description
0	Disables the output of main oscillation stabilization interrupt request. [Initial value]
1	Enables the output of main oscillation stabilization interrupt request.

- This bit is initialized to "0" by a reset (INIT).
- This bit is readable and writable.

[bit13] WEN: Timer operation enable bit

WEN enables the timer operation.

When this bit is set to "1", the timer performs count operation.

Value	Description
0	The timer stops. [Initial value]
1	The timer operates.

- This bit is initialized to "0" by a reset (INIT).
- This bit is readable and writable.

[bit12, bit11] Reserved: Reserved bits

These are reserved bits. Write "0" on writing (writing "1" is disabled).

Read value is undefined.

[bit10, bit9] WS1, WS0:Timer interval time selection bits

These bits select the cycle for the interval timer.

The cycle is selected from the following 3 output bits of the counter for the main oscillation stabilization wait timer.

WS1	WS0	Interval timer cycle (When F_{CL} = 16.5 MHz)
0	0	Prohibited setting [Initial value]
0	1	$2^{11}/F_{CL}$ (124 µs)
1	0	$2^{16}/F_{CL}$ (3.9 ms)
1	1	$2^{22}/F_{CL}$ (254 ms)

- These bits are initialized to "00_B" by a reset (INIT).
- These bits are readable and writable.
- To use the main oscillation stabilization wait time timer, write data to this register.

[bit8] WCL: Timer clear bit

When "0" is written to WCL, the oscillation stabilization wait timer is cleared to "0".

For write operation, only "0" can be written. Writing "1" has no effect on operation.

- Reading always returns "1".

■ Main Oscillation Stabilization Wait Interrupt

The counter for the main oscillation stabilization wait timer counts on the main clock, and sets the main oscillation stabilization wait interrupt request flag (WIF) to "1" when the set interval time has elapsed. In this case, if the interrupt request enable bit has been enabled (WIE=1), an interrupt request is generated to the CPU. However, if the oscillation of the main clock is stopped (see the next section "■ Operation of the Main Oscillation Stabilization Wait Timer"), the count operation also stops. As a result, no main oscillation stabilization wait interrupt is generated.

To clear an interrupt request, write "0" to the WIF flag in the interrupt processing routine.

Note that WIF is set at the falling edge of the specified division output, regardless of the WIE value.

Note:

When enabling the output of an interrupt request (WIE=1) after reset release or modifying WS1,WS0 bit, always clear WIF and WCL at the same time (WIF=WCL=0).

References:

- When WIF is set to "1", an interrupt request is generated as soon as WIE is enabled from the disabled state (0 → 1).
- WIF is not set, if the counter is cleared (WPCR:WCL=1) at the same time as an overflow occurs at the selected bit.

■ Operation of Interval Timer Function

The counter for the main oscillation stabilization wait timer counts up on the main clock. Under the following conditions, however, the count operation stops because the oscillation of the main clock stops.

- When WEN is set to "0"
- If the device enters stop mode when the main oscillation is set to stop in stop mode (bit0:OSCD1 in the standby control register STCR = 1), the count operation stops during stop mode.
  OSCD1 is initialized to "1" at a reset (INIT). Therefore, to operate the main oscillation stabilization wait timer even during stop mode, set OSCD2 to "0" before the device enters standby mode.
- When OSCDS1 (bit8) in OSCCR (oscillation control register) is set to "1" in sub clock mode, the main oscillation stops and the timer also stops the count operation.

When the counter is cleared (WCL=0), it starts count operation from "000000_H". Once it reaches "$7FFFFF_H$", it goes back to "000000_H" and continues to count. When a falling edge is generated at the selected division output for the interval timer, the main oscillation stabilization wait interrupt request bit (WIF) is set to "1". This means that a main oscillation stabilization wait timer interrupt request is generated at every selected interval time, based on the cleared time.

■ Operation of Clock Supply Function

The time-base counter is used to secure the oscillation stabilization wait timer after INIT or stop mode. However, to secure the oscillation stabilization wait time for the main clock when the sub clock is selected as the clock source, the main oscillation stabilization wait timer is used, as it operates on the main clock regardless of the clock source selection.

To perform the main clock oscillation stabilization wait from the main oscillation stop state in the sub clock operation, follow the procedure described below.

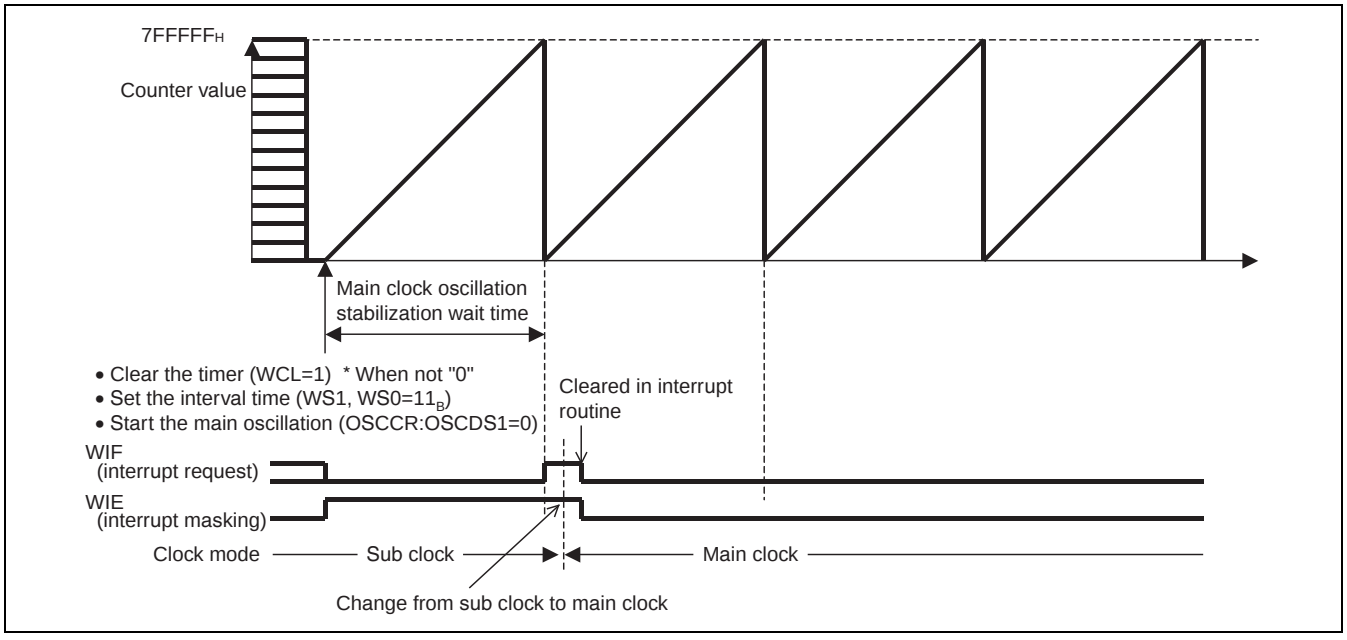
1) Set the time required to stabilize the oscillation of the main clock in WT1 and WT0 and clear the counter to "0" (WT1 and WT0 =oscillation stabilization wait time; write "0" to WCL).
   To perform processing after the completion of oscillation stabilization wait by an interrupt, also initialize the interrupt flag (write "0" to WIF and WIE).
2) Start the oscillation of the main clock (write "1" to OSCDS1 (bit8) in OSCCR).
3) Wait until the WIF flag is set to "1" by program.
4) Make sure that the WIF flag has been set to "1" and then perform the processing after the completion of the oscillation stabilization wait. If interrupts are enabled, an interrupt occurs when WIF is set to "1". In this case, perform the processing after the completion of the oscillation stabilization wait in the interrupt routine.

Also, when switching from the sub clock to main clock, make sure that WIF has been set to "1" beforehand, as described in 4). (If the clock is switched to the main clock without waiting until the oscillation stabilizes, an unstable clock signal is supplied through the device, and the succeeding operation is not guaranteed.)

■ Operation of the Main Oscillation Stabilization Wait Timer

Figure 9.1-3 shows the state of the counter during transition to the main clock when the main oscillation stabilization wait timer is activated.

Figure 9.1-3 State of Counter during Transition to Main Clock when Main Oscillation Stabilization Wait Timer is Activated



■ Notes on Using the Main Oscillation Stabilization Wait Timer

The oscillation stabilization wait time should be used for reference only, as the oscillation cycle is unstable immediately after the oscillation starts.

While the oscillation of the main clock is stopped, the counter is also stopped. Consequently, no main oscillation stabilization interrupt occurs. Therefore, to perform any processing using a main oscillation stabilization interrupt, do not stop the main oscillation.

If the WIF flag set request and the clearing to "0" from the CPU occur simultaneously, the flag set request will have higher priority; therefore, the clearing to "0" will be cancelled.

MB91313A Series

9.2 Watch Timer

The watch timer is a 21-bit free-run counter that is synchronized with the sub clock to count up. It includes an interval timer function that continues to generate interrupts in regular time intervals.

■ Interval Time of Watch Timer

The following 7 interval times are available for selection.

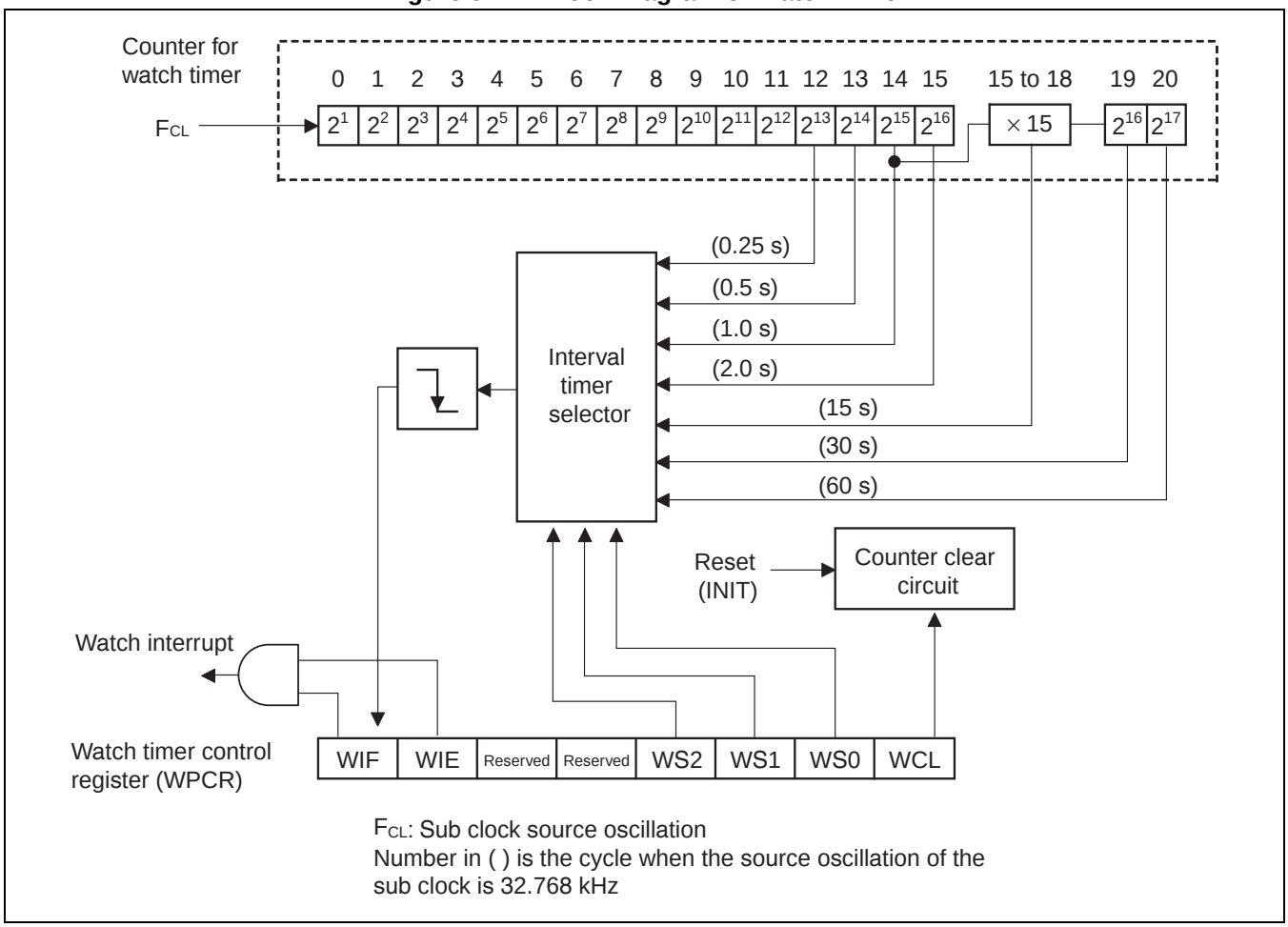
Table 9.2-1 Interval Time of Watch Timer

Sub clock cycle	Interval time
$1/F_{CL}$ (approx. 30.5 µs)	$2^{13}/F_{CL}$ (0.25 s)
	$2^{14}/F_{CL}$ (0.50 s)
	$2^{15}/F_{CL}$ (1.00 s)
	$2^{16}/F_{CL}$ (2.00 s)
	$15\times2^{15}/F_{CL}$ (15 s)
	$15\times2^{16}/F_{CL}$ (30 s)
	$15\times2^{17}/F_{CL}$ (60 s)

Note: F_{CL} is the sub clock oscillation frequency (at F_{CL} = 32.768 kHz).

■ Block Diagram of Watch Timer

Figure 9.2-1 Block Diagram of Watch Timer



[Watch timer]

This timer is a 21-bit up-counter that uses the source oscillation of the sub clock for its count clock.

[Counter clear circuit]

This circuit clears the counter at a reset (INIT), other than the WPCR register setting (WCL=0).

[Interval timer selector]

Out of 4 different division outputs of the counter for the watch timer, this circuit selects one to be used for the interval timer. The falling edge of the selected division output is used as the interrupt source.

[Watch timer control register (WPCR)]

This register selects the interval time, clears the counter, controls interrupts and checks the interrupt state.

MB91313A Series

■ Registers of Watch Timer

Figure 9.2-2 Bit Configuration of Registers of Watch Timer

WPCR

Address	bit	15	14	13	12	11	10	9	8
$00048C_H$		WIF	WIE	Reserved	Reserved	WS2	WS1	WS0	WCL
		R/W	R/W	W	W	R/W	R/W	R/W	W
Initial value (INIT)		0	0	0	0	0	0	0	0
Initial value (RST)		X	X	X	X	X	X	X	X

R/W: Readable/writable
W: Write only

[bit15] WIF (Watch timer Interrupt Flag)

This bit is the watch interrupt request flag.

It is set to "1" at the falling edge of the selected division output for the interval timer.

A watch interrupt request is output, when this bit and the interrupt request enable bit are set to "1".

0	No watch interrupt request (initial value)
1	Watch interrupt request

- This bit is initialized to "0" by a reset (INIT).
- This bit is readable and writable. For write operation, however, only "0" can be written. Writing "1" does not change the bit value. Reading by read-modify-write (RMW) instruction always returns "1".

[bit14] WIE (Watch timer Interrupt Enable)

WIE enables/disables the output of an interrupt request to the CPU. A watch interrupt request is output, when this bit and the watch interrupt request flag bit are set to "1".

0	Watch interrupt request output disabled (initial value)
1	Watch interrupt request output enabled

- This bit is initialized to "0" by a reset (INIT).
- This bit is readable and writable.

[bit13 to bit12] Reserved: Reserved bits

These are reserved bits. Write "0" on writing (writing "1" is disabled).

Read value is undefined.

[bit11 to bit9] WS2 to WS0 (Watch timer interval Select 2 to 0)

These bits select the cycle for the interval timer.

The cycle is selected from the following 7 output bits of the counter for the watch timer.

WS1	WS0	WS0	Interval timer cycle (When F_{CL} = 32.768 kHz)
0	0	0	2^{13} / F_{CL} (0.25 s) (Initial value)
0	0	1	2^{14} / F_{CL} (0.50 s)
0	1	0	2^{15} / F_{CL} (1.00 s)
0	1	1	2^{16} / F_{CL} (2.0 s)
1	0	0	15×2^{15} / F_{CL} (15 s)
1	0	1	15×2^{16} / F_{CL} (30 s)
1	1	0	15×2^{17} / F_{CL} (60 s)
1	1	1	Setting disabled

- These bits are initialized to "000_B" by a reset (INIT).
- These bits are readable and writable.

[bit8] WCL (Watch timer CLear)

When "0" is written to WCL, the watch timer is cleared to "0".

- For write operation, only "0" can be written. Writing "1" has no effect on operation.
- Reading always returns "1".

■ Watch Interrupt

The counter for the watch timer counts on the sub clock, and sets the watch interrupt request flag (WIF) to "1" when the set interval time has elapsed. In this case, if the interrupt request enable bit has been enabled (WIE=1), an interrupt request is generated to the CPU. However, if the oscillation of the sub clock is stopped (see "■ Operation of Interval Timer Function"), the count operation also stops. As a result, no watch interrupt is generated.

To clear an interrupt request, write "0" to the WIF flag in the interrupt processing routine.

Note that WIF is set at the falling edge of the specified division output, regardless of the WIE value.

Note:

When enabling the output of an interrupt request (WIE=1) after reset release or modifying WS1,WS0 bit, always clear WIF and WCL at the same time (WIF=WCL=0).

- When WIF is set to "1", an interrupt request is generated as soon as WIE is enabled from the disabled state (0 → 1).
- WIF is not set, if the counter is cleared (WPCR:WCL=1) at the same time as an overflow occurs at the selected bit.

■ Operation of Interval Timer Function

The counter for the watch timer always counts up during the sub clock oscillation. Under the following conditions, however, the count operation stops because the oscillation of the sub clock stops.

- When PLL2EN (bit11) of the clock source register CLKR is "0".
  PLL2EN is initialized to "0" at reset (INIT). When using the watch timer, write "1" to PLL2EN to start the oscillation of the sub clock.
- If the device enters stop mode when the sub oscillation is set to stop in stop mode (OSCD2 (bit1) in the standby control register STCR = 1), the count operation stops during stop mode. In this model, OSCD2 is initialized to "1" at a reset (INIT). Therefore, to operate the watch timer even during stop mode, set OSCD2 to "0" before the device enters standby mode.

When the counter is cleared (WCL=0), it starts count operation from "000000_H". Once it reaches "$1FFFFF_H$", it goes back to "000000_H" and continues to count. When a falling edge is generated at the selected division output for the interval timer, the watch interrupt request bit (WIF) is set to "1". This means that a watch interrupt request is generated at every selected interval time, based on the cleared time.

■ Operation of Clock Supply Function

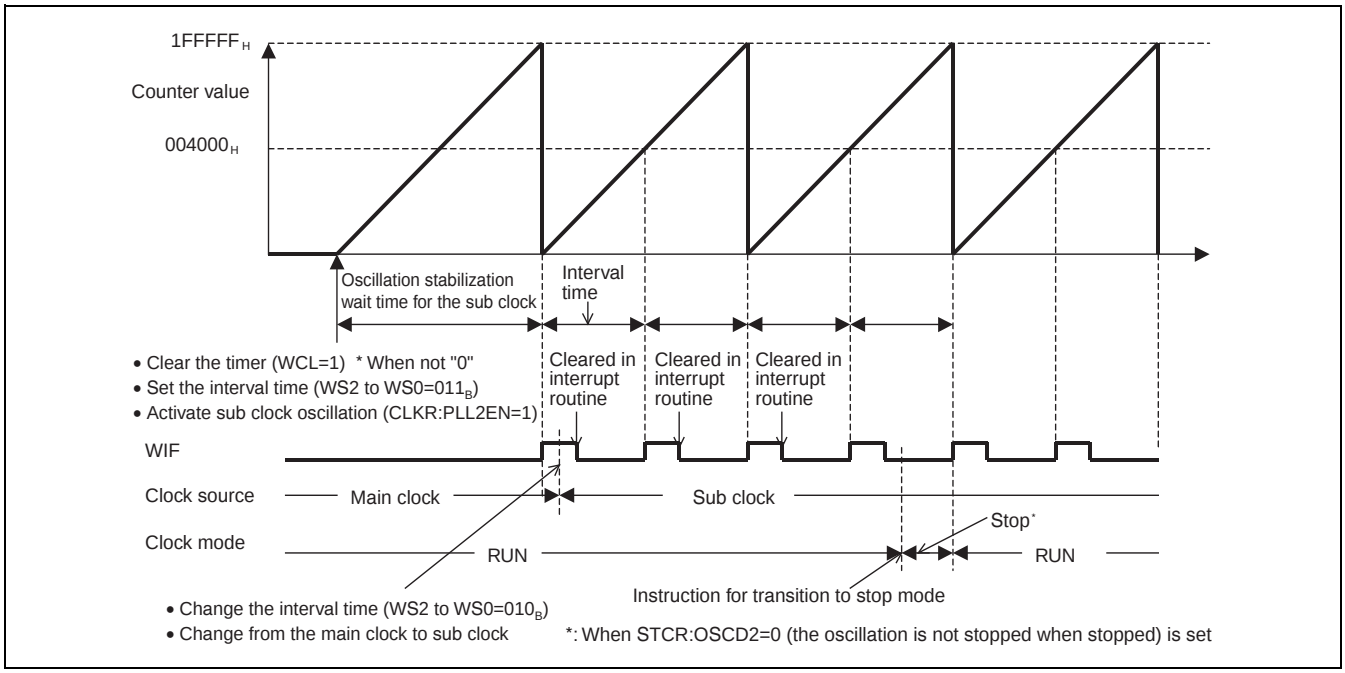
The time-base counter is used to secure the oscillation stabilization wait time after INIT or stop mode. However, to secure the oscillation stabilization wait time for the sub clock when the main clock is selected as the clock source, this watch timer is used, as it operates on the sub clock regardless of the clock source selection.

To perform the sub clock oscillation stabilization wait from the main clock operation, follow the procedure described below:

1) Set the interval time of the watch timer to 1 s (when F_{CL} = 32.768 kHz) and clear the counter to "0". (Write "011" to WS2 to WS0, and "0" to WCL)
   To perform processing after the completion of oscillation stabilization wait by an interrupt, also initialize the interrupt flag (write "0" to WIF and "1" to WIE).
2) Start the oscillation of the sub clock (write "1" to PLL2EN (bit11) in CLKR).
3) Wait until the WIF flag is set to "1" by program.
4) Make sure that the WIF flag has been set to "1" and then perform the processing after the completion of the oscillation stabilization wait. If interrupts are enabled, an interrupt occurs when WIF is set to "1". In this case, perform the processing after the completion of the oscillation stabilization wait in the interrupt routine.
   Also, when switching from the main clock to sub clock, make sure that WIF has been set to "1" beforehand, as described in 4). (If the clock is switched to the sub clock without waiting until the oscillation stabilizes, an unstable clock signal is supplied through the device, and the succeeding operation is not guaranteed.)

■ Operations of Watch Timer

The following shows the state of the counter when activating the watch timer, when making a transition to the sub clock, and when making a transition to the stop mode in the sub clock operation.



■ Notes on Using Watch Timer

- The oscillation stabilization wait time should be used for reference only, as the oscillation cycle is unstable immediately after the oscillation starts.
- While the oscillation of the sub clock is stopped, the watch timer is also stopped. Consequently, no watch interrupt occurs. Therefore, to perform any processing using a watch interrupt, do not stop the oscillation of sub clock.
- If the WIF flag set request and the clearing to "0" from the CPU occur simultaneously, the flag set request will have higher priority; therefore, the clearing to "0" will be cancelled.

CHAPTER 10

INTERRUPT CONTROLLER

This chapter gives an overview of the interrupt controller and explains its register configuration/ functions and its operations.

10.1 Overview of Interrupt Controller

The interrupt controller receives and arbitrates interrupts.

■ Hardware Configuration of the Interrupt Controller

This module consists of the following components:

- ICR register
- Interrupt priority judgment circuit
- Interrupt level and interrupt number (vector) generation unit
- Hold request cancel request generation unit

■ Major Functions of the Interrupt Controller

This module has the following major functions:

- Priority judgment (by interrupt level and number)
- Transmission of the interrupt level of the interrupt source selected by priority judgment (to the CPU)
- Transmission of the interrupt number of the interrupt source selected by priority judgment (to the CPU)
- Generation of a request to the bus master to cancel a hold request

■ Register List of Interrupt Controller

Figure 10.1-1 lists the registers of the interrupt controller.

Figure 10.1-1 List of Interrupt Controller Registers

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000440_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR00
000441_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01
000442_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02
000443_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03
000444_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04
000445_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05
000446_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06
000447_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07
000448_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR08
000449_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR09
$00044A_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR10
$00044B_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR11
$00044C_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR12
$00044D_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR13
$00044E_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR14
$00044F_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR15
000450_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR16
000451_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR17
000452_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR18
000453_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR19
000454_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR20
000455_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR21
000456_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR22
000457_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR23
000458_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR24
000459_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR25
$00045A_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR26
$00045B_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR27
$00045C_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR28
$00045D_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR29
$00045E_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR30
$00045F_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR31
000460_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR32
000461_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR33
000462_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR34
000463_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR35
000464_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR36
				R	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

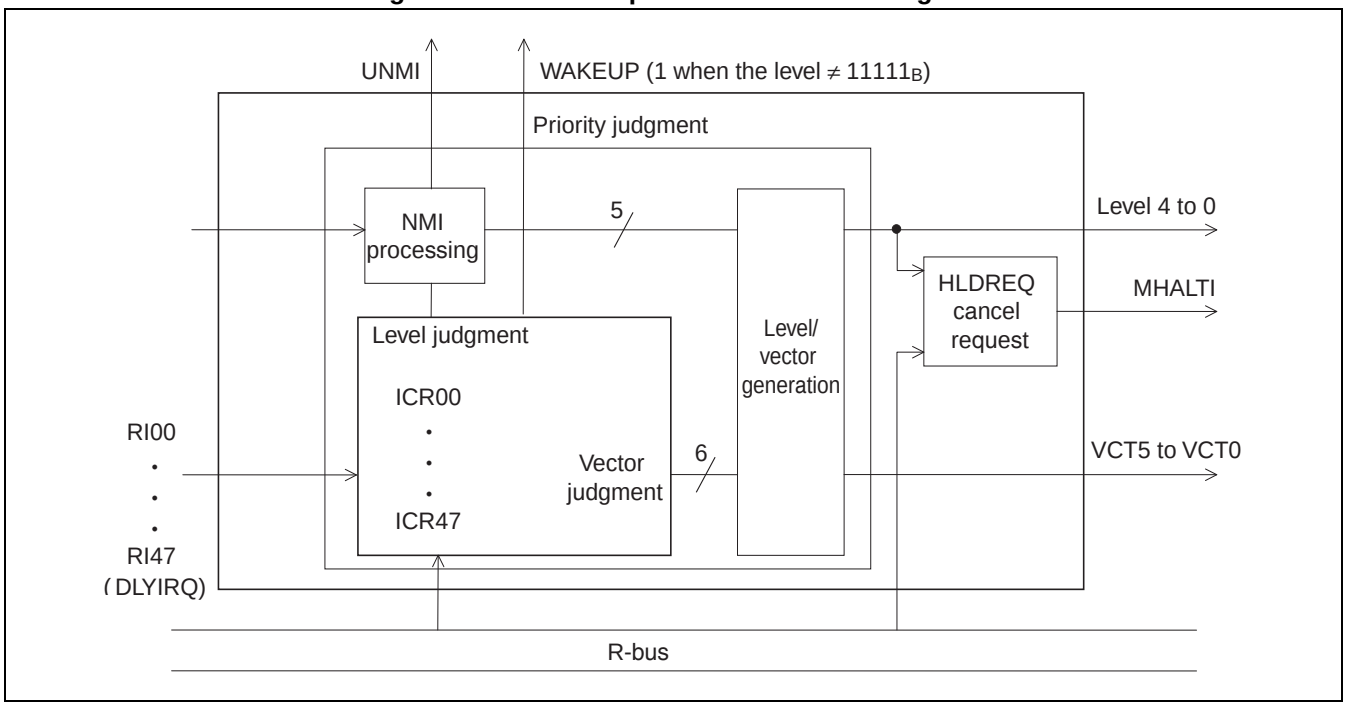
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000465_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR37
000466_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38
000467_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR39
000468_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR40
000469_H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR41
$00046A_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR42
$00046B_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR43
$00046C_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR44
$00046D_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR45
$00046E_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR46
$00046F_H$	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR47
				R	R/W	R/W	R/W	R/W	

Address:									
000045_H	-	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	HRCL
				R	R/W	R/W	R/W	R/W	

■ Interrupt Controller Block Diagram

Figure 10.1-2 is a block diagram of the interrupt controller.

Figure 10.1-2 Interrupt Controller Block Diagram



10.2 Registers of Interrupt Controller

This section describes the register configuration and functions of the interrupt controller.

■ Details of Interrupt Controller Registers

The interrupt controller has the following two types of registers:

- ICR (Interrupt Control Register)
- HRCL (Hold Request Cancel Request Register)

10.2.1 ICR (Interrupt Control Register)

An interrupt control register (ICR) is provided for each interrupt input to set the interrupt level of the corresponding interrupt request.

■ Interrupt Control Register (ICR)

The interrupt control register (ICR) consists of the following bits:

Figure 10.2-1 Bit Configuration of Interrupt Control Register (ICR)

ICR

Address	bit	7	6	5	4	3	2	1	0	Initial value
000440_H to $00046F_H$		-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	$---11111_B$
					R	R/W	R/W	R/W	R/W	

R/W: Readable/writable
R: Read only
-: Unused

[bit4 to bit0] ICR4 to ICR0

These bits are interrupt level setting bits to specify the interrupt level of the corresponding interrupt request.

The CPU masks the interrupt request if the interrupt level set in this register is greater than or equal to the level mask value set in the ILM register of the CPU.

The bits are initialized to "11111_B" at a reset.

Table 10.2-1 lists the available settings of the interrupt level setting bits and their respective interrupt levels.

Table 10.2-1 Available Settings of Interrupt Level Setting Bits and Corresponding Interrupt Levels

ICR4*	ICR3	ICR2	ICR1	ICR0	Interrupt level	
0	0	0	0	0	0	System-reserved
0	1	1	1	0	14	
0	1	1	1	1	15	NMI
1	0	0	0	0	16	Highest level available
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	Interrupts disabled

*: ICR4 is fixed to be "1"; "0" cannot be written to it.

10.2.2 HRCL (Hold Request Cancel Request Register)

The hold request cancel request register (HRCL) is a level setting register for generating a request to cancel a hold request.

■ Hold Request Cancel Request Register (HRCL)

The hold request cancel request register (HRCL) consists of the following bits:

Figure 10.2-2 Bit Configuration of Hold Request Cancel Request Register (HRCL)

HRCL

Address	bit	7	6	5	4	3	2	1	0	Initial value
00000045_H		-	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	$0--11111_B$
		-	-	-	R	R/W	R/W	R/W	R/W	

R/W: Readable/writable
R: Read only
-: Unused

[bit4 to bit0] LVL4 to LVL0

These bits set the interrupt level for generating a request to the bus master to cancel a hold request.

If an interrupt request with a higher priority level than the interrupt level set in this register occurs, a request to cancel the hold request is issued to the bus master.

The LVL4 bit is fixed to be "1"; "0" cannot be written to it.

10.3 Operations of Interrupt Controller

This section describes the operations of the interrupt controller.

■ Determining the Priority

This module selects the highest-priority interrupt among any interrupt sources that occur simultaneously and outputs its interrupt level and interrupt number to the CPU.

The criteria for determining the priority of interrupt sources are as follows.

- Interrupt source whose interrupt level is not 31 (31 indicates "interrupt disabled")
- Interrupt source with the lowest interrupt level value
- Interrupt source with the smallest interrupt number while satisfying the above

If no interrupt source is selected by the above criteria, 31 (11111_B) is output as the interrupt level. The interrupt number in this case is indeterminate.

For information on relationship between the interrupt sources, interrupt numbers, and interrupt levels, see "APPENDIX B Vector Table".

■ Hold Request Cancel Request

If you want to process high- priority interrupts during a CPU hold (during DMA transfer), the module that generated the hold request needs to cancel the request. Use the HRCL register to set the reference interrupt level at which a request to cancel is to be generated.

● Generation criteria

If an interrupt source with a higher priority level than the level set in the HRCL register occurs, a request to cancel the hold request is generated.

If interrupt level in HRCL register > level of interrupt after priority judgment, then generate cancel request.

If interrupt level in HRCL register ≤ level of interrupt after priority judgment, then do not generate cancel request.

The cancel request remains active until the interrupt source that generated the cancel request is cleared and therefore no DMA transfer occurs during this time. Always clear the associated interrupt source.

● Possible levels

The values able to be set in the HRCL register are "10000_B" to "11111_B", the same as in the ICR.

If "11111_B" is set, a cancel request is generated for all interrupt levels.

Table 10.3-1 shows the interrupt level settings for generating a request to cancel a hold request.

Table 10.3-1 Interrupt Level Settings That Generate a Hold Request Cancel Request

HRCL register	Interrupt levels that generate a cancel request
16	None
17	Interrupt level 16
18	Interrupt levels 16 and 17
≀	≀
31	Interrupt levels 16 to 30 [Initial value]

Once a reset occurs, DMA transfer is inhibited for all interrupt levels. As this means that no DMA transfer will be performed when an interrupt occurs, set the required value in the HRCL register.

■ Returning from Standby (Stop or Sleep) Mode

The function for using an interrupt request to return from stop mode is performed by this module. If even one interrupt request from a peripheral (with interrupt level other than "11111_B") occurs, a request to return from stop mode is issued to the clock control unit.

As the priority judgment unit restarts operation once the clock supply starts after recovery from stop mode, the CPU is able to execute instructions until the priority judgment unit produces a result.

The same operation occurs when returning from sleep mode. Access to the registers in this module remains possible even in sleep mode.

Note:

Set the interrupt level for interrupt sources that you do not want to cause the device to return from stop or sleep mode to "11111_B" in the corresponding peripheral control register.

■ Example of Using the Function to Generate a Request to Cancel a Hold Request (HRCR)

If you want the CPU to perform high-priority processing during DMA transfer, you need to cancel the hold state by requesting the DMA to cancel its hold request. This example uses an interrupt to cause the DMA to cancel its hold request and to give priority to CPU operation.

● Control registers

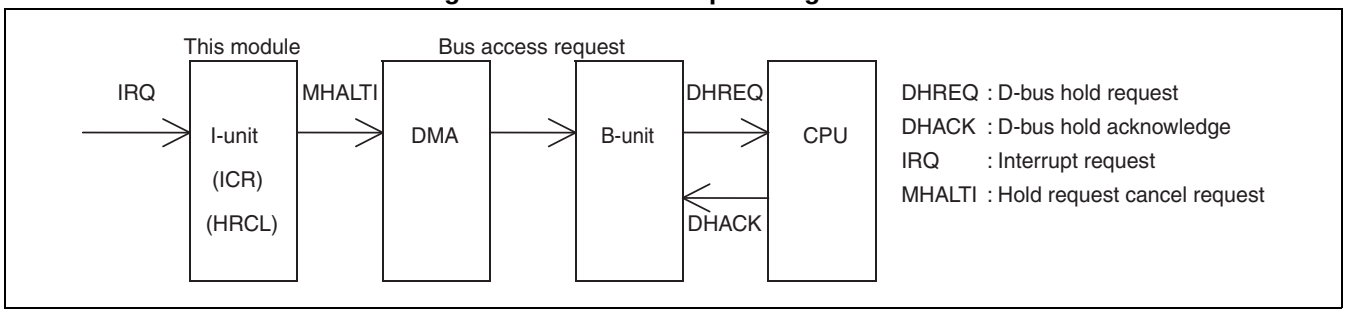
(1) HRCL (Hold request cancel level setting register): this module:
If an interrupt with a higher-priority level than the interrupt level set in this register occurs, a request to cancel the hold request is passed to the DMA. Set the level to use as the criterion.

(2) ICR: this module:
Set an interrupt level with a higher priority than the level set in the HRCL register in the ICRs of the interrupt sources you want to use.

● Hardware configuration

Figure 10.3-1 shows the flow of each signal for a hold request.

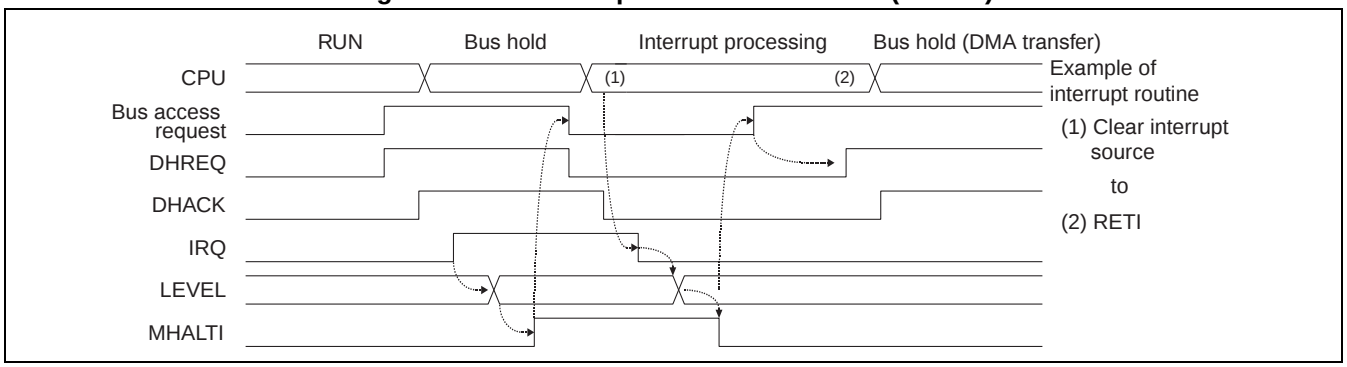
Figure 10.3-1 Hold Request Signal Flow



● Sequence

Figure 10.3-2 shows the interrupt level for a higher priority level than the level set in the HRCL register.

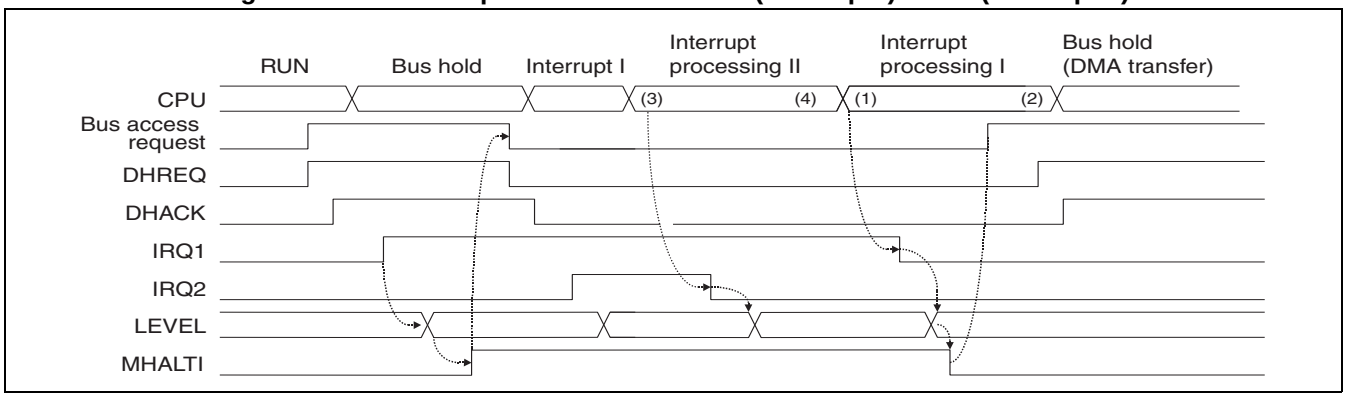
Figure 10.3-2 Interrupt Level: HRCL < ICR (LEVEL)



When an interrupt request occurs and the interrupt level changes, the MHALTI signal to the DMA goes active if the new level has a higher priority than the level set in the HRCL register. This causes the DMA to cancel access requests and the CPU to return from the hold state and start processing the interrupt.

Figure 10.3-3 shows the interrupt level for multiple interrupts.

Figure 10.3-3 Interrupt Level: HRCL < ICR (Interrupt I) < ICR (Interrupt II)



[Example of interrupt routine]

(1), (3) Clear interrupt source

to

(2), (4) RETI

The above example shows the case when a higher priority interrupt occurs during execution of interrupt routine I.

DHREQ remains low while the interrupt with an interrupt level higher than the interrupt level set in the HRCL register is present.

Note:

Pay due attention to the relationship between the interrupt levels set in the HRCL register and ICRs.

CHAPTER 11

EXTERNAL INTERRUPT CONTROL UNIT

This chapters gives an overview of the external interrupt control unit and describes its register configuration/ functions and its operations.

11.1 Overview of External Interrupt Control Unit

The external interrupt control unit is a block to control an external interrupt request input to the INT pin.
The external interrupt to be detected can be selected from among the following four:

- **"H" level**
- **"L" level**
- **Rising edge**
- **Falling edge**

These levels can be used for the STOP restore.

■ List of Registers of External Interrupt Control Unit

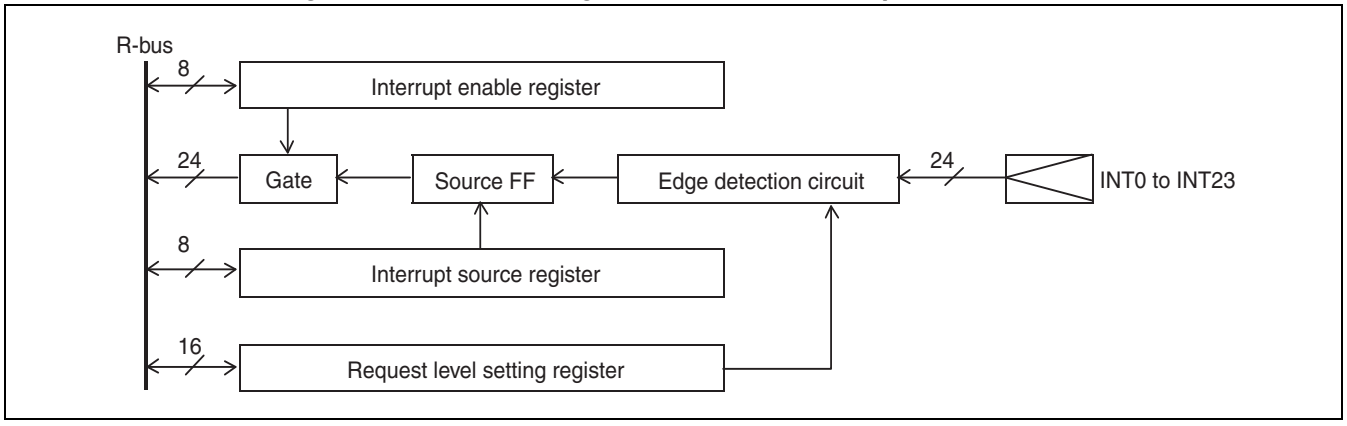
The registers of the external interrupt control unit are listed below.

Address	bit31 24	23 16	15 8	7 1
000040_H	EIRR0	ENIR0	ELVR0	
to				
$0000C0_H$	EIRR1	ENIR1	ELVR1	
$0000C4_H$	EIRR2	ENIR2	ELVR2	

■ Block Diagram of External Interrupt Control Unit

Figure 11.1-1 is a block diagram of the external interrupt control unit.

Figure 11.1-1 Block Diagram of External Interrupt Control Unit



11.2 Registers of External Interrupt Control Unit

This section describes the register configuration and functions of the external interrupt control unit.

■ Details of Registers of External Interrupt Control Unit

The external interrupt control unit has the following three types of registers:

- Interrupt Enable Register (ENIR)
- External Interrupt Source Register (EIRR)
- External Interrupt Request Level Setting Register (ELVR)

11.2.1 Interrupt Enable Register (ENIR)

The interrupt enable register (ENIR) controls the masking of the external interrupt request output.

■ Interrupt Enable Register (ENIR)

The interrupt enable register consists of the following bits:

Figure 11.2-1 Bit Configuration of Interrupt Enable Register (ENIR)

ENIR0

Address	bit	23	22	21	20	19	18	17	16	Initial value
000041_H		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ENIR1

Address	bit	15	14	13	12	11	10	9	8	Initial value
$0000C1_H$		EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ENIR2

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000C5_H$		EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

When "1" is written to a bit in this register, the interrupt request output corresponding to the bit is enabled (for example, EN0 controls the enabling of INT0), and the interrupt request is output to the interrupt controller. The pin corresponding to the bit to which "0" is written holds the interrupt source but does not generate a request to the interrupt controller.

11.2.2 External Interrupt Source Register (EIRR)

The external interrupt source register (EIRR) is a register to indicate that a corresponding external interrupt request exists when read, and to clear a content of the flip-flop showing this request when written.

■ External Interrupt Source Register (EIRR)

The external interrupt source register consists of the following bits:

Figure 11.2-2 Bit Configuration of External Interrupt Source Register (EIRR)

EIRR0

Address	bit	23	22	21	20	19	18	17	16	Initial value
000040_H		ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EIRR1

Address	bit	15	14	13	12	11	10	9	8	Initial value
$0000C0_H$		ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EIRR2

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000C4_H$		ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

The operation performed when this EIRR register is read depends on the read value as follows.

When a bit contains "1", it indicates that there is an external interrupt request at the pin corresponding to that bit. Writing "0" to a bit in this register clears the request flip-flop of that bit.

Writing "1" is ignored. When this bit is read to a read-modify-write (RMW) instruction, "1" is always read.

Depending on the pin state, the bit value of the external interrupt source register can be "1" even if "0" is written to the corresponding bit of the external interrupt enable register.

11.2.3 External Interrupt Request Level Setting Register (ELVR)

The external interrupt request level setting register (ELVR) is a register to select request detections.

■ External Interrupt Request Level Setting Register (ELVR)

The external interrupt request level setting register (ELVR) consists of the following bits.

Figure 11.2-3 Bit Configuration of External Interrupt Request Level Setting Register (ELVR)

ELVR0

Address	bit	15	14	13	12	11	10	9	8	Initial value
000042_H		LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit	7	6	5	4	3	2	1	0	Initial value
000043_H		LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ELVR1

Address	bit	15	14	13	12	11	10	9	8	Initial value
$0000C2_H$		LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000C3_H$		LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ELVR2

Address	bit	15	14	13	12	11	10	9	8	Initial value
$0000C6_H$		LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit	7	6	5	4	3	2	1	0	Initial value
$0000C7_H$		LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

In the ELVR register, two bits are assigned to each interrupt channel, which results in the settings shown in the table below.

When each bit in the EIRR register is cleared while the level is in the request input level, the corresponding bit is set again as long as the input is at active level.

Table 11.2-1 shows assignment of ELVR.

Table 11.2-1 Assignment of ELVR

LBx, LAx	Operation
00	"L" level indicates the presence of a request. [Initial value]
01	"H" level indicates the presence of a request.
10	A rising edge indicates the presence of a request. *
11	A falling edge indicates the presence of a request. *

*: The setting to INT16-to-INT23 is disabled.

Note: Any request level can be set for restoring from STOP. (INT0 to INT15)

Notes:

- If external interrupt request level is changed, internal interrupt request may be occurred. So clear the external interrupt register after changing the external interrupt request level. When you want to clear the external interrupt request level register once.
- Edge detection is not available for the INT17 and INT19 channels. Always set these channels to use level detection.
  This restriction does not apply to the MB91F313A.
- All request levels are able to be set after recovery from STOP (except for INT17 and INT19).
  This restriction does not apply to the MB91F313A.

11.3 Operations of External Interrupt Control Unit

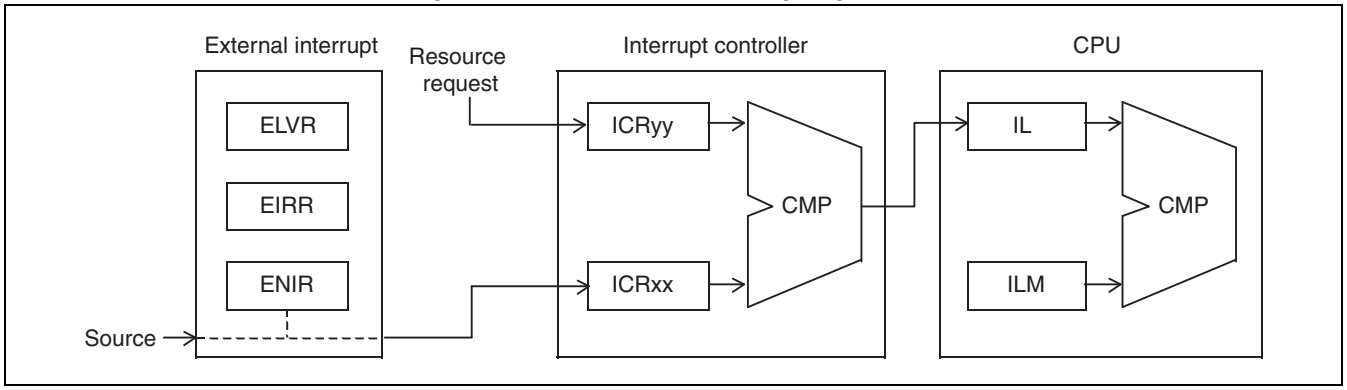
This section describes the operations of the external interrupt control unit.

■ Operations of an External Interrupt

If, after a request level and an enable register are set, a request defined in the ELVR register is input to the corresponding pin, this module generates an interrupt request signal to the interrupt controller. The interrupt controller identifies the priorities of interrupts simultaneously generated within the interrupt controller and, if it determines that the interrupt request from this resource has the highest priority, the corresponding interrupt generates.

Figure 11.3-1 shows the external interrupt operation.

Figure 11.3-1 External Interrupt Operation



■ Operating Procedure for an External Interrupt

Set up the registers located inside the external interrupt control unit as follows:

1. Set the general-purpose I/O port served dual use as external interrupt input pin as the input port.
2. Disable the target bit in the enable interrupts register (ENIR).
3. Set the target bit in the external interrupt request level setting register (ELVR).
4. Read the external interrupt request level setting register (ELVR).
5. Clear the target bit in the enable interrupts register (ENIR).
6. Enable the target bit in the enable interrupts register (ENIR).

However, simultaneous writing of 16-bit data is allowed for steps 5. and 6.

Before setting a register in this module, you must disable the enable register. In addition, before enabling the enable register, you must clear the interrupt source register. This procedure is required to prevent an interrupt source from occurring by mistake while a register is being set or an interrupt is enabled.

■ External Interrupt Request Level

If the request level is an edge request, a pulse width of at least 3 machine cycles (peripheral clock machine cycles) is required to detect an edge.

When the request input level is a level setting, the required pulse width is a minimum of 3 machine cycles. While the interrupt input pin is holding its active level, the interrupt request to the interrupt controller keeps on being generated even with the external interrupt source register cleared.

If the request input level is a level setting, a request input is entered from outside and is then cancelled, the request to the interrupt controller remains active because a source holding circuit exists internally.

The external interrupt source register must be cleared to cancel a request to the interrupt controller.

Figure 11.3-2 illustrates the clearing of the source holding circuit when a level is set.

Figure 11.3-2 Clearing the Source Holding Circuit When a Level is Set

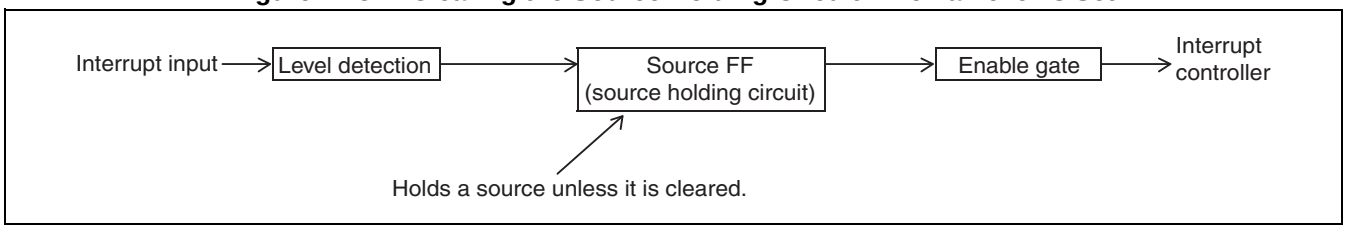
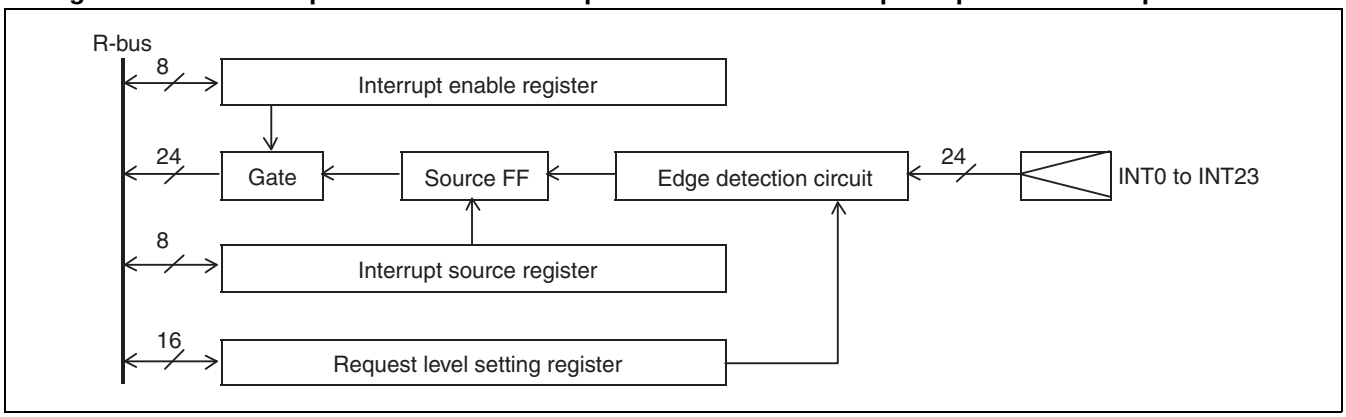


Figure 11.3-3 shows an interrupt source and an interrupt request to the interrupt controller when interrupts are enabled.

Figure 11.3-3 Interrupt Source with Interrupts Enables and Interrupt Request to Interrupt Controller



CHAPTER 12

DELAY INTERRUPT MODULE

This chapter explains the overview of the delay interrupt module, configuration/functions of the registers, and operations of the module.

12.1 Overview of Delay Interrupt Module

**The delay interrupt module generates an interrupt for switching tasks.
Using this module, software can generate or release an interrupt request for the CPU.**

■ Register List of Delay Interrupt Module

Register list of the delay interrupt module is as follows:

DICR

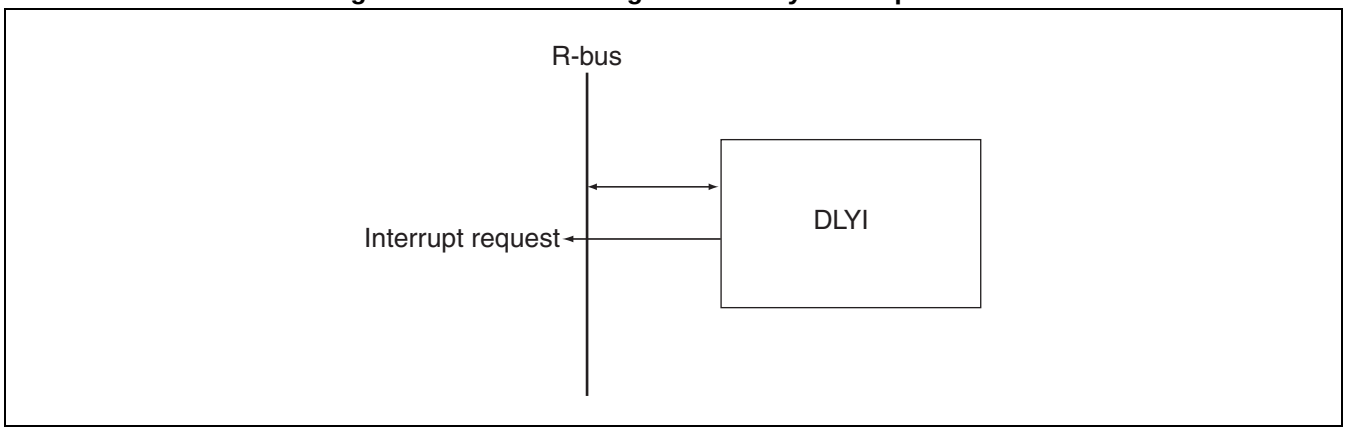
Address	bit	7	6	5	4	3	2	1	0
000044_H		-	-	-	-	-	-	-	DLYI
									R/W

R/W: Readable/writable
-: Undefined

■ Block Diagram of Delay Interrupt Module

Figure 12.1-1 shows a block diagram of the delay interrupt module.

Figure 12.1-1 Block Diagram of Delay Interrupt Module



12.2 Registers of Delay Interrupt Module

This section explains the register configurations/functions of the delay interrupt module.

■ DICR (Delay Interrupt Module Register)

DICR controls the delay interrupt.

The bit configuration of the delay interrupt module register (DICR) is as follows:

DICR

Address	bit	7	6	5	4	3	2	1	0	Initial value
000044_H		-	-	-	-	-	-	-	DLYI	$-------0_B$
									R/W	

R/W: Readable/writable
-: Undefined

[bit0] DLYI

DLYI	Description
0	No release and request of delay interrupt source [Initial value]
1	Generated delay interrupt source

This bit controls generating and releasing of the corresponding interrupt sources.

12.3 Operations of Delay Interrupt Module

The delay interrupt is an interrupt generated for switching tasks.Use this function to allow a software program to generate an interrupt request for the CPU or to release an interrupt request.

■ Interrupt Number

A delay interrupt is assigned to the interrupt source corresponding to the largest interrupt number.

In this model, a delay interrupt is assigned to interrupt number 63 ($3F_H$).

■ DLYI Bit of DICR

Writing "1" to this bit generates a delay interrupt source.Writing "0" clears a delay interrupt source.

This bit is the same as the interrupt source flag for a normal interrupt. Therefore, clear this bit and switch tasks in the interrupt routine.

CHAPTER 13

BIT SEARCH MODULE

This chapter explains the overview of the bit search module, configurations/functions of the registers, and operations of the module.

13.1 Overview of Bit Search Module

The bit search module searches for "0", "1", or any points of change for data written to the input register and then returns the detected bit locations.

■ Register List of Bit Search Module

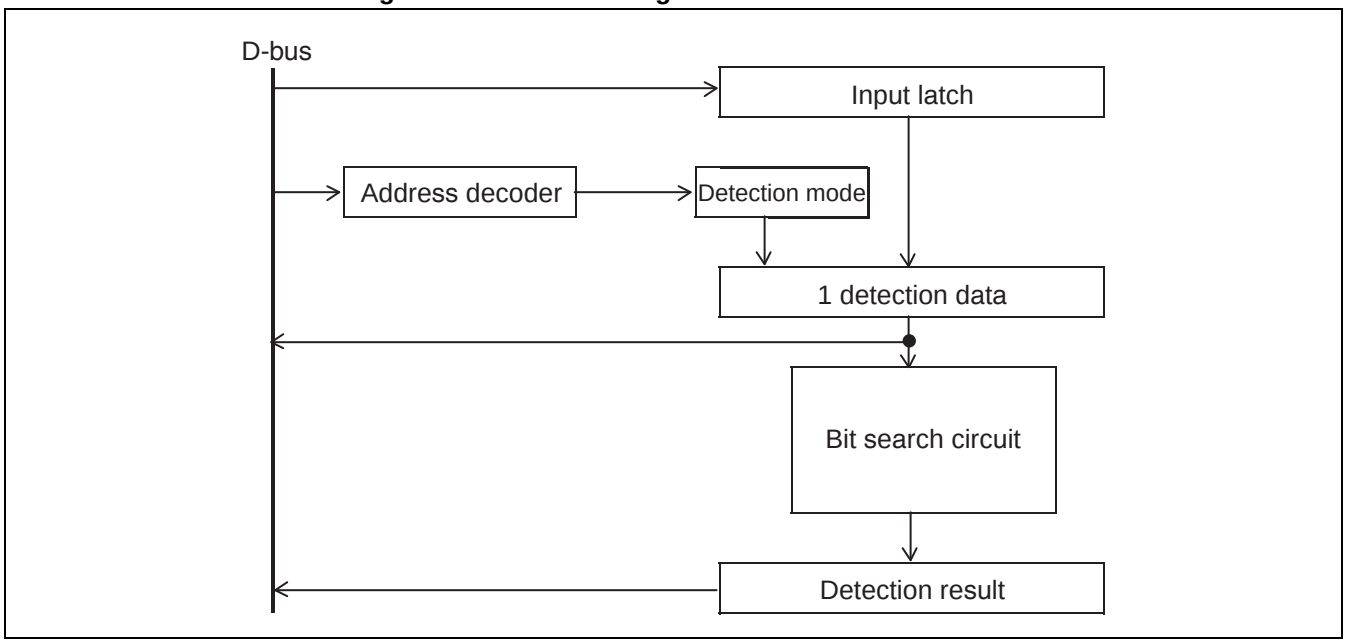
Register list of the bit search module is as follows:

	bit31	bit0	
Address: $0003F0_H$	BSD0		0 Detection data register
Address: $0003F4_H$	BSD1		1 Detection data register
Address: $0003F8_H$	BSDC		Change point detection data register
Address: $0003FC_H$	BSRR		Detection result register

■ Block Diagram of Bit Search Module

Figure 13.1-1 shows a block diagram of the bit search module.

Figure 13.1-1 Block Diagram of Bit Search Module



13.2 Registers of Bit Search Module

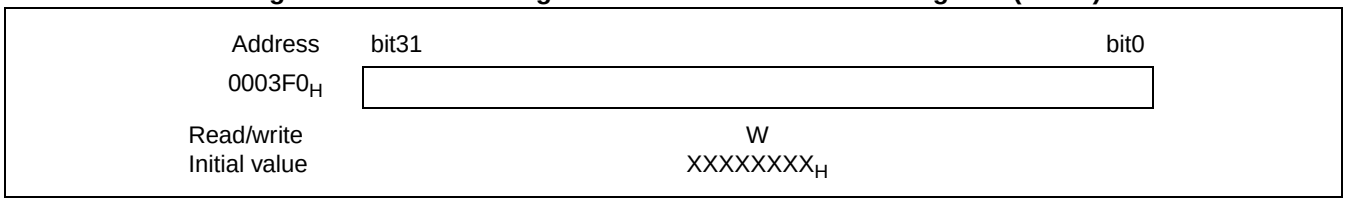
This section explains the register configurations/functions of the bit search module.

■ 0 Detection Data Register (BSD0)

0 detection is performed for written value.

Shown below is the configuration of the 0 detection data register (BSD0):

Figure 13.2-1 Bit Configuration of 0 Detection Data Register (BSD0)



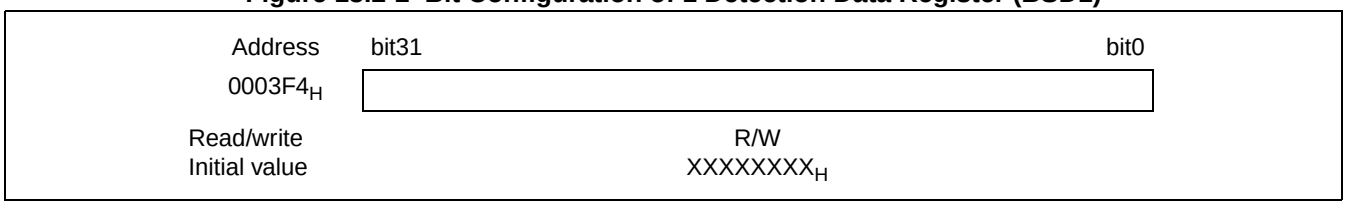
The initial value after a reset is undefined. Read value is undefined.

Use a 32-bit length data transfer instruction for data transfer (Do not use 8-bit or 16-bit length data transfer instructions).

■ 1 Detection Data Register (BSD1)

Shown below is the configuration of the 1 detection data register (BSD1):

Figure 13.2-2 Bit Configuration of 1 Detection Data Register (BSD1)



Use a 32-bit length data transfer instruction for data transfer (Do not use 8-bit or 16-bit length data transfer instructions).

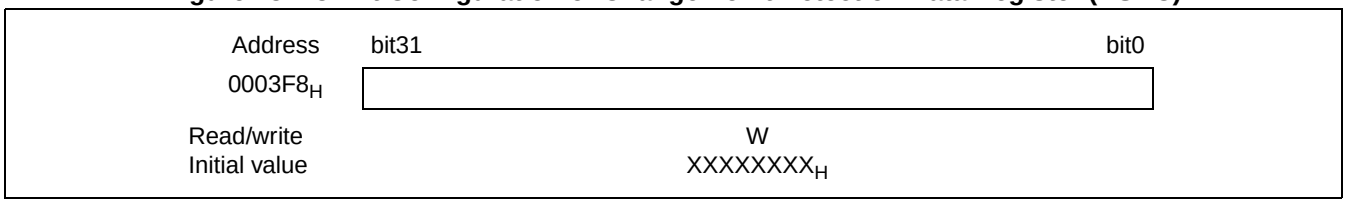
- Writing:
  "1" detection is performed for the written value.
- Reading:
  Saved data of the internal state in the bit search module is read. This register is used to save and restore to the original state when the bit search module is used by, for example, an interrupt handler.
  Even though data is written to the 0 detection, change point detection, or data register, the data can be saved and restored only by using the 1 detection data register.
  The initial value after a reset is undefined.

■ Change Point Detection Data Register (BSDC)

Point of change is detected in the written value.

Shown below is the configuration of the change point detection data register (BSDC):

Figure 13.2-3 Bit Configuration of Change Point Detection Data Register (BSDC)



The initial value after a reset is undefined.

Read value is undefined.

Use a 32-bit length data transfer instruction for data transfer (Do not use 8-bit or 16-bit length data transfer instructions).

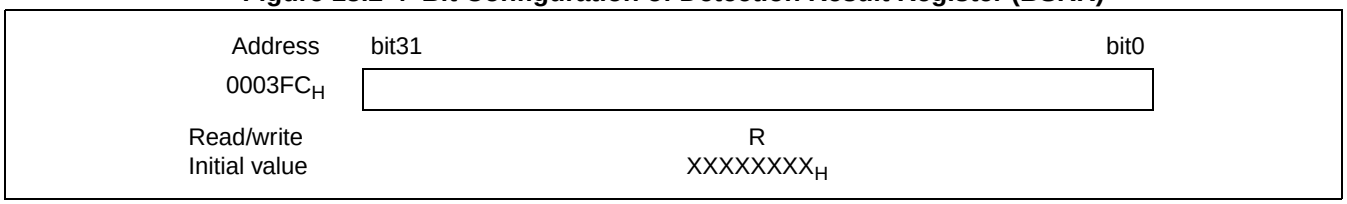
■ Detection Result Register (BSRR)

The result of 0 detection, 1 detection, or change point detection is read.

Which detection result is to be read is determined by the data register that has been written to last.

Register configuration of the detection result register (BSRR) is as follows:

Figure 13.2-4 Bit Configuration of Detection Result Register (BSRR)



13.3 Operations of Bit Search Module

This section explains the operations of the bit search module.

■ 0 Detection

The bit search module scans data written to the 0 detection data register from MSB to LSB and returns the location where the first "0" is detected.

The detection result can be obtained by reading the detection result register. The relationship between the detected location and the return value is described in Table 13.3-1.

If "0" is not found (that is, the value is $FFFFFFFF_H$), 32 is returned as the search result.

[Execution example]

Written data		Read value (decimal)
$11111111111111111111000000000000_B$ ($FFFFF000_H$)	→	20
$11111000010010011110000010101010_B$ ($F849E0AA_H$)	→	5
$10000000000000101010101010101010_B$ ($8002AAAA_H$)	→	1
$11111111111111111111111111111111_B$ ($FFFFFFFF_H$)	→	32

■ 1 Detection

The bit search module scans data written to the 1 detection data register from MSB to LSB and returns the location where the first "1" is detected.

The detection result can be obtained by reading the detection result register. The relationship between the detected location and the return value is described in Table 13.3-1.

If "1" is not found (that is, the value is 00000000_H), 32 is returned as the search result.

[Execution example]

Written data		Read value (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

The bit search module scans data written to the change point detection data register from bit30 to LSB for comparison with the MSB value.

The first location where a value that is different from that of the MSB is detected is returned. The detection result can be obtained by reading the detection result register.

The relationship between the detected location and the return value is described in Table 13.3-1.

If a change point is not detected, 32 is returned. In change point detection, "0" is never returned as a result.

[Execution example]

Written data		Read value (decimal)
$0010000000000000000000000000000_B$ (20000000_H)	→	2
$00000001001000110100010101100111_B$ (01234567_H)	→	7
$00000000000000111111111111111111_B$ ($0003FFFF_H$)	→	14
$00000000000000000000000000000001_B$ (00000001_H)	→	31
$00000000000000000000000000000000_B$ (00000000_H)	→	32
$11111111111111111111000000000000_B$ ($FFFFF000_H$)	→	20
$11111000010010011110000010101010_B$ ($F849E0AA_H$)	→	5
$10000000000000101010101010101010_B$ ($8002AAAA_H$)	→	1
$11111111111111111111111111111111_B$ ($FFFFFFFF_H$)	→	32

Table 13.3-1 shows the bit locations and return values (decimal).

Table 13.3-1 Bit Locations and Return Values (Decimal)

Detected Bit Location	Return Value	Detected Bit Location	Return Value	Detected Bit Location	Return Value	Detected Bit Location	Return Value
31	0	23	8	15	16	7	24
30	1	22	9	14	17	6	25
29	2	21	10	13	18	5	26
28	3	20	11	12	19	4	27
27	4	19	12	11	20	3	28
26	5	18	13	10	21	2	29
25	6	17	14	9	22	1	30
24	7	16	15	8	23	0	31
						Does not exist	32

■ Process of Save/Restore

If it is necessary to save and restore the internal state of the bit search module, such as when the bit search module is used in an interrupt handler, use the following procedure:

1) Read the 1 detection data register and save its content (save).
2) Use the bit search module.
3) Write the data saved in 1) to the 1 detection data register (restore).

With the above operation, the value obtained when the detection result register is read the next time corresponds to the value written to the bit search module before 1).

If the data register written to last is the 0 detection or change point detection register, the value is restored correctly with the above procedure.

CHAPTER 14

10-BIT A/D CONVERTER

This chapter gives an overview of the 10-bit A/D converter, register configuration and functions, and 10- bit A/D converter operation.

14.1 Overview of the 10-Bit A/D Converter

The 10-bit successive approximation A/D converter has two operation modes: conversion start by software and conversion start by external trigger.

■ Features of the 10-Bit A/D Converter

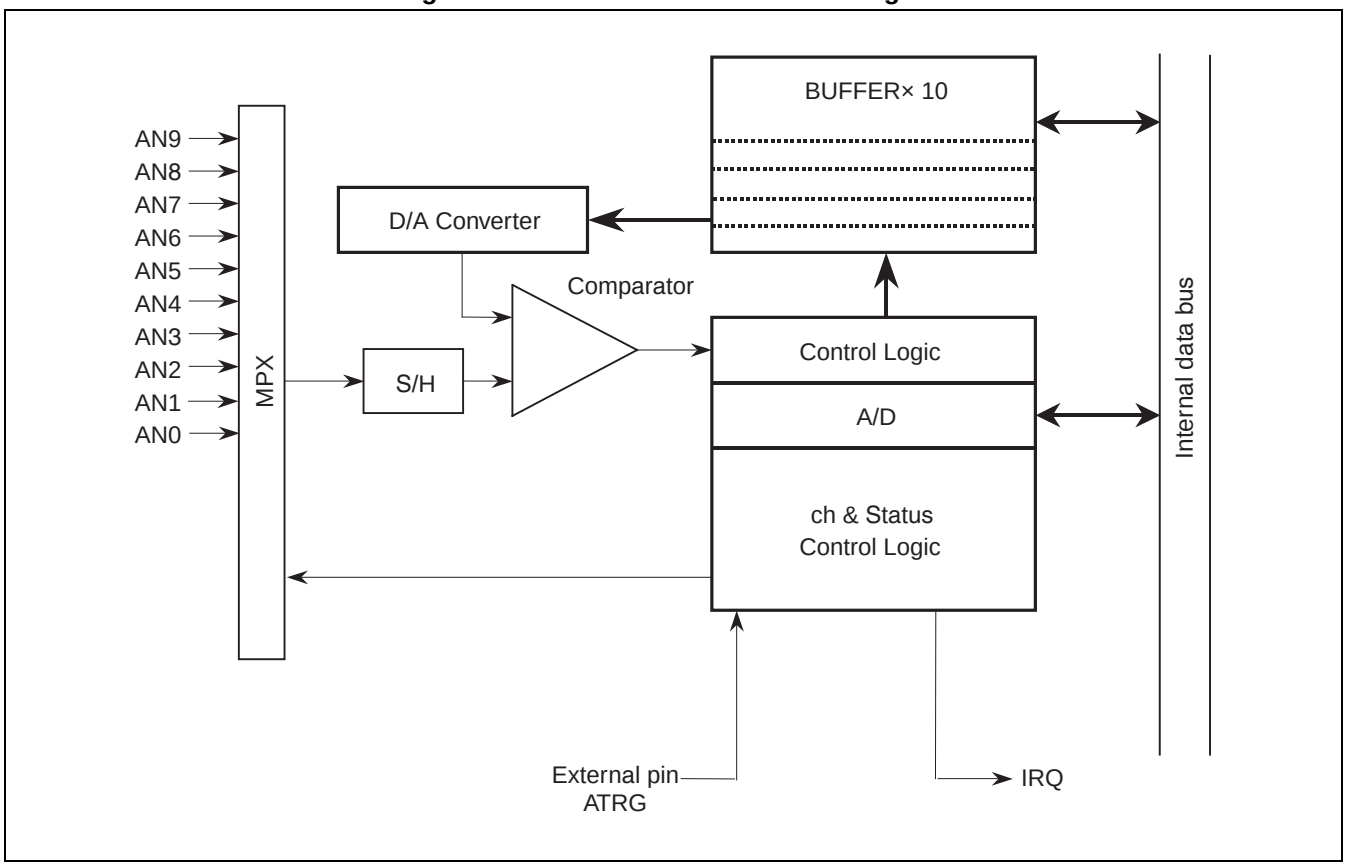
- Conversion time: 7.94 µs (sampling: 5.91 µs, conversion: 2.03 µs) when fch is @33MHz
- A/D conversion result register available for each channel
- Channel scan function

■ Registers

Address	bit15		bit0	
000020H	ADCTH	ADCTL		A/D Converter control register
000022H	ADCH			Software conversion analog input select register
000024H	ADAT0			A/D conversion result register ch.0
000026H	ADAT1			A/D conversion result register ch.1
000028H	ADAT2			A/D conversion result register ch.2
00002AH	ADAT3			A/D conversion result register ch.3
00002CH	ADAT4			A/D conversion result register ch.4
00002EH	ADAT5			A/D conversion result register ch.5
000030H	ADAT6			A/D conversion result register ch.6
000032H	ADAT7			A/D conversion result register ch.7
000034H	ADAT8			A/D conversion result register ch.8
000036H	ADAT9			A/D conversion result register ch.9
000038H	TEST			A/D converter test register (access prohibited)

■ Block Diagram

Figure 14.1-1 A/D Converter Block Diagram



■ A/D Converter Control Register (ADCTH, ADCTL)

Figure 14.1-2 Bit Configuration of A/D Converter Control Register (ADCTH, ADCTL)

ADCTH

Address	bit	15	14	13	12	11	10	9	8	Initial value
000020_H		-	-	-	-	-	-	TRG	STR	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ADCTL

Address	bit	7	6	5	4	3	2	1	0	Initial value
000021_H		ASS3	ASS2	ASS1	ASS0	BUSY	-	INT	INTE	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

[bit15 to bit10] Unused bits

The read value of these bits are always "000000_B".

[bit9] TRG

0	Start by external pin trigger is prohibited.
1	Start by external pin trigger

When this bit is set to "1", A/D conversion is started when a rising edge is detected at external pin (ATRG) input.

This bit is ignored if an edge is detected during A/D conversion.

[bit8] STR

This bit is the A/D conversion start bit.

0	No effect
1	Software start/restart (write during conversion)

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

[bit7 to bit4] ASS3 to ASS0

These bits enable reading of the selected analog channel.

This bit enables reading of effective data when [bit3] BUSY = 1.

0000_B to 1001_B	Selected channel
Others	Setting disabled

[bit3] BUSY

This bit is a flag that indicates A/D conversion is in progress.

0	A/D conversion is not in progress.
1	A/D conversion is in progress.

[bit2] Unused bit

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

[bit1] INT

This bit is the A/D conversion end flag.

0	No conversion, or conversion is in progress.
1	Conversion is completed.

[bit0] INTE

This bit is the A/D conversion interrupt enable bit.

0	Interrupt is disabled.
1	Interrupt is enabled.

When INT and INTE are both set to "1", an interrupt request is generated.

■ Software Conversion Analog Input Select Register

Figure 14.1-3 Bit Configuration of Software Conversion Analog Input Select Register

ADCH

Address	bit	15	14	13	12	11	10	9	8	Initial value
000022_H		-	-	-	-	-	-	i9	i8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address	bit	7	6	5	4	3	2	1	0	Initial value
000022_H		i7	i6	i5	i4	i3	i2	i1	i0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W:Readable/writable

[bit15 to bit10] Unused bits

The read value of these bits are always "0".

[bit9 to bit0] i9 to i0

These bits are the software conversion analog input select bits.

0	Input is not selected.
1	Input is selected.

If multiple inputs are selected, data is sequentially converted for all selected inputs.

■ A/D Conversion Result Register (Channels 0 to 9)

Figure 14.1-4 Bit Configuration of A/D Conversion Result Register (Channels 0 to 9)

ADAT0 to ADAT9

Address	bit	15	14	13	12	11	10	9	8	Initial value
000022_H		-	-	-	-	-	-	d9	d8	00000000_B
		R	R	R	R	R	R	R	R	
Address	bit	7	6	5	4	3	2	1	0	Initial value
000022_H		d7	d6	d5	d4	d3	d2	d1	d0	00000000_B
		R	R	R	R	R	R	R	R	

R: Read only

[bit15 to bit10] Reserved

The read value of these bits are always "000000_B".

[bit9 to bit0] d9 to d0

These bits store the A/D conversion result for the channels.

■ A/D Conversion Started by Software

To perform A/D conversion started by software, select the required channel from analog input pins AN0 to AN9. Write "1" to the corresponding bit of the ADCH register to enable A/D conversion.

(1) Single channel

If only one channel is selected as the analog input pin for conversion, writing "1" to the STR bit of the ADCTH register starts software-started conversion and sets the BUSY bit of the ADCH register to "1".

Writing "1" to the STR bit again during conversion initializes the converter and restarts conversion.

After A/D conversion ends, the BUSY bit of the ADCH register is reset to "0" and the INT bit of the ADCTL register is set to "1". These status bits can be read to determine whether conversion has ended. To generate an interrupt to complete conversion, set the INTE bit of the ADCTL register to "1" beforehand.

(2) Multiple channels (scan conversion)

If multiple channels are selected as the analog input pins for conversion, the converter performs A/D conversion for the first selected channel and then stores the conversion result in the register corresponding to the channel. The converter then repeats this process for the remaining selected channels.

Writing 1 to the corresponding bit of the ADCH register to select the channel for conversion and writing "1" to the STR bit of the ADCH register start conversion and set the BUSY bit of the ADCTL register to "1". The channels are converted sequentially from 0 to 9. If a channel is not selected in the ADCH register, the converter skips that channel and starts conversion for the next selected channel. Writing "1" to the STR bit again during conversion initializes the converter and restarts conversion for the selected channels in the order of 0 to 9.

When A/D conversion for all selected channels ends, the BUSY bit of the ADCTL register is reset to "0" and the INT bit of the ADCTL register is set to "1". To generate an interrupt to complete conversion, set the INTE bit of the ADCTL register to "1" beforehand.

The results of A/D conversion are stored in the registers of individual channels.

■ A/D Conversion Started by External Trigger

If external trigger start is enabled (ADCTH: TRG = 1), detection of a rising edge at external pin (ATRG) input starts A/D conversion. If the signal for A/D conversion by software is received when the external trigger is enabled, conversion is also started. If a rising edge is detected again at external pin (ATRG) input during A/D conversion, conversion is initialized and restarts the operation from the beginning.

Note:

When entering the stop mode in the low-power consumption mode, be sure to confirm that A/D conversion is not in progress.

CHAPTER 15
MULTI FUNCTION SERIAL INTERFACE

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

15.1 Overview of the Multi Function Serial Interface

This section describes the overview of multi function serial interface.

■ Interface Mode

The multi function serial interface can select the following interface modes depending on the operating mode setting:

- UART0 (asynchronous normal multi function serial interface)
- UART1 (asynchronous multiprocessor multi function serial interface)
- CSIO (clock synchronous multi function serial interface) (capable of supporting SPI)
- I^2C (I^2C bus interface)

■ Switching the Interface Modes

When communicating with each multi function serial interface, set the operating mode in the registers shown in Table 15.1-1 before starting the communication.

Figure 15.1-1 Bit Configuration of Serial Mode Register (SMR)

SMR

bit	7	6	5	4	3	2	1	0	Initial value
	MD2	MD1	MD0	-	WUCR	SBL	SCKE	SOE	$000\text{-}0000_B$
	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	

R/W: Readable/writable
-: Undefined

Table 15.1-1 Switching the Interface Modes

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (asynchronous normal multi function serial interface)
0	0	1	UART1 (asynchronous multiprocessor multi function serial interface)
0	1	0	CSIO (clock synchronous multi function serial interface) (capable of supporting SPI)
1	0	0	I^2C (I^2C bus interface)

Note: Any mode other than those above is prohibited to set.

Notes:

- If you switch the mode while transmission/reception operation using 1 multi function serial interface, the transmission/reception operation cannot be guaranteed.
- Set the operating mode first because the other registers will be initialized once the operating mode has been changed. However, when SCR and SMR are written at the same time by 16-bit writing, the written contents will be reflected on SCR.

■ Transmission and Reception FIFO (Ch.0, Ch.1 and Ch.2)

16-byte transmission FIFO and 16-byte reception FIFO are provided to ch.0, ch.1 and ch.2. Read the number of stages of FIFO described from here on as 16-byte. Since FIFO will not be provided other than ch.0, ch.1 and ch.2, ignore the description about FIFO and refer to the description when FIFO is not used.

15.2 Functions of UART (Asynchronous Multi Function Serial Interface)

UART (asynchronous multi function serial interface) is a general-purpose serial data communication interface for asynchronous communication with external devices. UART supports bidirectional communication function (normal mode), master/slave communication function (multiprocessor mode: supports both master/slave). In addition, it has the transmission/reception FIFO.

■ Functions of UART (Asynchronous Multi Function Serial Interface)

		Function
1	Data	• Full-duplex, double buffering (when FIFO is not used) • Transmission/reception FIFO (maximum size 16-byte each) (when FIFO is used)
2	Serial input	Performs oversampling 3 times and determines the reception value by majority decision of the sampling values
3	Transfer format	Asynchronous
4	Baud rate	• Dedicated baud rate generator (15-bit reload counter configuration) • External clock input can be adjusted with the reload counter
5	Data length	• 5 to 9 bits (normal mode) and 7 or 8 bits (multiprocessor mode)
6	Signal method	NRZ (Non Return to Zero), invert NRZ
7	Start bit detection	• Synchronizes with the start bit falling edge (for NRZ method) • Synchronizes with the start bit rising edge (for invert NRZ method)
8	Reception error detection	• Framing error • Overrun error • Parity error*
9	Interrupt request	• Reception interrupt (reception completion, framing error, overrun error, and parity error*) • Transmission interrupt (transmission data empty, transmission bus idle) • Transmission FIFO interrupt (when the transmission FIFO is empty) • Both transmission/reception have the extended intelligent I/O service (EI^2OS) and DMA function
10	Master/slave communication function (multiprocessor mode)	1: n communication (1 = master, n = slaves) is enabled (supports both master and slave systems)
11	FIFO options	• Transmission/reception FIFO are provided (maximum size: transmission FIFO 16-byte, reception FIFO 16-byte) • Transmission FIFO and reception FIFO can be selected • Transmission data can be retransmitted • Reception FIFO interrupt timing can be changed from the software • Independent FIFO reset support

*: Parity error is enabled in normal mode only.

15.3 Registers of UART (Asynchronous Multi Function Serial Interface)

This section shows the register list of UART (asynchronous multi function serial interface).

■ Register List of UART (Asynchronous Multi Function Serial Interface)

Table 15.3-1 Register List of UART (Asynchronous Multi Function Serial Interface)

	Address		bit15 bit8	bit7 bit0
UART	$000X0_H$	$000X1_H$	SCR (serial control register)	SMR (serial mode register)
	$000X2_H$	$000X3_H$	SSR (serial status register)	ESCR (extended communication control register)
	$000X4_H$	$000X5_H$	RDR1/TDR1 (transmission and reception data register 1)	RDR0/TDR0 (transmission and reception data register)
	$000X6_H$	$000X7_H$	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	$000X8_H$	$000X9_H$	-	-
FIFO	$000YA_H$	$000YB_H$	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)
	$000YC_H$	$000YD_H$	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)

(X = 06_H, 07_H, 08_H, 09_H, $0A_H$, $0B_H$, $1B_H$, $1C_H$, $1D_H$, $1E_H$, $1F_H$, Y = 06_H, 07_H, 08_H)

Table 15.3-2 Bit Arrangement of UART (Asynchronous Multi Function Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SBL	BDS	SCKE	SOE
SSR/ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	-	-	INV	PEN	P	L2	L1	L0
TDR (RDR)	-							D8(AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

■ Operating Modes

UART (asynchronous multi function serial interface) operates in 2 different modes. MD2, MD1, and MD0 in the serial mode register (SMR) are used to select the operating mode.

Table 15.3-3 Operating Mode of UART (Asynchronous Multi Function Serial Interface)

Operating mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multiprocessor mode)

15.3.1 Serial Control Register (SCR)

Serial control register (SCR) can enable/disable the transmission and reception, the transmission and reception interrupt, transmission bus idle interrupt, and perform UART reset.

■ Serial Control Register (SCR)

Figure 15.3-1 shows the bit configuration of the serial control register (SCR), and Table 15.3-4 shows the function of each bit.

Figure 15.3-1 Bit Configuration of Serial Control Register (SCR)

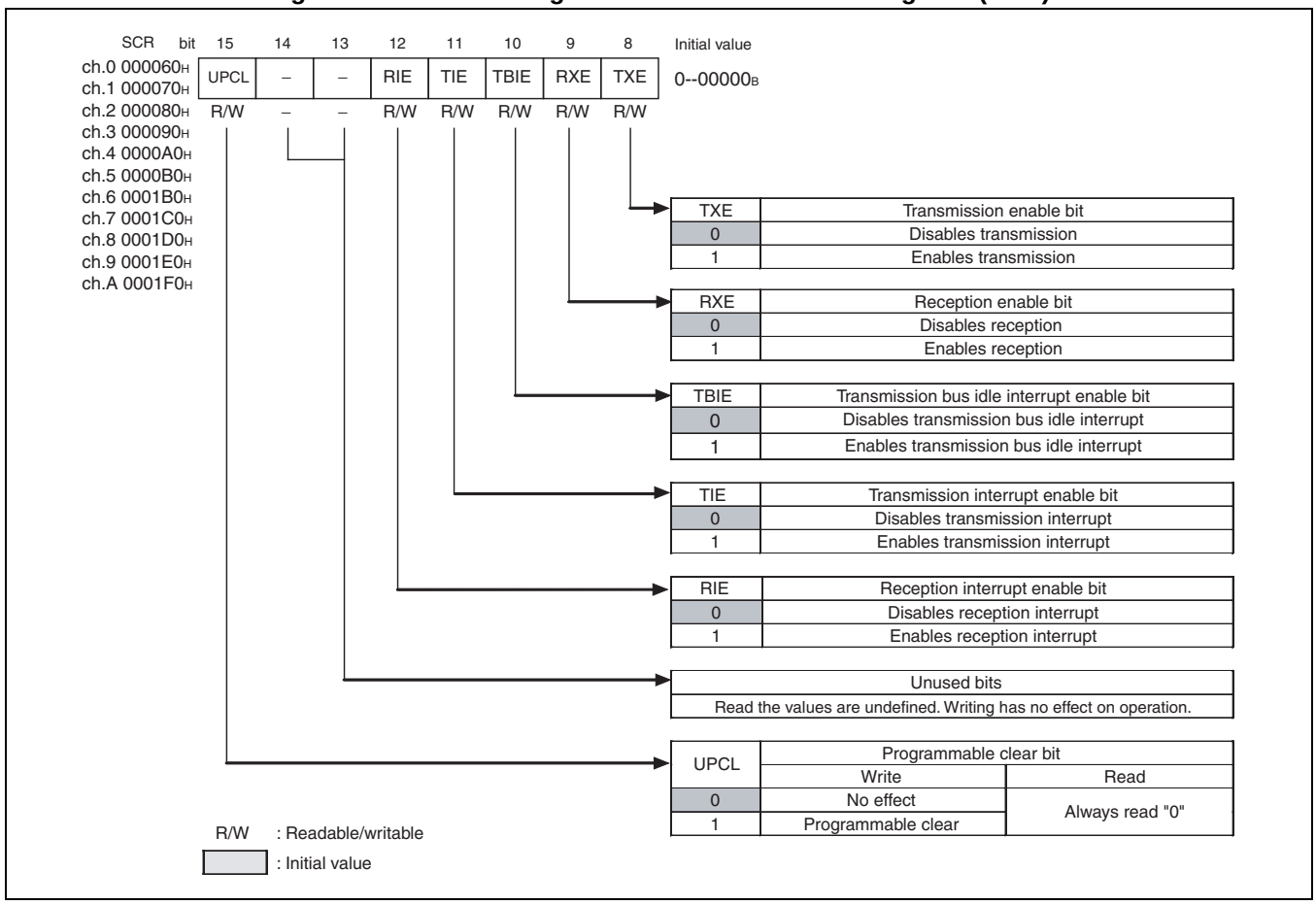


Table 15.3-4 Function Description of Each Bit in the Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	Initializes the internal state of UART. When "1" is set: • UART is directly reset (software reset). However, the register setting is retained. The UART that is under transmission/reception status is immediately disconnected. • Baud rate generator reloads the value set in BGR1/BGR0 register and restarts. • All transmission and reception interrupt sources (PE, FRE, ORE, RDRF, TDRE, and TBI) are initialized (000011_B). When "0" is set: There is no effect. When reading, "0" is always returned. Notes: • Execute programmable clear after you disable an interrupt. • When using FIFO, disable FIFO (FE2, FE1 = 0) before you execute programmable clear.
bit14, bit13	Unused bits	When reading : Values are undefined. When writing : No effect.
bit12	RIE: Reception interrupt enable bit	• Enables/disables the output of a reception interrupt request to the CPU. • A reception interrupt request is output when RIE bit and the reception data flag bit (RDRF) are "1" or when any of the error flag bits (PE, ORE, FRE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	• Enables/disables the output of a transmission interrupt request to the CPU. • A transmission interrupt request is output when TIE bit and the TDRE bit are "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	• Enables/disables the output of a transmission bus idle interrupt request to the CPU. • A transmission bus idle interrupt request is output when TBI bit and the TBIE bit are "1".
bit9	RXE: Reception operation enable bit	Enables/disables the reception operation of UART. • When "0" is set: Disables the reception operation. • When "1" is set: Enables the reception operation. Notes: • Even if the reception operation is enabled (RXE = 1), it will not start until the start bit falling edge (for NRZ format (INV = 0)) is input (For invert NRZ format (INV=1), it will not start until the rising edge is input). • The reception operation is immediately stopped if you disable it (RXE = 0) while receiving.
bit8	TXE: Transmission operation enable bit	Enables/disables the transmission operation of UART. When "0" is set: Disables the transmission operation. When "1" is set: Enables the transmission operation. Note: The transmission operation is immediately stopped if you disable it (TXE = 0) while transmitting.

15.3.2 Serial Mode Register (SMR)

Serial mode register (SMR) can set the operating mode, select transfer direction, data length and stop bit length, and enable/disable the output to the pin of the serial data and clock.

■ Serial Mode Register (SMR)

Figure 15.3-2 shows the bit configuration of the serial mode register (SMR), and Table 15.3-5 shows the function of each bit.

Figure 15.3-2 Bit Configuration of Serial Mode Register (SMR)

SMR
ch.0 000061H
ch.1 000071H
ch.2 000081H
ch.3 000091H
ch.4 0000A1H
ch.5 0000B1H
ch.6 0001B1H
ch.7 0001C1H
ch.8 0001D1H
ch.9 0001E1H
ch.A 0001F1H

bit	7	6	5	4	3	2	1	0	Initial value
	MD2	MD1	MD0	–	SBL	BDS	SCKE	SOE	000–0000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SOE	Serial data output enable bit
0	Disables SOT output
1	Enables SOT output

SCKE	Serial clock output enable bit
0	Disables SCK output or Enables SCK input
1	Enables SCK output

BDS	Transfer direction selection bit
0	LSB first (transferred with the lowest bit)
1	MSB first (transferred with the highest bit)

SBL	Stop bit length selection bit
0	1-bit
1	2-bit

Undefined bit
Read value is undefined. Writing has no effect on operation.

MD2	MD1	MD0	Operating mode setting bits
0	0	0	Operating mode 0 (asynchronous normal mode)
0	0	1	Operating mode 1 (asynchronous multiprocessor mode)
0	1	0	Operating mode 2 (clock synchronous mode)
1	0	0	Operating mode 4 (I²C mode)

R/W : Readable/writable
: Initial value

Note: This chapter explains the register and operation in the operating modes 0 and 1.

Table 15.3-5 Function Description of Each Bit in the Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2, MD1, MD0: Operating mode setting bits	Sets the operating mode of the asynchronous multifunction multi function serial interface. "000_B": Sets to the operating mode 0 (asynchronous normal mode) "001_B": Sets to the operating mode 1 (asynchronous multiprocessor mode) "010_B": Sets to the operating mode 2 (clock synchronous mode) "100_B": Sets to the operating mode 4 (I^2C mode) This chapter explains the register and operation in the operating mode 0 (asynchronous normal mode) and 1 (asynchronous multiprocessor mode). Notes: • Any setting other than those above is disabled. • When switching the operating mode, execute the programmable clear (SCR:UPCL =1) before switching it. • Set the operating mode, and then set each register.
bit4	Unused bit	When reading : Value is undefined. When writing : No effect.
bit3	SBL: Stop bit length selection bit	Specifies the bit length of the stop bit (frame end mark of the transmission data). When "0" is set: Sets the stop bit to 1-bit. When "1" is set: Sets the stop bit to 2-bit. Notes: • When receiving, only the first bit of the stop bit is always detected. • Specify this bit when the transmission is disabled (TXE = 0).
bit2	BDS: Transfer direction selection bit	Specifies whether the transfer serial data is transferred with the lowest bit (LSB first, BDS = 0) or the highest bit (MSB first, BDS = 1). Note: Specify this bit when the transmission/reception are disabled (TXE = RXE= 0).
bit1	SCKE: Serial clock output enable bit	Controls the I/O port of the serial clock. When "0" is set: Enables SCK "H" output or SCK input. When using as SCK input, set the general-purpose I/O port as an input port. In addition, select the external clock by using the external clock selection bit (BGR:EXT = 1). When "1" is set: Enables SCK output.
bit0	SOE: Serial data output enable bit	Enables/disables the output of the serial data. When "0" is set: Enables SOT "H" output. When "1" is set: Enables SOT output.

Note:

Set the operating mode first because the other registers will be initialized once the operating mode has been changed. However, when SCR and SMR are written at the same time by 16-bit writing, the written contents will be reflected on SCR.

15.3.3 Serial Status Register (SSR)

Serial status register (SSR) verifies the transmission/reception status and the reception error flag, and also clears the reception error flag.

■ Serial Status Register (SSR)

Figure 15.3-3 shows the bit configuration of the serial status register (SSR), and Table 15.3-6 shows the function of each bit.

Figure 15.3-3 Bit Configuration of Serial Status Register (SSR)

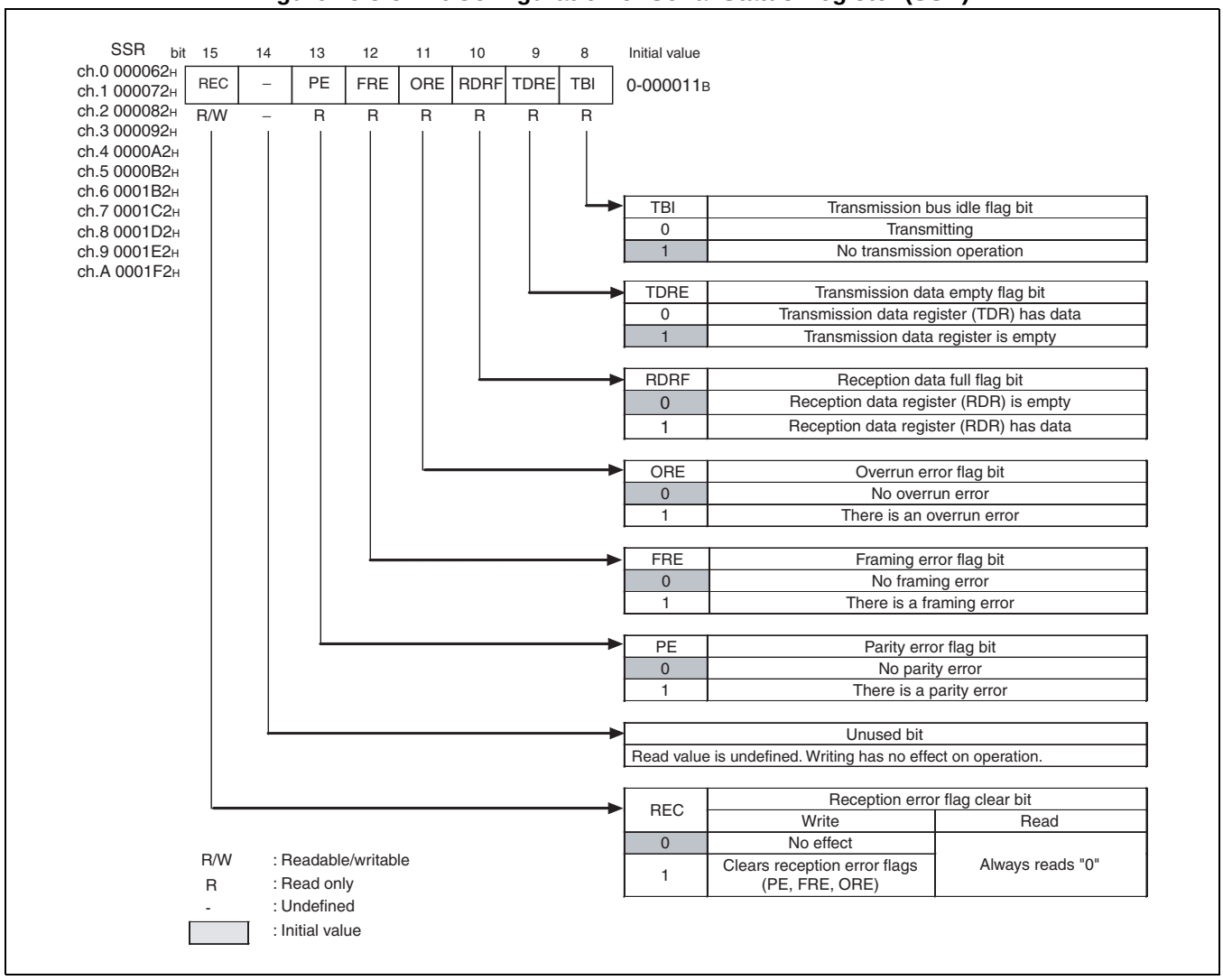


Table 15.3-6 Function Description of Each Bit in the Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	Clears PE, FRE and ORE flags in the serial status register (SSR). • Writing "1" to this bit clears the error flags. • Writing "0" has no effect. When reading, "0" is always returned.
bit14	Unused bit	When reading : Value is undefined. When writing : No effect.
bit13	PE: Parity error flag bit (only operating mode 0 functions)	• SMR: If PEN = 1 and when a parity error occurs while receiving, this bit is set to "1". When you write "1" to REC bit in the serial status register (SSR), this bit is cleared. • A reception interrupt request is output when PE bit and SCR: RIE bit are "1". • Data in the reception data register (RDR) is invalid if this flag is set. • When this bit is set while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the reception data will not be stored into the reception FIFO.
bit12	FRE: Framing error flag bit	• When a framing error occurs while receiving, this bit is set to "1". When you write "1" to REC bit in the serial status register (SSR), this bit is cleared. • A reception interrupt request is output when FRE bit and RIE bit are "1". • Data in the reception data register (RDR) is invalid if this flag is set. • When this bit is set while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the reception data will not be stored into the reception FIFO.
bit11	ORE: Overrun error flag bit	• When an overrun error occurs while receiving, this bit is set to "1". When you write "1" to REC bit in the serial status register (SSR), this bit is cleared. • A reception interrupt request is output when ORE bit and RIE bit are "1". • Data in the reception data register (RDR) is invalid if this flag is set. • When this bit is set while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the reception data will not be stored into the reception FIFO.
bit10	RDRF: Reception data full flag bit	• Indicates the status of the reception data register (RDR). • When the reception data is loaded to RDR, this bit is set to "1". If the reception data register (RDR) is read, this bit is cleared to "0". • A reception interrupt request is output when RDRF bit and RIE bit are "1". • When using the reception FIFO, RDRF is set to "1" when the reception FIFO has received a predefined number of data. • When using the reception FIFO, if the reception FIFO idle detection enable bit (FCR1:FRIIE) is "1" and the reception idle state continues over 8 clocks of the baud rate clock (because the reception FIFO has not received a predefined number of data and some data still remains in the reception FIFO), RDRF is set to "1". If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. • When using the reception FIFO, this bit is cleared to "0" when the reception FIFO gets empty.

Table 15.3-6 Function Description of Each Bit in the Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit9	TDRE: Transmission data empty flag bit	• Indicates the status of the transmission data register (TDR). • When you write a transmission data to TDR, this bit becomes "0" to indicate that there is some valid data. When the data is loaded to the transmission shift register to start transmission, this bit becomes "1" to indicate that there is no valid data in TDR. • A transmission interrupt request is output when TIE bit and the TDRE bit are "1". • If you set UPCL bit in the serial control register (SCR) to "1", TDRE bit becomes "1". • For more information about set/reset timing of TDRE bit when using the transmission FIFO, see "15.4.2 Interrupt Generation and Flag Set Timing When Using Reception FIFO".
bit8	TBI: Transmission bus idle flag bit	• Indicates that UART is not processing the transmission operation. • This bit becomes "0" when transmission data is written to the transmission data register (TDR). • This bit becomes "1" when the transmission data register is empty (TDRE = 1) and the transmission operation is not in progress. • If you set "1" to UPCL bit in the serial control register (SCR), TBI bit becomes "1". • A transmission interrupt request is output when this bit is "1" and the transmission bus idle interrupt is enabled (SCR:TBIE = 1).

15.3.4 Extended Communication Control Register (ESCR)

Extended communication control register (ESCR) can set the transmission/reception data length, enable/disable the parity bit, select the parity bit, and specify the inversion of the serial data format.

■ Bit Configuration of Extended Communication Control Register (ESCR)

Figure 15.3-4 shows the bit configuration of the extended communication control register (ESCR), and Table 15.3-7 shows the function of each bit.

Figure 15.3-4 Bit Configuration of Extended Communication Control Register (ESCR)

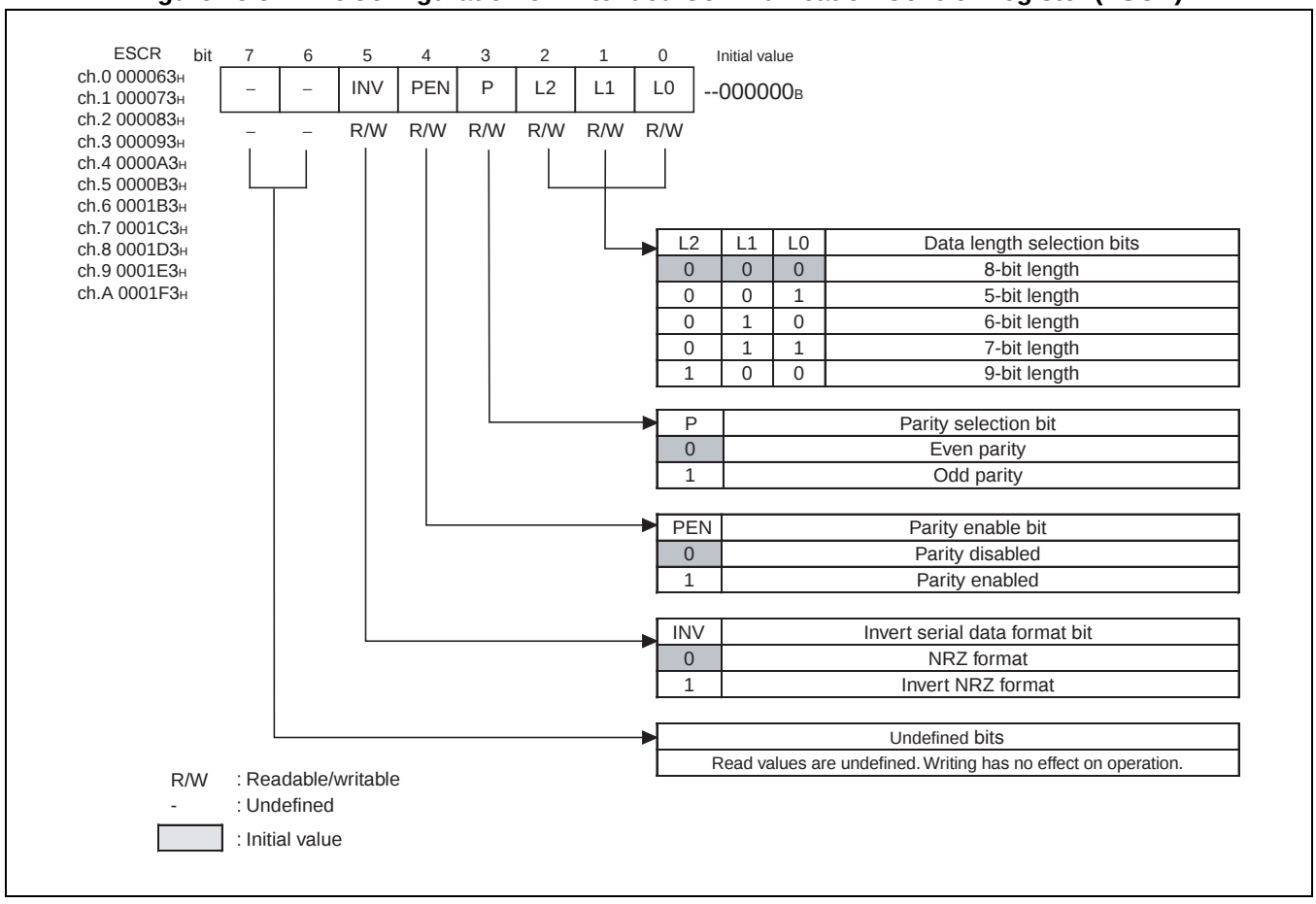


Table 15.3-7 Function Description of Each Bit in the Extended Communication Control Register (ESCR)

Bit name		Function
bit7, bit6	Unused bits	When reading : Values are undefined. When writing : No effect.
bit5	INV: Invert serial data format bit	Specifies whether the serial data format should be NRZ format or invert NRZ format.
bit4	PEN: Parity enable bit (only operating mode 0 functions)	Specifies whether to provide (when transmitting) and detect (when receiving) a parity bit. • When "0" is set: Does not add a parity bit. • When "1" is set: Adds a parity bit. Note: This bit is internally fixed to "0" in the operating mode 1.
bit3	P: Parity selection bit (only operating mode 0 functions)	When parity is enabled (ESCR:PEN = 1), this bit selects even parity "0" or odd parity "1". • When "0" is set: Selects even parity. • When "1" is set: Selects odd parity.
bit2 to bit0	L2 to L0: Data length selection bits	Specifies the data length of the transmission/reception data. • When "000_B" is set: Sets the data length to 8-bit. • When "001_B" is set: Sets the data length to 5-bit. • When "010_B" is set: Sets the data length to 6-bit. • When "011_B" is set: Sets the data length to 7-bit. • When "100_B" is set: Sets the data length to 9-bit. Notes: • Any setting other than those above is disabled. • Sets the data length to 7-bit or 8-bit when using the operating mode 1. Any setting other than this is disabled.

15.3.5 Reception/Transmission Data Registers (RDR/TDR)

Reception/transmission data registers are arranged in the same address. When reading, it functions as the reception data register and when writing, it functions as the transmission data register.
If the FIFO operating is enabled, RDR/TDR address works as FIFO reading/writing address.

■ Reception Data Register (RDR)

Figure 15.3-5 shows the bit configuration of the reception data register (RDR).

Figure 15.3-5 Bit Configuration of the Reception Data Register (RDR)

RDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R	R	R	R	R	R	R	R	R	

R: Read only

Reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal transmitted to the serial input pin (SIN pin) is converted at the shift register and then stored in this reception data register (RDR).
- As described below, "0" is set in the upper bits depending on the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9-bit	X	X	X	X	X	X	X	X	X
8-bit	0	X	X	X	X	X	X	X	X
7-bit	0	0	X	X	X	X	X	X	X
6-bit	0	0	0	X	X	X	X	X	X
5-bit	0	0	0	0	X	X	X	X	X

(X represents the received data bit)

- When the received data is stored in the reception data register (RDR), the reception data full flag bit (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SSR:RIE = 1), a reception interrupt request occurs.
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". If the reception data register (RDR) is read, the reception data full flag bit (SSR:RDRF) is automatically cleared to "0".

- If a reception error has occurred (any of PE, ORE or FRE in SSR is "1"), the data in the reception data register (RDR) becomes invalid.
- For the operating mode 1 (multiprocessor mode), the operation becomes 7-bit or 8-bit length and the received AD bit is stored into D8 bit.
- For 9-bit length transfer and the operating mode 1, RDR reading is performed by 16-bit access.

Notes:

- When using the reception FIFO, RDRF is set to "1" if the reception FIFO has received a predefined number of data.
- When using the reception FIFO, RDRF is cleared to "0" if the reception FIFO gets empty.
- When a reception error occurs (any of PE, ORE or FRE in SSR is "1") while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the received data will not be stored into the reception FIFO.

■ Transmission Data Register (TDR)

Figure 15.3-6 shows the bit configuration of the transmission data register (TDR).

Figure 15.3-6 Bit Configuration of the Transmission Data Register (TDR)

TDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	8	7	6	5	4	3	2	1	0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	111111111_B
	W	W	W	W	W	W	W	W	W	

W: Write only

Transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If data to be transmitted is written to the transmission data register (TDR) when the transmission operation is enabled (SCR:TXE=1), the data is transferred to the transmission shift register where the data is converted to serial data, and then transmitted from the serial data output pin (SOT pin).
- As described below, data becomes invalid from the upper bits depending on the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9-bit	X	X	X	X	X	X	X	X	X
8-bit	Invalid	X	X	X	X	X	X	X	X
7-bit	Invalid	Invalid	X	X	X	X	X	X	X
6-bit	Invalid	Invalid	Invalid	X	X	X	X	X	X
5-bit	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- A transmission data empty flag (SSR:TDRE) is cleared to "0" when the transmission data is written into the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) is set to "1" when the transmission data is transferred to the transmission shift register and the transmission is started.
- When the transmission data empty flag (SSR:TDRE) is "1", you can write the transmission data. If transmission interrupt is enabled, a transmission interrupt request occurs. Write a transmission data when a transmission interrupt occurs or when the transmission data empty flag (SSR:TDRE) is "1".
- You cannot write the transmission data when the transmission data empty flag (SSR:TDRE) is "0" and also the transmission FIFO is disabled or full.
- For the operating mode 1 (multiprocessor mode), the operation becomes 7-bit or 8-bit length and the AD bit is transmitted by writing to D8 bit.
- For 9-bit length transfer and the operating mode 1, writing to TDR is performed by 16-bit access.

Notes:

- The transmission data register (TDR) is a write only register and the reception data register (RDR) is a read only register. These registers are located at the same address, so the read value is different from the write value. Therefore an instruction that operates read-modify-write (RMW) instruction, such as INC/DEC, cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR:TDRE) when using the transmission FIFO, see "15.4.2 Interrupt Generation and Flag Set Timing When Using Reception FIFO".

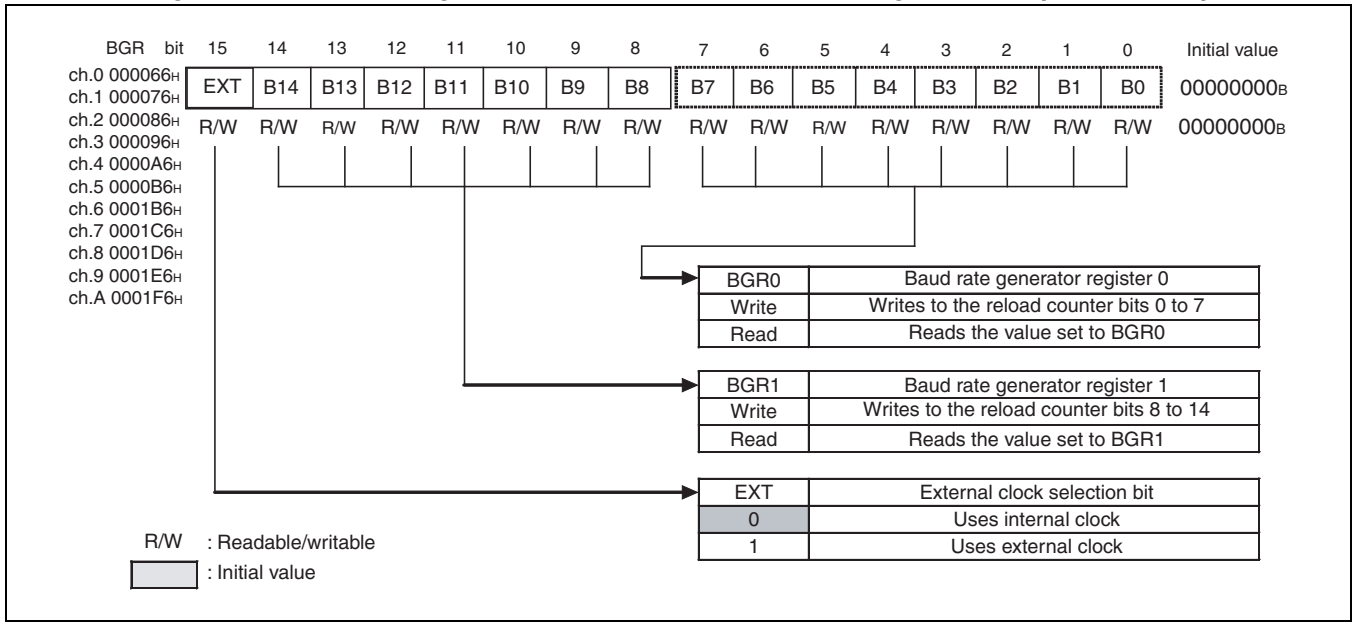
15.3.6 Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

The baud rate generator registers 1, 0 (BGR1/BGR0) sets the division ratio for the serial clock. They can also select an external clock as the clock source of the reload counter.

■ Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

Figure 15.3-7 shows the bit configuration of the baud rate generator registers 1, 0 (BGR1/BGR0).

Figure 15.3-7 Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)



- The baud rate generator registers (BGR) set the division ratio for the serial clock.
- BGR1 (supports the upper bits) and BGR0 (supports the lower bits) can write the reload value to count and read the value set to BGR1/BGR0.
- The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).
- Specify whether to use the internal clock or external clock for a clock source of the reload counter using EXT bit in bit15. EXT = 0 specifies the internal clock. EXT = 1 specifies the external clock.

Notes:

- Write to the baud rate generator registers 1, 0 (BGR1/BGR0) by 16-bit access.
- When you change the value set to the baud rate generator registers 1, 0 (BGR1/BGR0), new value is reloaded after the counter value reaches "000000000000000_B". Therefore, if you want to reflect the change immediately, execute programmable clear (UPCL) after you change the value set to BGR1/BGR0.
- If the reload value is an even number, "L" width of the reception serial clock is longer than "H" width by 1 cycle of the machine clock. If it is an odd number, "H" and "L" widths of the serial clock have the same length.
- Set a value more than 4 to BGR1/BGR0. However, they may not be able to receive data properly depending on a baud rate error and the setting for the reload value.
- If you want to change to the external clock setting (EXT = 1) while operating the baud rate generator, write "0" into the baud rate generator 1, 0 (BGR1/BGR0) and execute programmable clear (UPCL), and then specify the external clock (EXT = 1).

15.3.7 FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the test for FIFO, specifies the transmission/ reception FIFO, enables the transmission FIFO interrupt, and controls the interrupt flag.

■ Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 15.3-8 shows the bit configuration of the FIFO control register 1 (FCR1), and Table 15.3-8 shows the function of each bit.

Figure 15.3-8 Bit Configuration of FIFO Control Register 1 (FCR1)

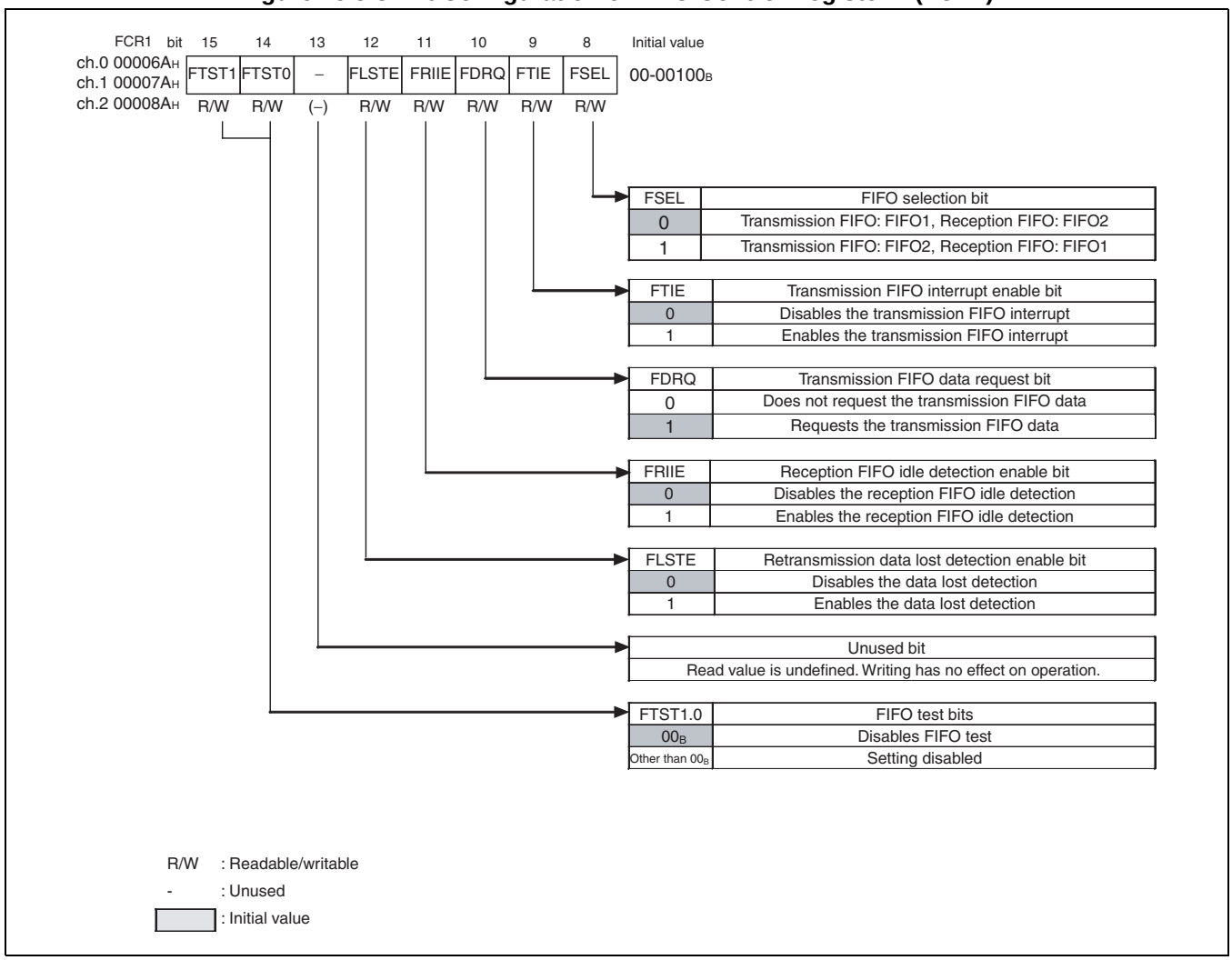


Table 15.3-8 Function Description of Each Bit in the FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	FTST1, FTST0: FIFO test bits	These are the test bits for FIFO. Be sure to set these bits to "0". Note: If you set these bits to "1", the test for FIFO will be executed.
bit13	Unused bit	When reading : Value is undefined. When writing : No effect.
bit12	FLSTE: Retransmission data lost detection enable bit	Enables data lost detection. When "0" is set: Disables data lost detection. When "1" is set: Enables data lost detection. Note: When you set this bit to "1", set FSET bit to "1" before doing so.
bit11	FRIIE: Reception FIFO idle detection enable bit	Specifies whether to detect the reception idle state that continues over 8-bit time with the reception FIFO holding valid data. If the reception interrupt is enabled (SCR:RIE = 1), a reception interrupt occurs when the reception idle state is detected. When "0" is set: Disables the reception FIFO idle state detection. When "1" is set: Enables the reception FIFO idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a data request bit for the transmission FIFO. When this bit is set to "1", it indicates that the transmission data is being requested. A FIFO transmission interrupt request is output when this bit is "1" and the transmission FIFO interrupt is enabled (FTIE = 1). FDRQ set condition FBYTE (for transmission) = 0 (transmission FIFO is empty) FDRQ reset condition • When writing "0" to this bit • When the transmission FIFO gets full Notes: • When the transmission FIFO is enabled, writing "0" is valid. • When FBYTE (for transmission) = 0, writing "0" to this bit is disabled. • Setting "1" to this bit has no effect on the operations. • "1" is read by a read-modify-write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is an interrupt enable bit for the transmission FIFO. If this bit is set to "1", an interrupt occurs when FDRQ bit is "1".
bit8	FSEL: FIFO selection bit	Selects the transmission/reception FIFO. When "0" is set: Assigns the transmission FIFO:FIFO1 and the reception FIFO:FIFO2. When "1" is set: Assigns the transmission FIFO:FIFO2 and the reception FIFO:FIFO1. Notes: • This bit cannot be cleared by the FIFO reset (FCL2, FCL1 = 1). • When you change this bit, disable FIFO operation (FCR:FE2, FE1 = 0) first.

Note:

There are 2 transmission interrupts: transmission FIFO interrupt request and transmission buffer interrupt request.

15.3.8 FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, performs FIFO reset, saves the read pointer, and sets the retransmission.

■ Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 15.3-9 shows the bit configuration of the FIFO control register 0 (FCR0), and Table 15.3-9 shows the function of each bit.

Figure 15.3-9 Bit Configuration of FIFO Control Register 0 (FCR0)

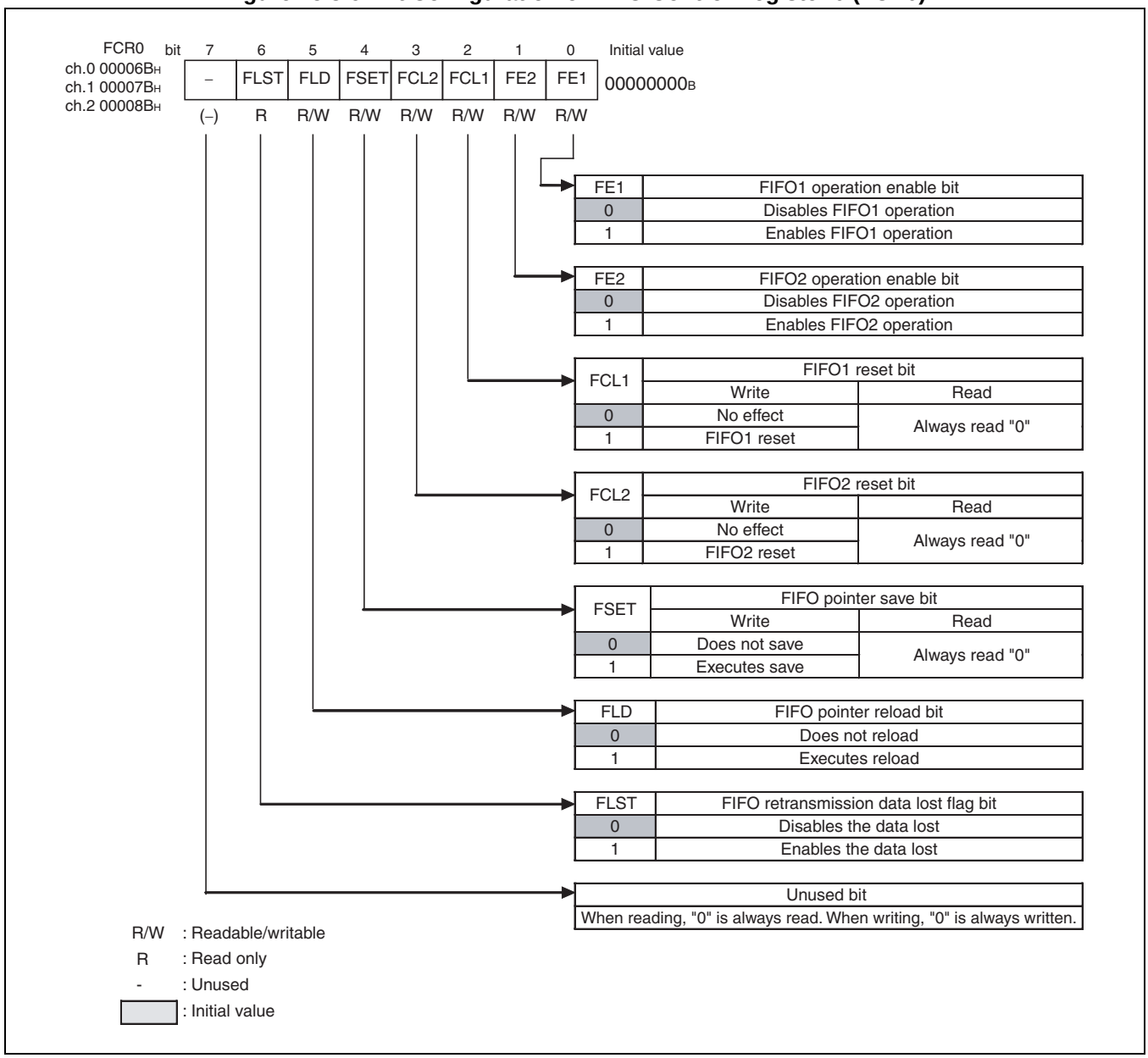


Table 15.3-9 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Unused bit	When reading : "0" is always read. When writing : Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	Indicates that retransmission data of the transmission FIFO has been lost. FLST set condition Writing (Overwriting) to FIFO when FLSTE bit in the FIFO control register 1 (FCR1) is "1" and also the write pointer of the transmission FIFO and the read pointer saved by FSET bit match each other. FLST reset condition • FIFO reset (writing "1" into FCL) • When writing "1" to FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer saved with FSET bit, therefore FLD bit cannot set the retransmission when an error occurs. If you retransmit with this bit set to "1", perform FIFO reset and write the data again into FIFO.
bit5	FLD: FIFO pointer reload bit	Reloads the data saved to the transmission FIFO by FSET bit to the read pointer. This bit is used for the retransmission due to a communication error. When a retransmission setting is completed, this bit becomes "0". Notes: • As long as this bit is set to "1", this bit is reloading to the read pointer. Therefore, do not write anything other than FIFO reset. • Setting this bit to "1" is disabled while transmitting or being in FIFO enabled state. • Set TIE and TBIE bits to "0" and then write "1" into this bit. After you enabled the transmission FIFO, set TIE and TBIE bits to "1".
bit4	FSET: FIFO pointer save bit	Saves the read pointer of the transmission FIFO. Once you save the read pointer before communication, when FLST bit is "0" it is possible to retransmit in the case a communication error. When "1" is set: Retains the current value set to the read pointer. When "0" is set: No effect. Note: Set this bit to "1" when the number of bytes for transmission (FBYTE) indicates "0".
bit3	FCL2: FIFO reset bit	Resets FIFO2. If you set this bit to "1", the internal state of FIFO2 is initialized. Only the FLST bit in the FIFO control register 1 (FCR1) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables any transmission/reception before performing FIFO2 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE2 register becomes "0".
bit2	FCL1: FIFO1 reset bit	Resets FIFO1. If you set this bit to "1", the internal state of FIFO1 is initialized. Only the FLST bit in the FIFO control register 1 (FCR1) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables any transmission/reception before performing FIFO1 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE1 register becomes "0".

Table 15.3-9 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit1	FE2: FIFO2 operation enable bit	Enables/disables the operation of FIFO2. • When using FIFO2, set this bit to "1". • Set FIFO2 to the transmission FIFO (FCR1:FSEL = 1). When there is some data in FIFO2 and UART is enabled for transmission (TXE = 1), writing "1" into this bit starts transmitting immediately. Set TIE and TBIE bits to "0" and then write "1" into this bit to make TIE and TBIE bits to "1". • If specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • When using as the transmission FIFO, set "1" or "0" to this bit when the transmission buffer is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception buffer is empty (RDRF = 0). • Even if FIFO2 is disabled, the state of FIFO2 is still retained.
bit0	FE1: FIFO1 operation enable bit	Enables/disables the operation of FIFO1. • When using FIFO1, set this bit to "1". • Set FIFO1 to the transmission FIFO (FCR1:FSEL = 0). When there is some data in FIFO1 and UART is enabled for transmission (TXE = 1), writing "1" into this bit starts transmitting immediately. Set TIE and TBIE bits to "0" and then write "1" into this bit to make TIE and TBIE bits to "1". • If specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • If using as the transmission FIFO, set "1" or "0" to this bit when the transmission buffer is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception buffer is empty (RDRF = 0). • Even if FIFO1 is disabled, the state of FIFO1 is still retained.

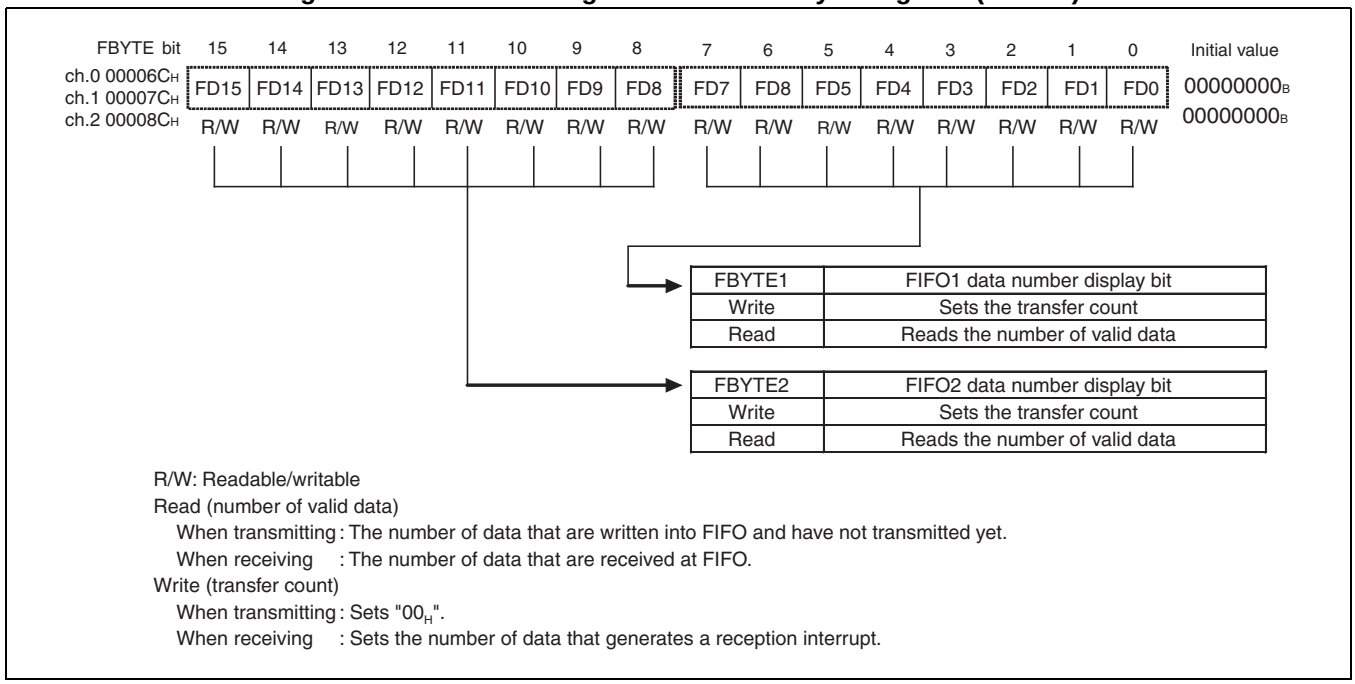
15.3.9 FIFO Byte Register (FBYTE)

FIFO byte register (FBYTE) indicates the number of valid data for FIFO. In addition, it can specify whether to generate a reception interrupt when the predefined number of data has been received at the reception FIFO.

■ Bit Configuration of FIFO Byte Register (FBYTE)

Figure 15.3-10 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 15.3-10 Bit Configuration of FIFO Byte Register (FBYTE)



FIFO byte register (FBYTE) indicates the number of valid data written or received into FIFO. The number varies as follows, depending on the setting of FSEL bit in the FCR1 register.

Table 15.3-10 The Number of Data Displayed

FSEL	FIFO selection	The number of data displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value of the transfer count of the FBYTE register is "08_H".
- Set the number of data to generate a reception interrupt flag to the FBYTE in the reception FIFO. An interrupt flag (RDRF bit in the SSR) is set to "1" when the defined transfer count matches with the number of data displayed in the FIFO byte register (FBYTE).

- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.

Notes:

- Set "00_H" to the FIFO byte register (FBYTE) in the transmission FIFO.
- Set a data more than "1" to the FBYTE in the reception FIFO.
- Disable the reception before you change the setting.
- You cannot use any read-modify-write (RMW) instruction to this register.
- The setting that exceeds the FIFO size is disabled.

15.4 Interrupts of UART

UART uses both reception and transmission interrupts. An interrupt request can be generated by the following sources:

- **When the received data is set in the reception data register (RDR), or a reception error occur.**
- **When the transmission data is transferred from the transmission data register (TDR) to the transmission shift register and the transmission has started.**
- **Transmission bus idle (no transmission operation)**
- **Transmission FIFO data request**

■ Interrupts of UART

Table 15.4-1 shows the UART interrupt control bit and the interrupt source.

Table 15.4-1 UART Interrupt Control Bit and Interrupt Source

Interrupt type	Interrupt request flag bit	Flag register	Operating mode		Interrupt source	Interrupt source enable bit	How to clear the interrupt request flag
			0	1			
Reception	RDRF	SSR	○	○	Receive 1 byte	SCR:RIE	Read reception data (RDR)
					The value set to FBYTE is received		Read reception data (RDR) until the reception FIFO gets empty
					Detect the reception idle state that continues over 8-bit time with the reception FIFO holding valid data when FRIIE bit is "1"		
	ORE	SSR	○	○	Overrun error		Write "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	○	○	Framing error		
	PE	SSR	○	×	Parity error		
Transmis-sion	TDRE	SSR	○	○	Transmission register is empty	SCR:TIE	Write to the transmission data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)*
	TBI	SSR	○	○	No transmission operation	SCR:TBIE	Write to the transmission data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)*
	FDRQ	FCR1	○	○	Transmission FIFO is empty	FCR1:FTIE	Write "0" to the FIFO transmission data request bit (FCR1:FDRQ), or the transmission FIFO is full

*: Set TIE bit to "1" after TDRE bit has become "0".

15.4.1 Reception Interrupt Generation and Flag Set Timing

A reception completion (RDRF bit in SSR) and a reception error occurrence (PE, ORE, FRE bits in SSR) are the sources of the reception interrupt.

■ Reception Interrupt Generation and Flag Set Timing

Detection of the first stop bit stores the reception data to the reception data register (RDR). When a reception is completed (SSR:RDRF = 1) or a reception error occurs (SSR:PE, ORE, FRE = 1), each flag is set. If reception interrupt is enabled (SSR:RIE = 1) at this point, a reception interrupt occurs.

Note:

If a reception error occurs, the data in the reception data register (RDR) becomes invalid.

Figure 15.4-1 Set Timing of RDRF (Reception Data Full) Flag Bit

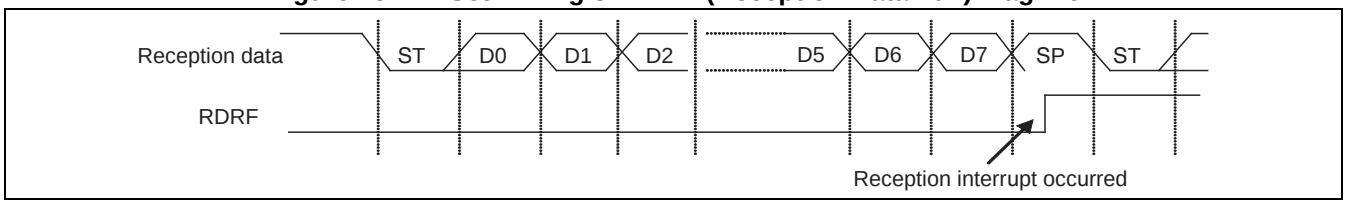
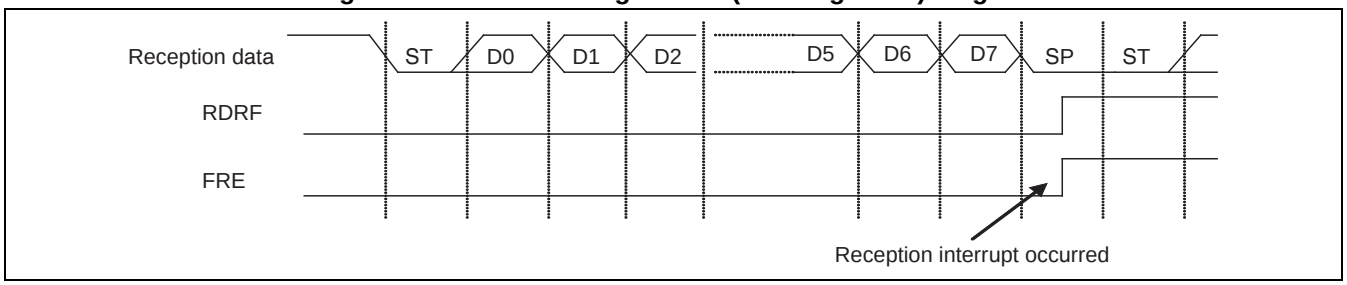


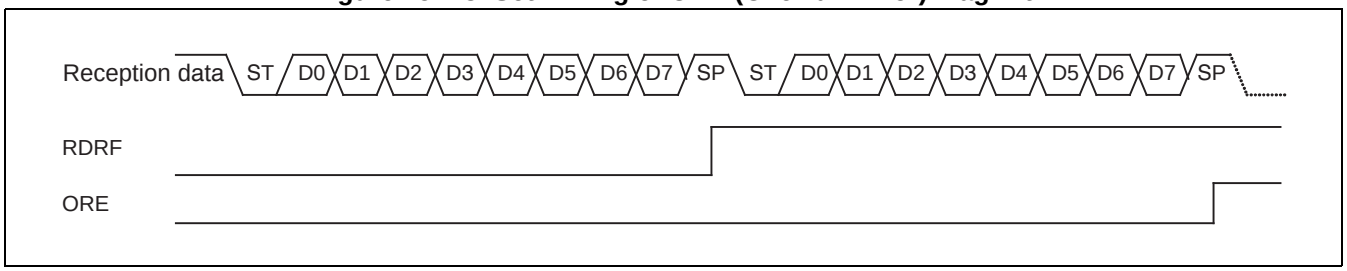
Figure 15.4-2 Set Timing of FRE (Framing Error) Flag Bit



A framing error occurs when the first stop bit is "L" level.

Even when a framing error occurs, RDRF is set to "1" to receive the data however, the reception data becomes invalid.

Figure 15.4-3 Set Timing of ORE (Overrun Error) Flag Bit



An overrun error occurs when next data is transferred before the reception data is read (RDRF = 1).

15.4.2 Interrupt Generation and Flag Set Timing When Using Reception FIFO

When using the reception FIFO, an interrupt is generated when the same value set to the FIFO byte register (FBYTE) has been received.

■ Reception Interrupt Generation and Flag Set Timing When Using Reception FIFO

When using the reception FIFO, the generation of an interrupt depends on the value set to the FIFO byte register (FBYTE).

- When the amount of data set to the FIFO byte register (FBYTE) as the transfer count has been received, the reception data full flag bit in the serial status register (RDRF bit in SSR) is set to "1". If reception interrupt is enabled (SCR:RIE) at this point, a reception interrupt occurs.
- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.
- When the reception data (RDR) is read until the reception FIFO gets empty, the reception data full flag bit (SSR:RDRF) is cleared.
- If next data is received with the FIFO size displayed as the number of data that can be received, an overrun error occurs (SSR:ORE = 1).

Figure 15.4-4 Reception Interrupt Generation Timing When Using Reception FIFO

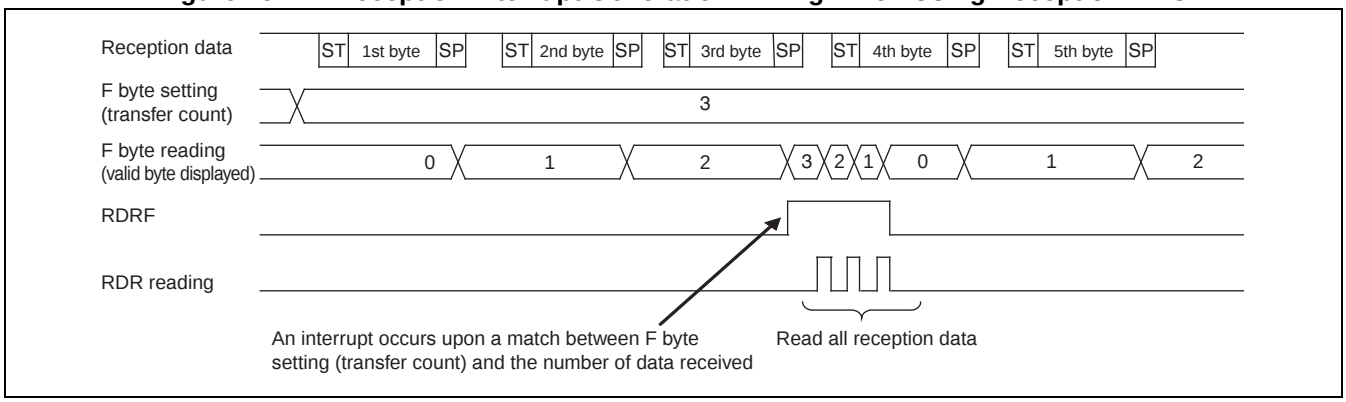
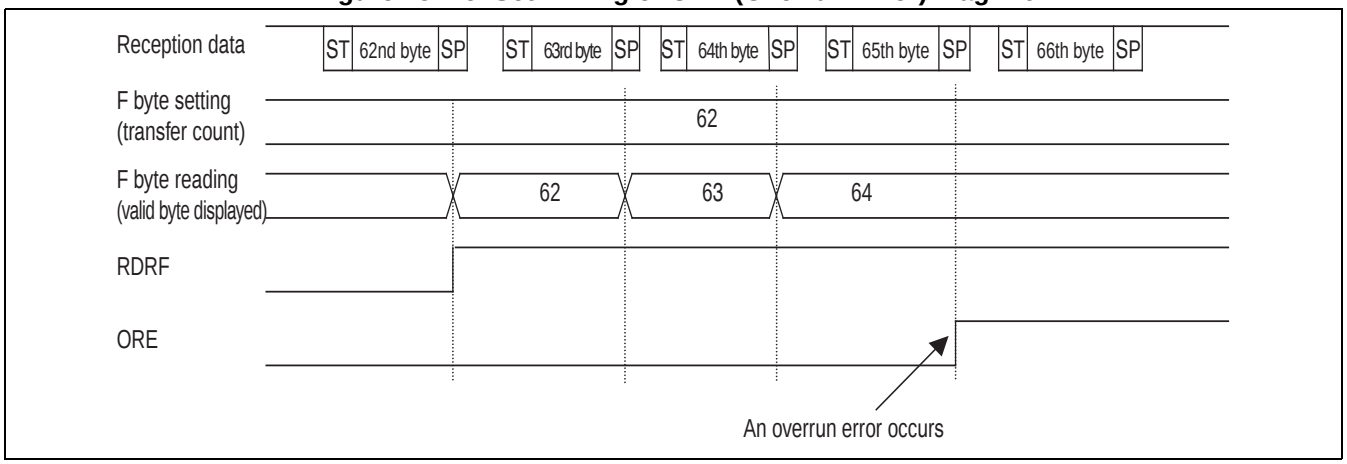


Figure 15.4-5 Set Timing of ORE (Overrun Error) Flag Bit



An overrun error occurs when next data is received with the FIFO size displayed as the F byte reading. The figure shows an example where a FIFO size of 64 bytes is used.

15.4.3 Transmission Interrupt Generation and Flag Set Timing

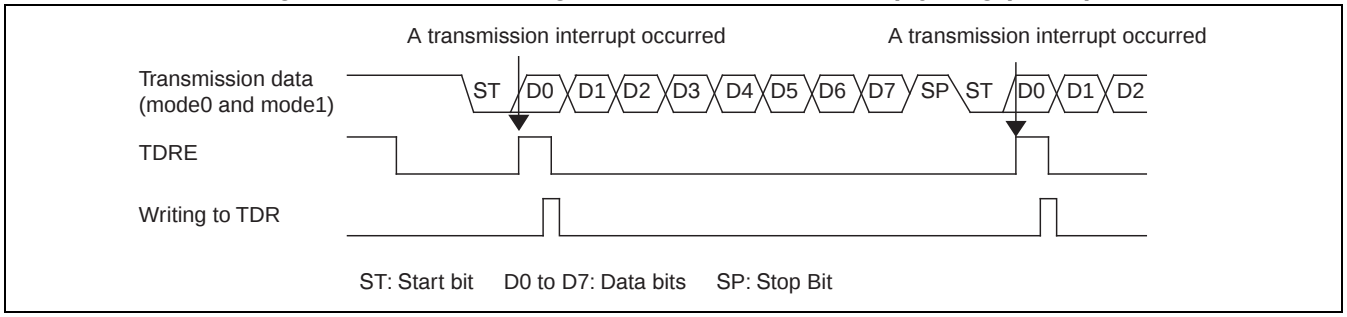
A transmission interrupt occurs when the transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and the transmission has started, or when the transmission operation is not in progress (SSR:TBI=1).

■ Transmission Interrupt Generation and Flag Set Timing

- Set timing of the transmission data empty flag (TDRE)

  Once the data written into the transmission data register (TDR) is transferred to the transmission shift register, next data can be written (SSR:TDRE = 1). If transmission interrupt is enabled (SCR:TIE = 1) at this point, a transmission interrupt occurs. As TDRE bit is read only, it is cleared to "0" by writing data into the transmission data register (TDR).

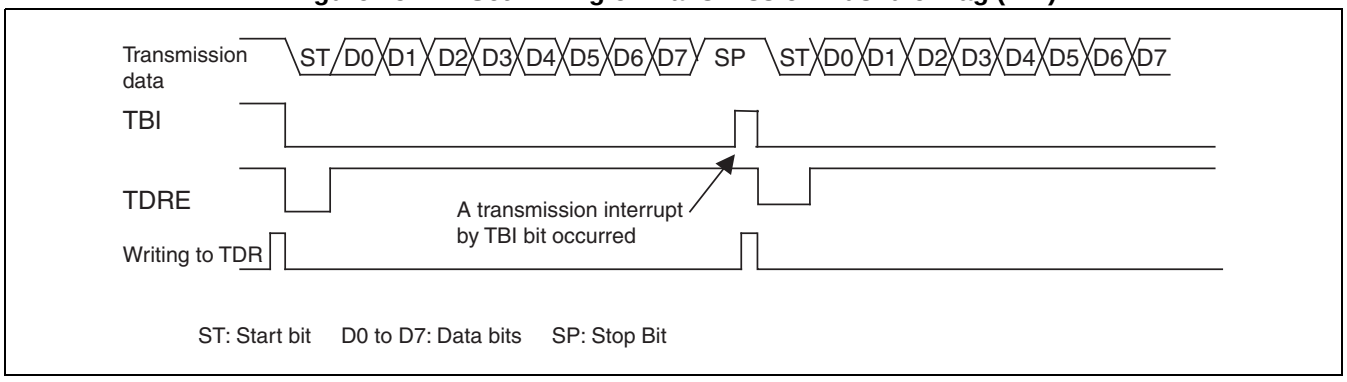
Figure 15.4-6 Set Timing of Transmission Data Empty Flag (TDRE)



- Set timing of the transmission bus idle flag (TBI)

  TBI bit in the serial status register (SSR) is set to "1" when the transmission data register is empty (TDRE = 1) and the transmission operation is not in progress. If transmission bus idle interrupt is enabled (SCR:TBIE = 1) at this point, a transmission interrupt occurs. TBI bit and the transmission interrupt request are cleared when a transmission data is set to the transmission data register (TDR).

Figure 15.4-7 Set Timing of Transmission Bus Idle Flag (TBI)



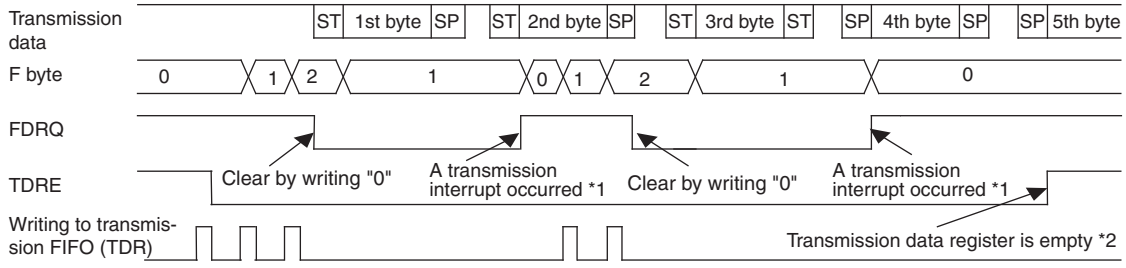
15.4.4 Interrupt Generation and Flag Set Timing When Using Transmission FIFO

When using the transmission FIFO, an interrupt occurs when the transmission FIFO has no data.

■ Transmission Interrupt Generation and Flag Set Timing When Using Transmission FIFO

- When the transmission FIFO has no data, FIFO transmission data request bit (FCR1:FDRQ) is set to "1".
  If FIFO transmission interrupt is enabled (FCR1:FTIE = 1) at this point, a transmission interrupt occurs.
- Once the transmission interrupt has been generated and you have written the required data into the transmission FIFO, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- The FIFO transmission data request bit (FCR1:FDRQ) becomes "0" when the transmission FIFO gets full.
- You can verify if the transmission FIFO has data or not by reading the FIFO byte register (FBYTE).
  FBYTE = 00_H indicates that the transmission FIFO has no data.

Figure 15.4-8 Transmission Interrupt Generation Timing When Using Transmission FIFO



*1: FDRQ is set to "1" as the transmission FIFO is empty.

*2: TDRE is set to "1" as the transmission shift register and the transmission buffer register have no data.

MB91313A Series

15.5 Operations of UART

UART operates in the mode 0 for bidirectional serial asynchronous communication and the mode 1 for master/slave multiprocessor communication.

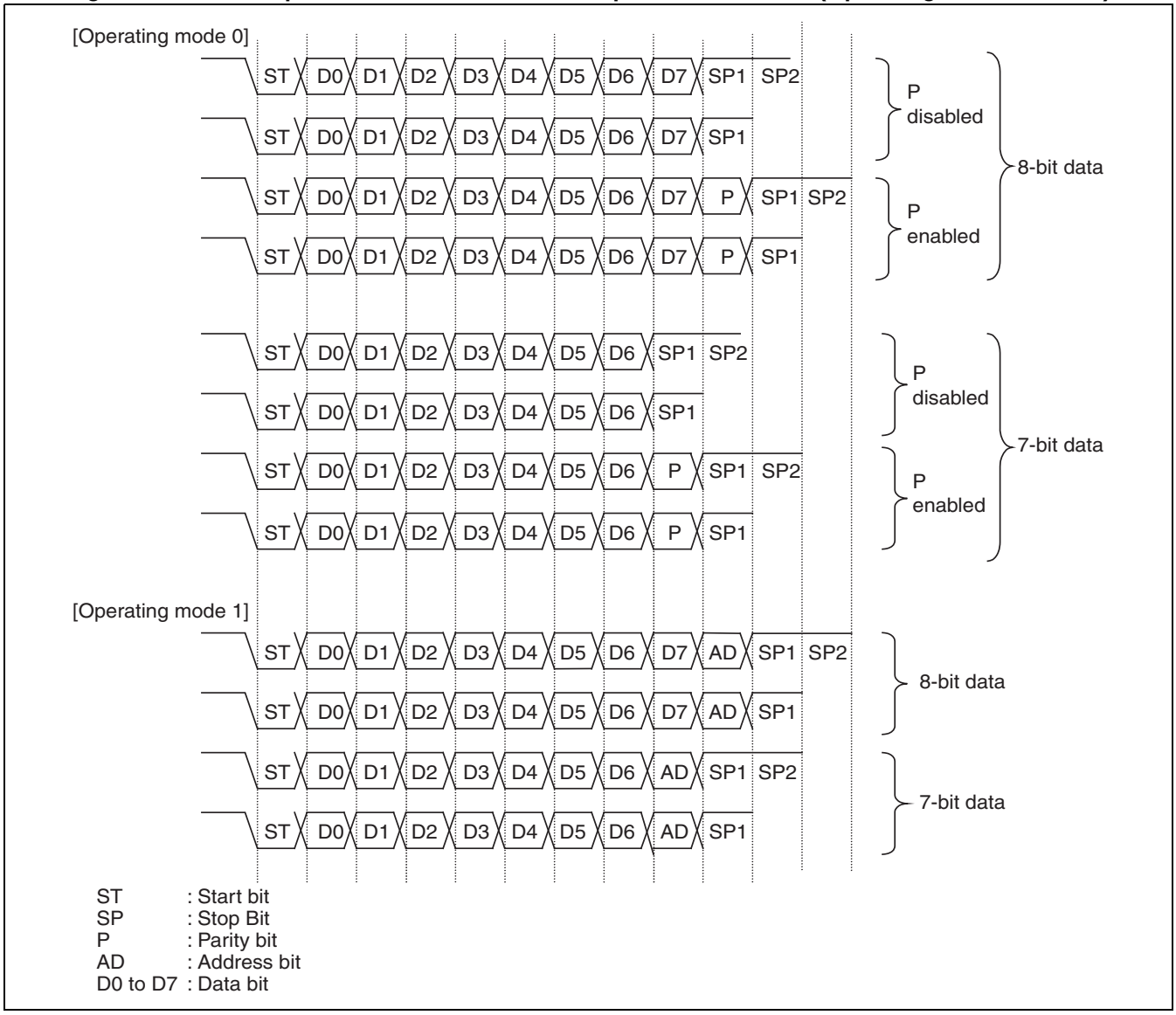
■ Operations of UART

● Transmission and reception data formats

- The transmission and reception data always start with the start bit, continue as long as the data bit length specified, and end with at least 1 bit of the stop bit.
- The data transfer direction (LSB first or MSB first) is determined by BDS bit in the serial mode register (SMR). If parity is enabled, a parity bit is always placed between the last data bit and the first stop bit.
- In operating mode 0 (normal mode), you can select whether to enable or disable the parity.
- In operating mode 1 (multiprocessor mode), AD bit is added instead of the parity.

Figure 15.5-1 shows the transmission and reception data formats in operating modes 0 and 1.

Figure 15.5-1 Example of Transmission and Reception Data Format (Operating Modes 0 and 1)



Notes:

- The figure above shows the case where the data length is set to 7 and 8 bits (the data length can be set between 5 and 9 bits in operating mode 0).
- If BDS bit in the serial mode register (SMR) is set to "1" (MSB first), the bit is processed in order as D7, D6, D5, ..., D1, D0, (P).
- If the data length is set to X-bit, the lower X-bit in the transmission and reception data registers (RDR/TDR) are enabled.

● Transmission operation

- If the transmission data empty flag bit (TDRE) in the serial status register (SSR) is "1", you can write a transmission data into the transmission data register (TDR) (if the transmission FIFO is enabled, you can also write the transmission data when TDRE = 0).
- The transmission data empty flag bit (TDRE) is cleared to "0" when a transmission data is written into the transmission data register (TDR).
- When the transmission operation enable (TXE) bit in the serial control register (SCR) is set to "1", the transmission data is loaded to the transmission shift register and the transmission starts with the start bit.
- Once the transmission has started, the transmission data empty flag bit (TDRE) is set back to "1". If transmission interrupt is enabled (SCR:TIE = 1) at this point, a transmission interrupt occurs. In the interrupt process, next transmission data can be written to the transmission data register.

Notes:

- Since the initial value of the transmission data empty bit flag (TDRE) in the serial status register (SSR) is "1", a transmission interrupt will occur immediately once the transmission interrupt is enabled (SCR:TIE).
- Since the initial value of the FIFO transmission data request bit (FDRQ) in the FIFO control register 1 (FCR1) is "1", a transmission interrupt will occur immediately once the FIFO transmission interrupt is enabled (FCR1:FTIE = 1).

● Reception operation

- A reception operation is performed when the reception operation is enabled (SCR:RXE = 1).
- When a start bit is detected, 1 frame data is received according to the data format set to the extended communication control register (ESCR:PEN, P, L2, L1, L0) and the serial mode register (SMR:BDS).
- When the reception of 1 frame of data is completed, the reception data full flag bit (SSR:RDRF) is set to "1". If reception interrupt is enabled (SCR:RIE = 1) at this point, a reception interrupt occurs.
- If you want to read the received data, wait until the reception of 1 frame of data is completed to verify the status of the error flag in the serial status register (SSR). If a reception error occurs, correct the error.
- When the received data is read, the reception data full flag bit (SSR:RDRF) is cleared to "0".
- When the reception FIFO is enabled, the reception data full flag bit (SSR:RDRF) is set to "1" after the certain frames set to the reception FBYTE have been received.
- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.

- If the reception FIFO is enabled, when the error flag in the serial status register (SSR) is set to "1", the data that generated the error is not stored into the reception FIFO. In this case, the reception data full flag bit (SSR:RDRF) is not set to "1" (however, RDRF flag will be set to "1" when it is an overrun error). The reception FBYTE displays the number of data that has been successfully received before the error. The reception FIFO will not be enabled until the error flag in the serial status register (SSR) is cleared to "0".
- If the reception FIFO is enabled, reception data full flag bit (SSR:RDRF) is cleared to "0" when the reception FIFO gets empty.

Note:

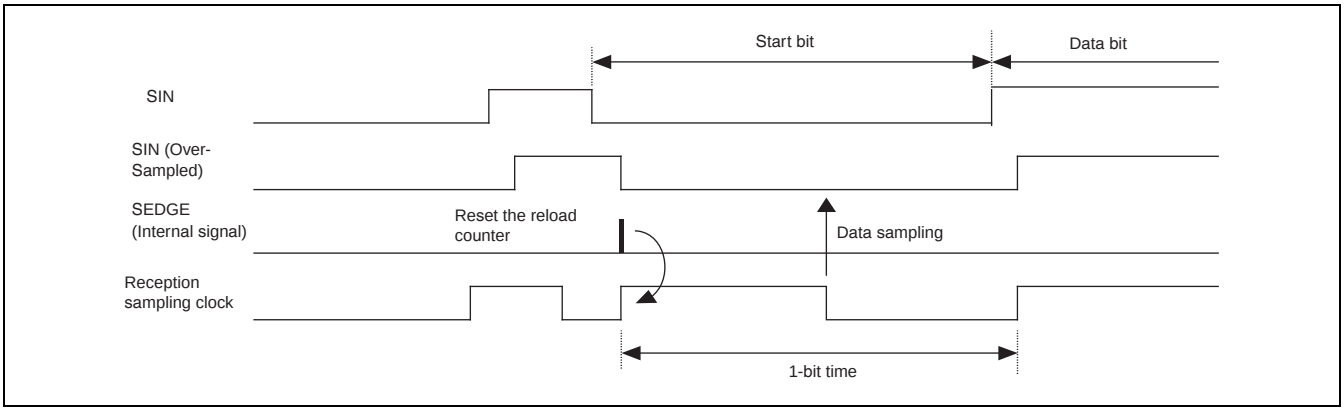
The data in the reception data register (RDR) becomes valid when the reception data full flag bit (SSR:RDRF) is set to "1" and no reception error is generated (SSR:PE, ORE, FRE = 0).

● Clock selection

- You can use the internal clock or the external clock.
- When using an external clock, set EXT bit in the serial mode register (SMR) to "1". If so, the external clock is divided by the baud rate generator.

● Start bit detection

- In asynchronous mode, the start bit is detected by the falling edge of the SIN signal.
  Therefore, the reception operation does not start without inputting the falling edge of the SIN signal, even if the reception operation is enabled (SCR:RXE = 1).
- When the falling edge of the start bit is detected, the reception reload counter of the baud rate generator is reset to reload and starts counting down. This makes it possible to take a sample always in the center of the data.



● Stop bit

- You can choose 1-bit or 2-bit length.
- The reception data full flag bit (SSR:RDRF) is set to "1" when the first stop bit is detected.

● Error detection

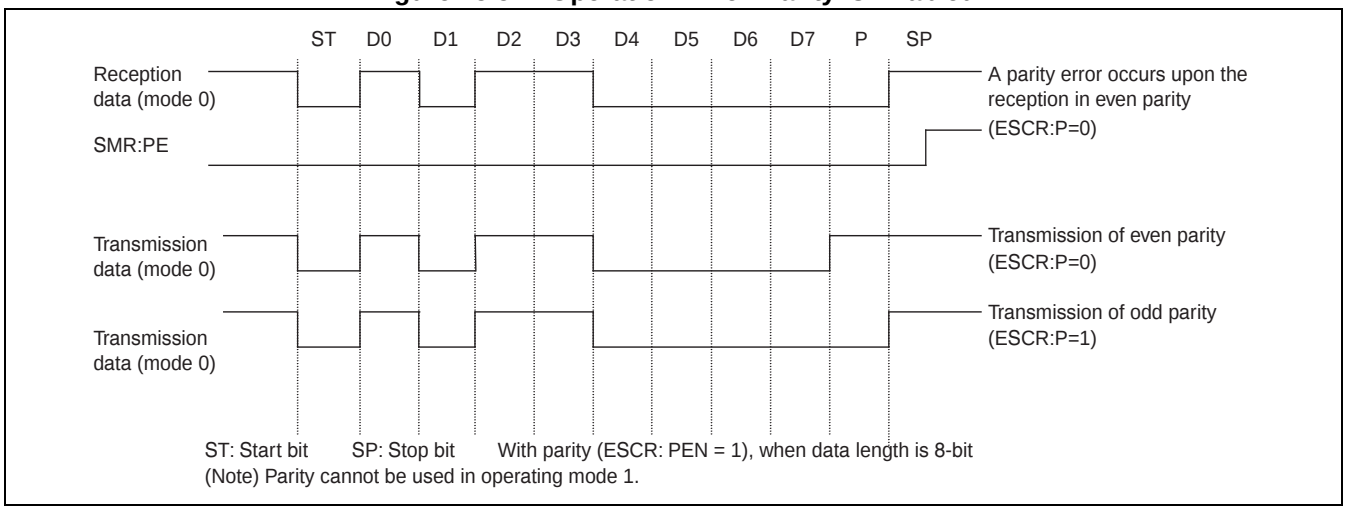
- In operating mode 0, parity, overrun, and framing errors can be detected.
- In operating mode 1, overrun and framing errors can be detected. Parity error cannot be detected.

● Parity bit

- Parity bit can be added only in the operating mode 0. The parity enable bit (ESCR:PEN) can specify whether to enable or disable the parity, and the parity selection bit (ESCR:P) can specify whether to use even parity or odd parity.
- Parity cannot be used in operating mode 1.

Figure 15.5-2 shows the operation when parity is enabled.

Figure 15.5-2 Operation When Parity Is Enabled

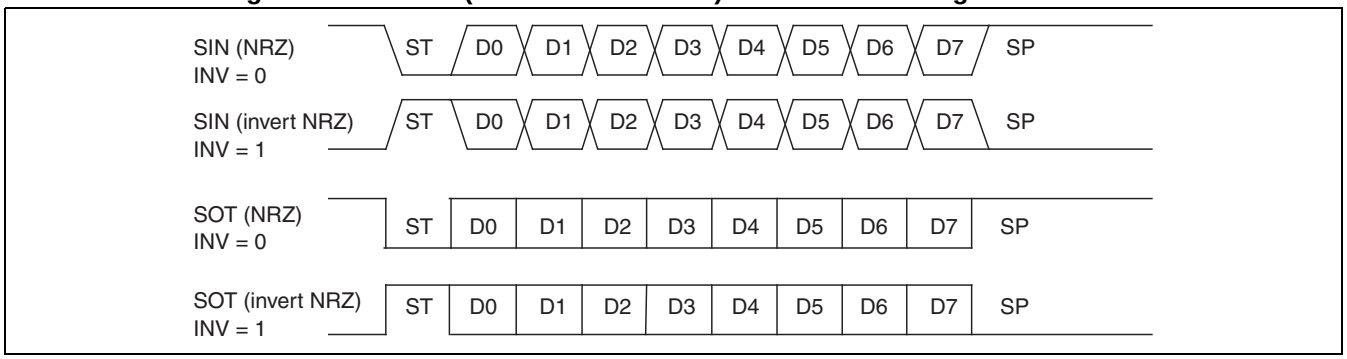


● Data signal method

You can specify the NRZ (Non Return to Zero) (ESCR:INV = 0) or invert NRZ (ESCR:INV = 1) signal methods depending on the setting of the INV bit in the extended communication control register.

Figure 15.5-3 shows the NRZ (Non Return to Zero) and invert NRZ signal methods.

Figure 15.5-3 NRZ (Non Return to Zero) and Invert NRZ Signal Methods



● Data transfer method

You can select the LSB first or MSB first as the data bit transfer method.

15.6 UART Dedicated Baud Rate Generator

One of the followings can be selected for the transmission and reception clock source of UART:

- **Dedicated baud rate generator (reload counter)**
- **Input an external clock to the baud rate generator (reload counter)**

■ UART Baud Rate Selection

You can select 1 type of baud rate out of the following 2 types:

● Baud rate that can be obtained by dividing the internal clock with the dedicated baud rate generator (reload counter)

There are 2 internal reload counters that correspond to transmission and reception serial clocks. Baud rate can be specified by setting the reload value for 15 bits with the baud rate generator registers 1, 0 (BGR1/BGR0).

The reload counter divides the internal clock by the set value.

To set the clock source, select the internal clock (SMR:EXT = 0).

● Baud rate that can be obtained by dividing the external clock with the dedicated baud rate generator (reload counter)

An external clock is used as the clock source of the reload counter.

Baud rate can be specified by setting the reload value for 15 bits with the baud rate generator registers 1, 0 (BGR1/BGR0).

The reload counter divides the external clock by the set value.

To set the clock source, select the external clock and enable the baud rate generator clock (SMR:EXT = 1).

This mode is provided for the case where the resonator that has special frequency is divided to use.

Notes:

- Be sure to specify the external clock (EXT = 1) after the reload counter has stopped (BGR1/BGR0 = 15"H00).
- When you specified to the external clock (EXT = 1), the minimum widths required for "H" and "L" of the external clock is 2-machine clock or more.

MB91313A Series

15.6.1 Baud Rate Setting

This section shows the setting for the baud rate. The calculation result of serial clock frequency is also described.

■ Calculating Baud Rate

2 of 15-bit reload counters are set using the baud rate generator registers 1, 0 (BGR1/BGR0).

The equation to calculate the baud rate is shown below:

(1) Reload value

$V = \phi / b - 1$

V: Reload value
b : Baud rate, ϕ: Machine clock, External clock frequency

(2) Example of calculation

Where the machine clock is 16 MHz, the internal clock is enabled, and the baud rate is set to 19200 bps, the reload value is calculated as follows:

Reload value:
$V = (16 \times 1000000)/19200 - 1 = 832$
Therefore, the baud rate is:
$b = (16 \times 1000000)/(832+1) = 19208$ bps

(3) Baud rate error

Baud rate error can be calculated by the following equation:

Error (%) = (calculated value - desired value) / desired value $\times$ 100

(Example) Where the machine clock is 20 MHz and the desired baud rate is set to 153600 bps:

Reload value = $(20 \times 10000000) / 153600 - 1 = 129$
Baud rate (calculated value) = $(20 \times 10000000) / (129 + 1) = 153846$ (bps)
Error (%) = (153846 -153600) / 153600 x100 = 0.16 (%)

Notes:

- If the reload value is set to "0", the reload counter will stop.
- If the reload value is an even number, "L" width of the reception serial clock is longer than "H" width by 1 cycle of the machine clock. If it is an odd number, "H" and "L" widths of the serial clock have the same length.
- Be sure to set the reload value to 4 or more. However, data may not be able to receive properly depending on a baud rate error and the setting for the reload value.

■ Reload Value and Baud Rate for Each Machine Clock Frequency

Table 15.6-1 Reload Value And Baud Rate

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4 M	-	-	-	-	-	0	4	0	5	0	7	0
2.5 M	-	-	-	0	-	-	-	-	-	-	-	-
2 M	-	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

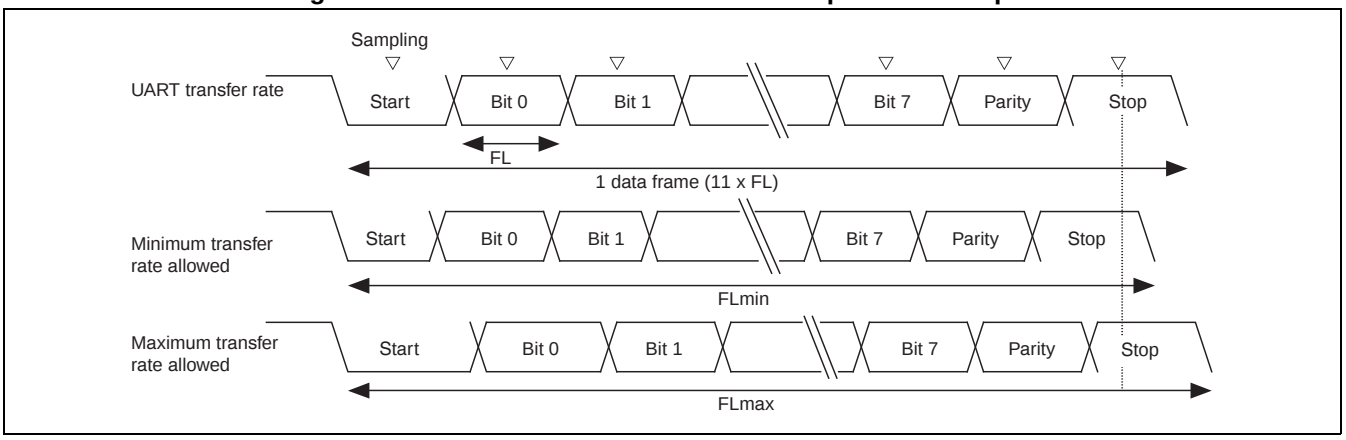
- Value : Value set to BGR1/BGR0 registers (decimal)
- ERR : Baud rate error (%)

■ Baud Rate Tolerance Level upon the Reception

The baud rate tolerance level upon the reception is shown as follows.

Make sure that the baud rate error upon the reception falls into the tolerance level using the equation shown below.

Figure 15.6-1 Baud Rate Tolerance Level upon the Reception



As described in the figure, once the start bit is detected, the counter set in the BGR1/BGR0 registers decide the sampling timing of the reception data. If the last data (stop bit) can be included in this sampling timing, the data is received successfully.

If this is applied to 11-bit reception, the tolerance level can theoretically be calculated as follows:

Where the margin of the sampling timing is 2 clocks of the machine clock (ϕ), the minimum transfer rate allowed (FLmin) can be as follows:

FLmin = (11 bit × (V+1) - (V+1)/2 + 3)/ϕ = (21V+27)/2ϕ (s)
V: Reload value, ϕ: Machine clock

Therefore, the receivable maximum baud rate (BGmax) for the destination is as follows:

BGmax = 11/FLmin = 22ϕ/(21V+27) (bps) V: Reload value, ϕ: Machine clock

Similarly, the maximum transfer rate allowed (FLmax) can be calculated as follows:

FLmax = (11 bits × (V+1) + (V+1)/2 - 3)/ϕ = (23V+17)/2ϕ (s)
V: Reload value, ϕ: Machine clock

Therefore, the receivable minimum baud rate (BGmin) for the destination is as follows:

BGmin = 11/FLmax = 22ϕ/(23V+17) (bps) V: Reload value, ϕ: Machine clock

The following table shows the allowable error between UART and the destination baud rate calculated by the equations for the maximum/minimum baud rate value described above:

Reload value (V)	Maximum baud rate error allowed	Minimum baud rate error allowed
3	0%	0
10	+2.98%	-2.81%
50	+4.37%	-4.02%
100	+4.56%	-4.18%
200	+4.66%	-4.26%
32767	+4.76%	-4.35%

Note: Accuracy of the reception depends on the number of bits for 1 frame, the machine clock, and the reload value. The higher the machine clock and the division ratio become, the higher the accuracy becomes.

■ External Clock

The baud rate generator divides the external clock when "1" is written to EXT bit in the baud rate generator registers 1, 0 (BGR1/BGR0).

Note:

The external clock signal is synchronized with the internal clock at UART. Therefore, the operation becomes unstable when the external clock is unsynchronizable.

■ Functions of the Reload Counter

The reload counter has the transmission and reception reload counters that function as the dedicated baud rate generator. The reload counter consists of 15-bit register and generates a transmission and reception clock by the external or internal clock.

■ Start of a Count

The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).

■ Restart

The reload counters can be restarted for the following conditions:

- For both transmission/reception reload counters

  Programmable reset (SCR: UPCL bit)

- For reception reload counter only

  Detection of the start bit falling edge in asynchronous mode

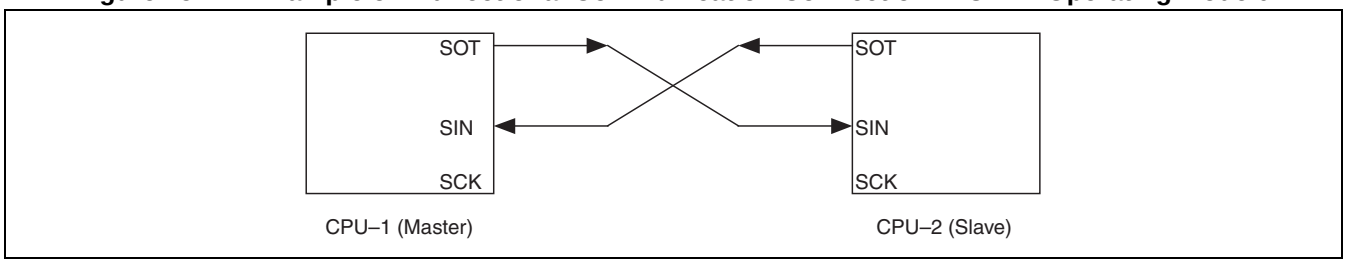
15.7 Setting Procedure and Program Flow for Operating Mode 0 (Asynchronous Normal Mode)

In operating mode 0, asynchronous serial bidirectional communication is available.

■ Inter-CPU Connection

In operating mode 0 (normal mode), the bidirectional communication is selected. As shown in Figure 15.7-1, 2 CPUs are connected each other.

Figure 15.7-1 Example of Bidirectional Communication Connection in UART Operating Mode 0



■ Flowchart

● Without FIFO

Figure 15.7-2 Example of Bidirectional Communication Flowchart (Without FIFO)

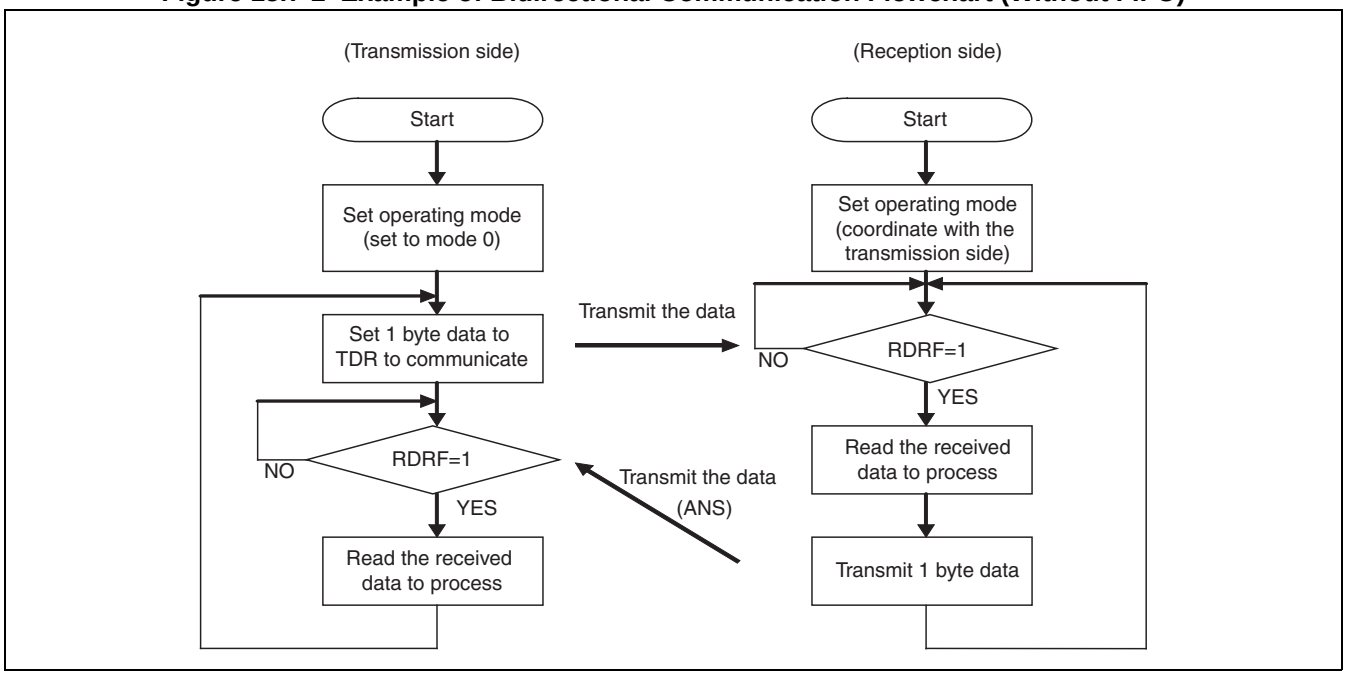
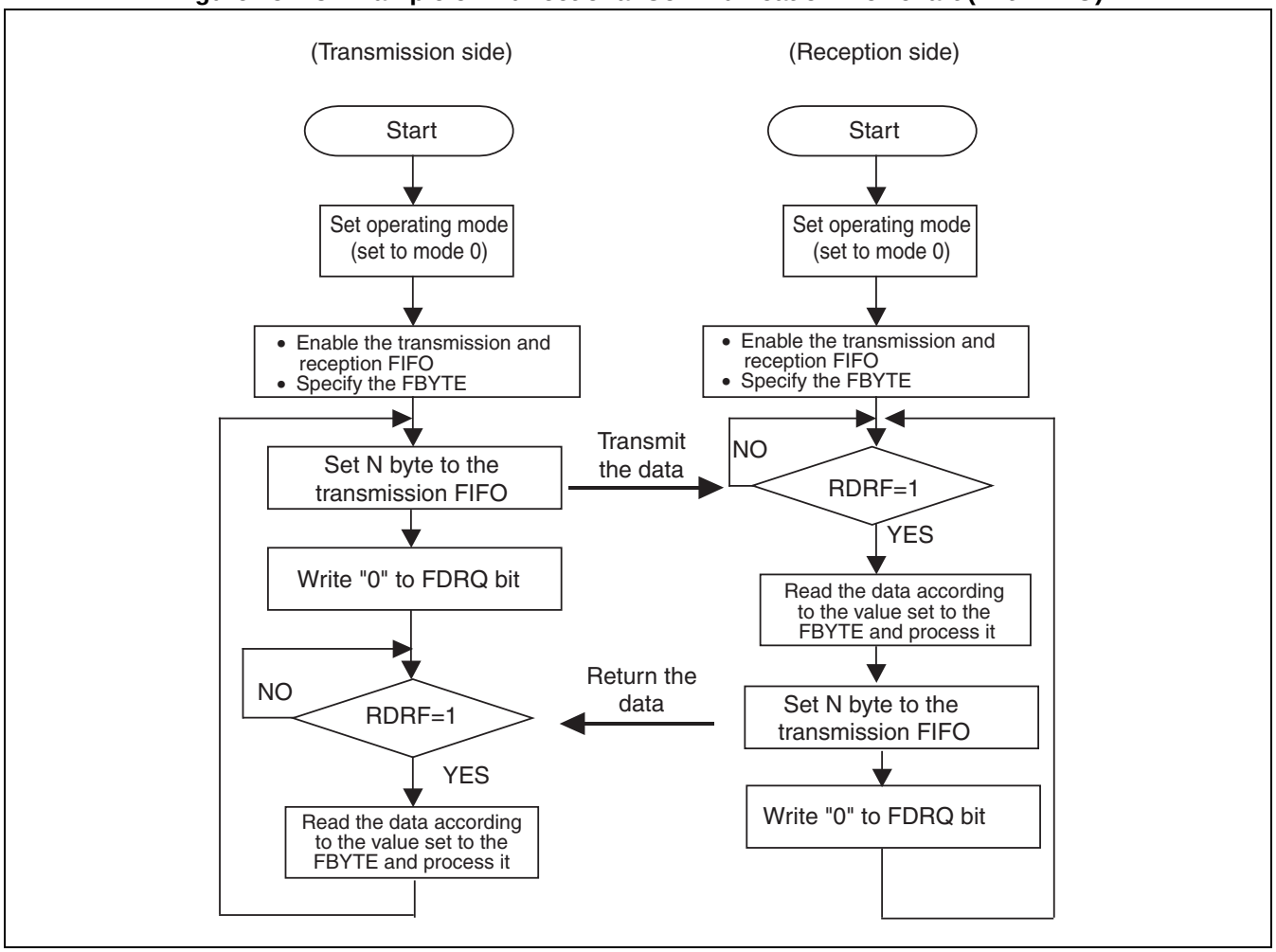


Figure 15.7-3 Example of Bidirectional Communication Flowchart (With FIFO)



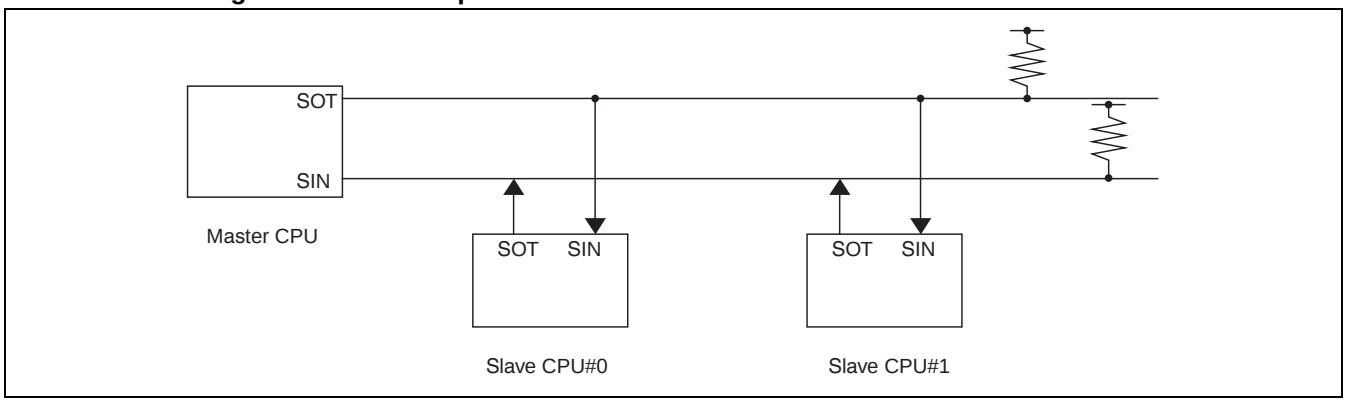
15.8 Setting Procedure and Program Flow for Operating Mode 1 (Asynchronous Multiprocessor Mode)

In operating mode 1 (multiprocessor mode), multiple CPUs communication using the master/slave connection is available. It is possible to use as master/slave.

■ Inter-CPU Connection

As shown in the figure below, the master/slave communication system consists of 1 master CPU and multiple slave CPUs connected by 2 common communication lines. UART can be used for the master or slave.

Figure 15.8-1 Example of UART Master/Slave Communication Connection



■ Function Selection

For master/slave communication, select the operating mode and data transfer method as shown in Table 15.8-1.

Table 15.8-1 Master/Slave Communication Function Selection

	Operating mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission and reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = 1 + 7-bit or 8-bit address	None	1-bit or 2-bit	LSB first or MSB first
Data transmission and reception			AD = 0 + 7-bit or 8-bit data			

Note:

Use word access to perform the transmission and reception data (TDR/RDR) in the operating mode 1.

● Communication procedure

The communication starts when the master CPU transmits address data. The address data, where D8 bit is set to "1", selects the slave CPU to be the destination of the communication. Each slave CPU identifies the address data with the program and communicates with the master CPU (normal data) when the address data matches with the address assigned to the slave CPU.

Figure 15.8-2 and Figure 15.8-3 show the flowcharts for master/slave communication.

■ Flowchart

● Without FIFO

Figure 15.8-2 Example of Master/Slave Communication Flowchart (Without FIFO)

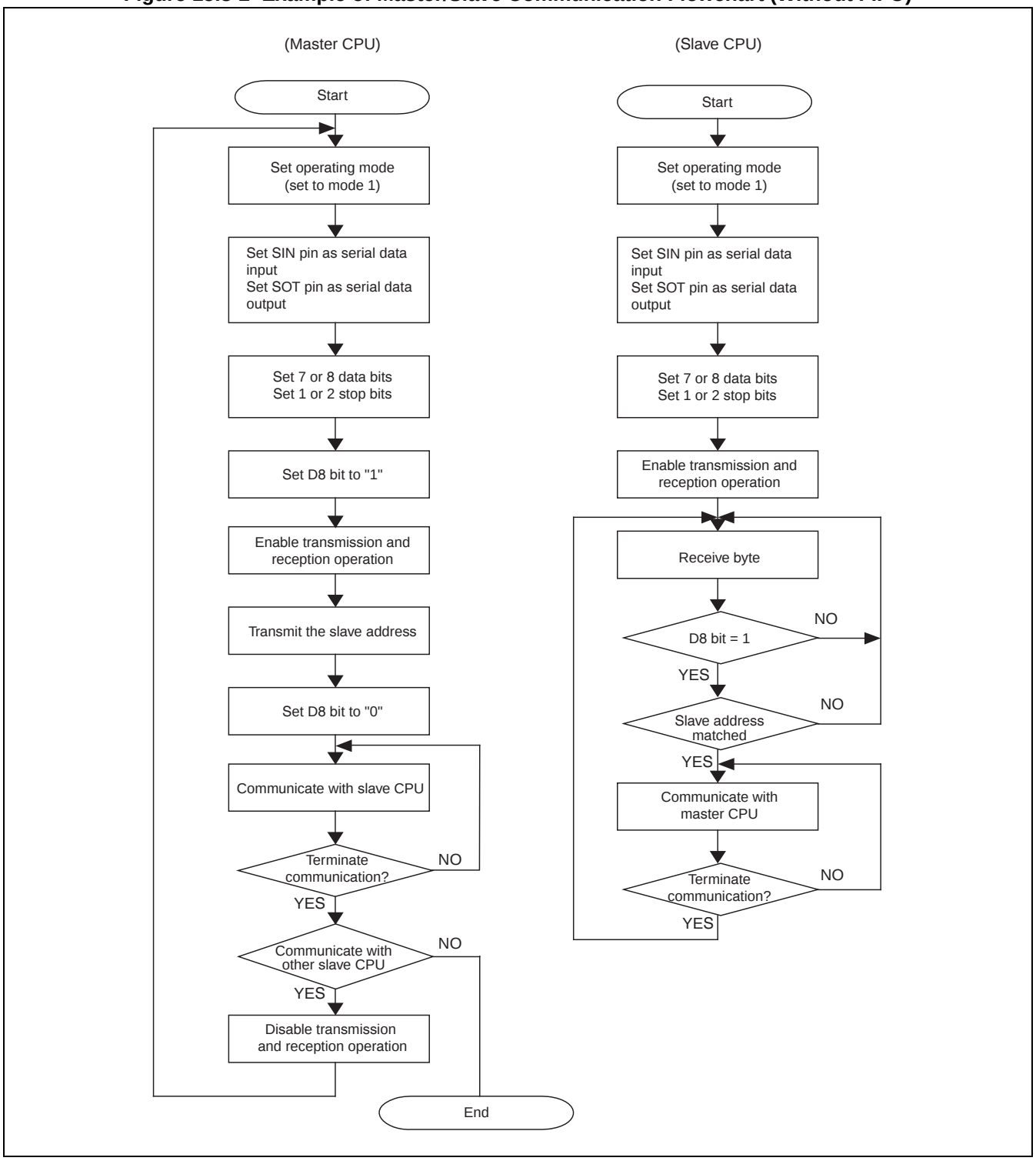
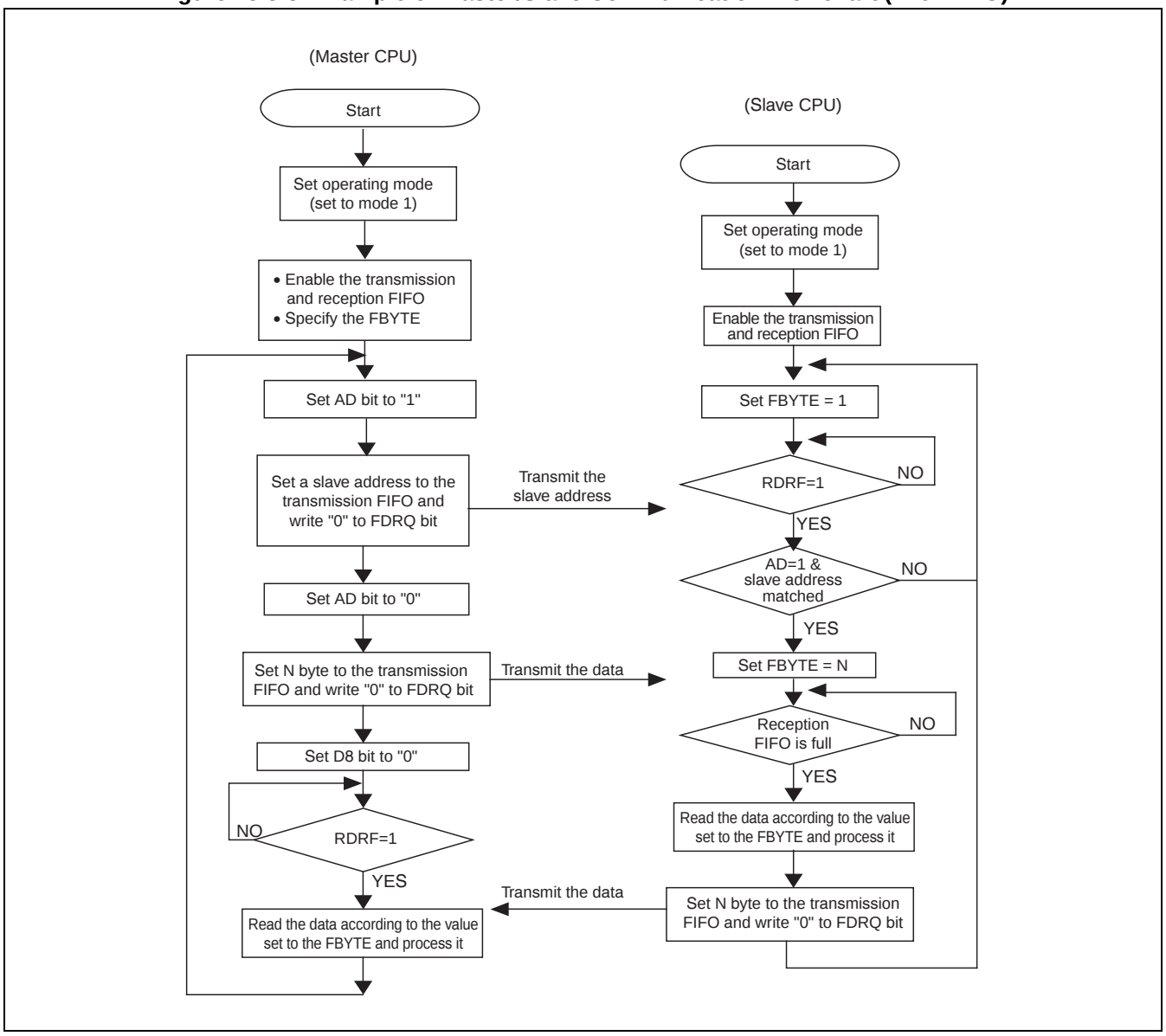


Figure 15.8-3 Example of Master/Slave Communication Flowchart (With FIFO)



15.9 Notes on UART Mode

The notes for when you use the UART mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.

15.10 Overview of CSIO (Clock Synchronous Multi Function Serial Interface)

CSIO (clock synchronous multi function serial interface) is a general-purpose serial data communication interface for synchronous communication with external devices (SPI compliant). In addition, it has the transmission/reception FIFO (Maximum 16-byte each).

■ Functions of CSIO (Clock Synchronous Multi Function Serial Interface)

		Function
1	Data buffer	• Full-duplex, double buffering (when FIFO is not used) • Transmission/reception FIFO (maximum 16-byte each) * (when FIFO is used)
2	Transfer format	• Clock synchronous (no start/stop bit) • Master/slave function • SPI compliant (both master/slave)
3	Baud rate	• Dedicated baud rate generator (15-bit reload counter configuration, when master operation) • External clock input capability (when slave operation)
4	Data length	• Variable 5 to 9 bits
5	Reception error detection	• Overrun error
6	Interrupt request	• Reception interrupt (reception completion, overrun error) • Transmission interrupt (transmission data empty, transmission bus idle) • Transmission FIFO interrupt (when the transmission FIFO is empty) • Both transmission/reception have the extended intelligent I/O service (EI^{2}OS) and DMA transfer function
7	Synchronous mode	• Master or slave function
8	Pin access	• "1" can be set to the serial data output pin
9	FIFO options	• Transmission/reception FIFO are provided (maximum size: transmission FIFO 16-byte, reception FIFO 16-byte)* • Transmission FIFO and reception FIFO can be selected • Transmission data can be retransmitted • Reception FIFO interrupt timing can be changed from the software • Independent FIFO reset support

*: FIFO is only provided to ch.0, ch.1 and ch.2 (16-byte each for transmission and reception).

15.11 Registers of CSIO (Clock Synchronous Multi Function Serial Interface)

This section shows the register list of CSIO (clock synchronous multi function serial interface).

■ Register List of CSIO (Clock Synchronous Multi Function Serial Interface)

Table 15.11-1 Register List of CSIO (Clock Synchronous Multi Function Serial Interface)

	Address		bit15 bit8	bit7 bit0
CSIO	$00X0_H$	$00X1_H$	SCR (serial control register)	SMR (serial mode register)
	$00X2_H$	$00X3_H$	SSR (serial status register)	ESCR (extended communication control register)
	$00X4_H$	$00X5_H$	RDR1/TDR1 (transmission and reception data register 1)	RDR0/TDR0 (transmission and reception data register 0)
	$00X6_H$	$00X7_H$	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	$00X8_H$	$00X9_H$	-	-
FIFO	$00YA_H$	$00YB_H$	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)
	$00YC_H$	$00YD_H$	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)

(X = 06_H, 07_H, 08_H, 09_H, $0A_H$, $0B_H$, $1B_H$, $1C_H$, $1D_H$, $1E_H$, $1F_H$, Y=06_H, 07_H, 08_H)

Table 15.11-2 Bit Arrangement of CSIO (Clock Synchronous Multi Function Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
TDR/RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

15.11.1 Serial Control Register (SCR)

Serial control register (SCR) can enable/disable the transmission and reception interrupts, the transmission and reception operations, and transmission idle interrupt. It is also able to set to connect to SPI and reset CSIO.

■ Serial Control Register (SCR)

Figure 15.11-1 shows the bit configuration of the serial control register (SCR), and Table 15.11-3 shows the function of each bit.

Figure 15.11-1 Bit Configuration of Serial Control Register (SCR)

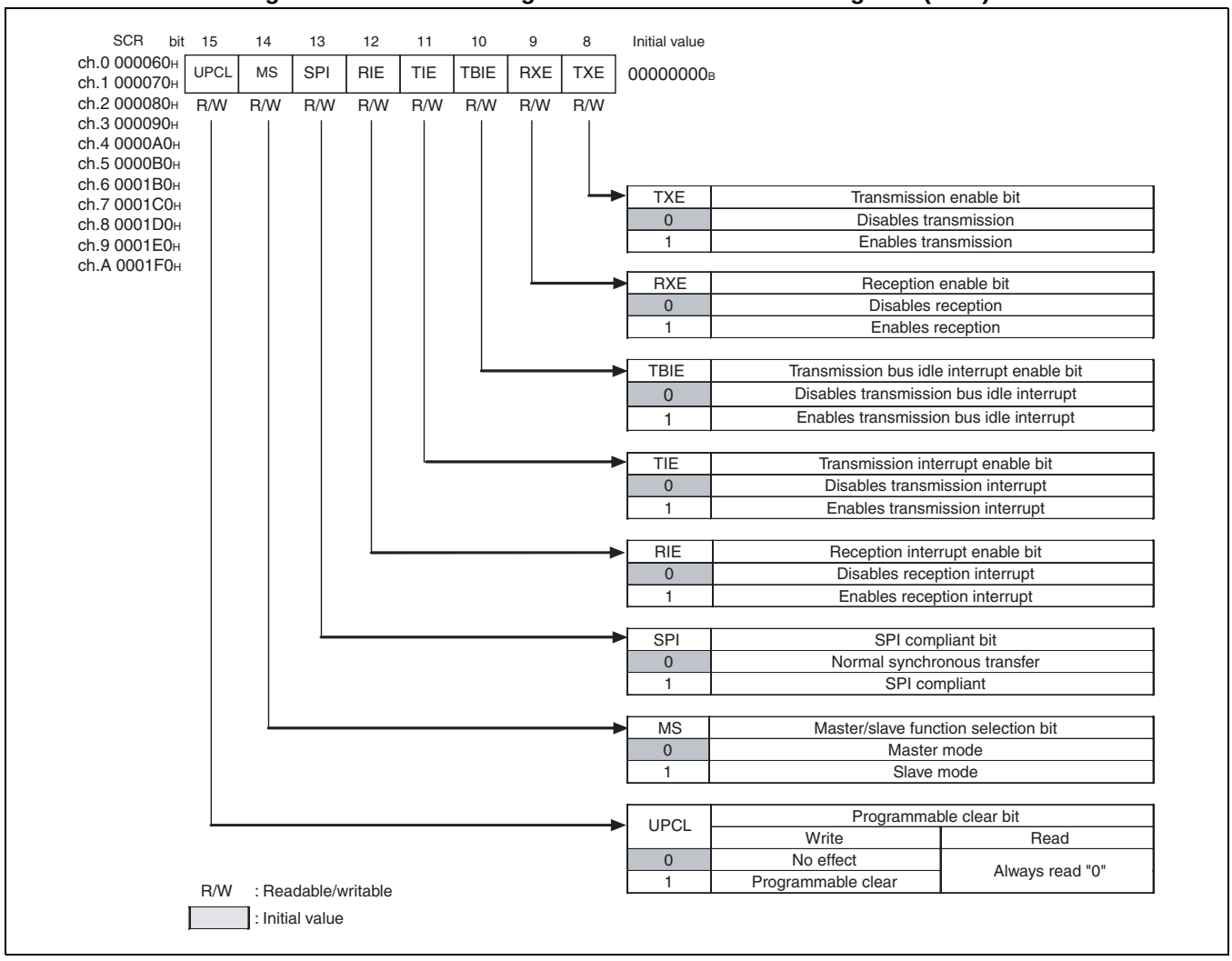


Table 15.11-3 Function Description of Each Bit in the Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	Initializes the internal state of CSIO. When "1" is set: • CSIO is directly reset (software reset). However, the register setting is retained. The CSIO that is under transmission/reception status is immediately disconnected. • Baud rate generator reloads the value set in BGR1/BGR0 register and restarts. • All transmission and reception interrupt sources (TDRE, TBI, RDRF, and ORE) are initialized. • When "0" is set: No effect on the operation. When reading, "0" is always read. Notes: • Execute programmable clear after you disable an interrupt. • When using FIFO, disable FIFO (FE2, FE1 = 0) before you execute programmable clear.
bit14	MS: Master/slave function selection bit	Specifies whether to use master or slave mode. When "0" is set: Sets to the master mode. When "1" is set: Sets to the slave mode. Note: If the slave mode is specified, an external clock is directly input when SCKE = 0 in the serial mode register (SMR).
bit13	SPI: SPI compliant bit	This bit is provided for the SPI compliant communication. When "0" is set: Performs the normal synchronous communication. When "1" is set: Performs the SPI compliant communication.
bit12	RIE: Reception interrupt enable bit	• Enables/disables the output of a reception interrupt request to the CPU. • A reception interrupt request is output when RIE bit and the reception data flag bit (RDRF) are "0" or when the error flag bit (ORE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	• Enables/disables the output of a transmission interrupt request to the CPU. • A transmission interrupt request is output when TIE bit and the TDRE bit are "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	• Enables/disables the output of a transmission bus idle interrupt request to the CPU. • A transmission bus idle interrupt request is output when TBI bit and the TBIE bit are "1".
bit9	RXE: Reception operation enable bit	Enables/disables the reception operation of CSIO. • When "0" is set: Disables the data frame reception operation. • When "1" is set: Enables the data frame reception operation. Note: The reception operation is immediately stopped if you disable it (RXE = 0) while receiving.
bit8	TXE: Transmission operation enable bit	Enables/disables the transmission operation of CSIO. • When "0" is set: Disables the data frame transmission operation. • When "1" is set: Enables the data frame transmission operation. Note: The transmission operation is immediately stopped if you disable it (TXE = 0) while transmitting.

15.11.2 Serial Mode Register (SMR)

Serial mode register (SMR) can set the operating mode, select transfer direction, data length and serial clock inversion, and enable/disable the output to the pin of the serial data and clock.

■ Serial Mode Register (SMR)

Figure 15.11-2 shows the bit configuration of the serial mode register (SMR), and Table 15.11-4 shows the function of each bit.

Figure 15.11-2 Bit Configuration of Serial Mode Register (SMR)

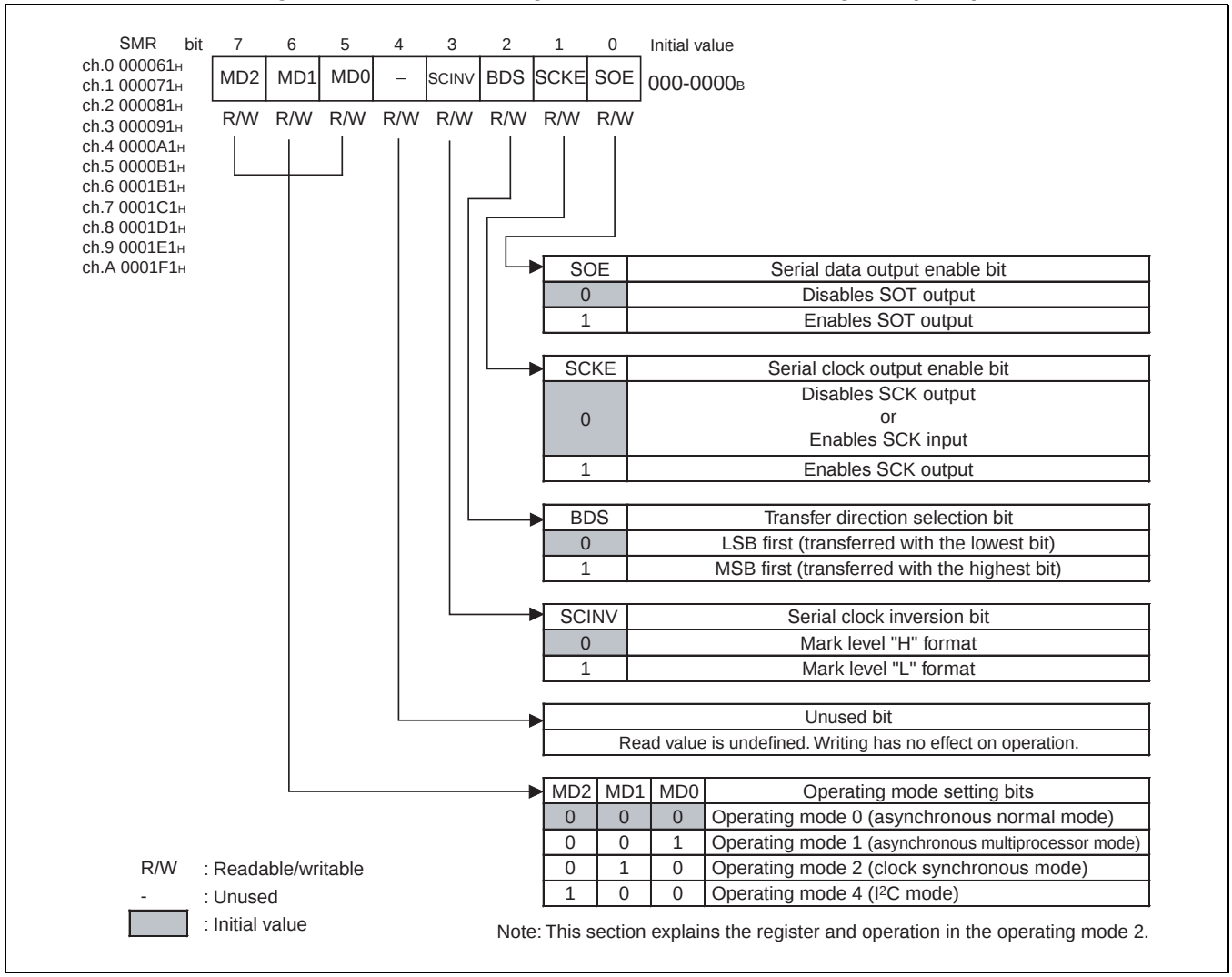


Table 15.11-4 Function Description of Each Bit in the Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operating mode setting bits	Sets the operating mode. "000_B": Sets to the operating mode 0 (asynchronous normal mode) "001_B": Sets to the operating mode 1 (asynchronous multiprocessor mode) "010_B": Sets to the operating mode 2 (clock synchronous mode) "100_B": Sets to the operating mode 4 (I^2C mode) Notes: • Any setting other than those above is disabled. • When switching the operating mode, execute the programmable clear (SCR:UPCL = 1) before switching it. • Set the operating mode, and then set each register.
bit4	Unused bit	When reading : Value is undefined. When writing : No effect.
bit3	SCINV: Serial clock inversion bit	Inverts the serial clock format. When "0" is set: • The mark level of the serial clock output is set to "H". • The transmission data is synchronized with the falling edge of a serial clock in normal transfer or the rising edge of a serial clock in SPI transfer to output. • The reception data is sampled at the rising edge of a serial clock in normal transfer or the falling edge of a serial clock in SPI transfer. When "1" is set: • The mark level of the serial clock output is set to "L". • The transmission data is synchronized with the rising edge of a serial clock in normal transfer or the falling edge of a serial clock in SPI transfer to output. • The reception data is sampled at the falling edge of a serial clock in normal transfer or the rising edge of a serial clock in SPI transfer. Note: Specify this bit when the transmission/reception are disabled (TXE = RXE= 0).
bit2	BDS: Transfer direction selection bit	Specifies whether the transfer serial data is transferred with the lowest bit (LSB first, BDS = 0) or the highest bit (MSB first, BDS = 1). Note: Specify this bit when the transmission/reception are disabled (TXE = RXE= 0).
bit1	SCKE: Serial clock output enable bit	Controls the I/O port of the serial clock. When "0" is set: Enables SCK "H" output or SCK input. When using as SCK input, set the general-purpose I/O port as an input port. When "1" is set: Enables SCK output.
bit0	SOE: Serial data output enable bit	Enables/disables the output of the serial data. When "0" is set: Enables SOT "H" output. When "1" is set: Enables SOT output.

Note:

Set the operating mode first because the other registers will be initialized once the operating mode has been changed. However, when SCR and SMR are written at the same time by 16-bit writing, the written contents will be reflected on SCR.

15.11.3 Serial Status Register (SSR)

Serial status register (SSR) verifies the transmission/reception status and the reception error flag, and also clears the reception error flag.

■ Serial Status Register (SSR)

Figure 15.11-3 shows the bit configuration of the serial status register (SSR), and Table 15.11-5 shows the function of each bit.

Figure 15.11-3 Bit Configuration of Serial Status Register (SSR)

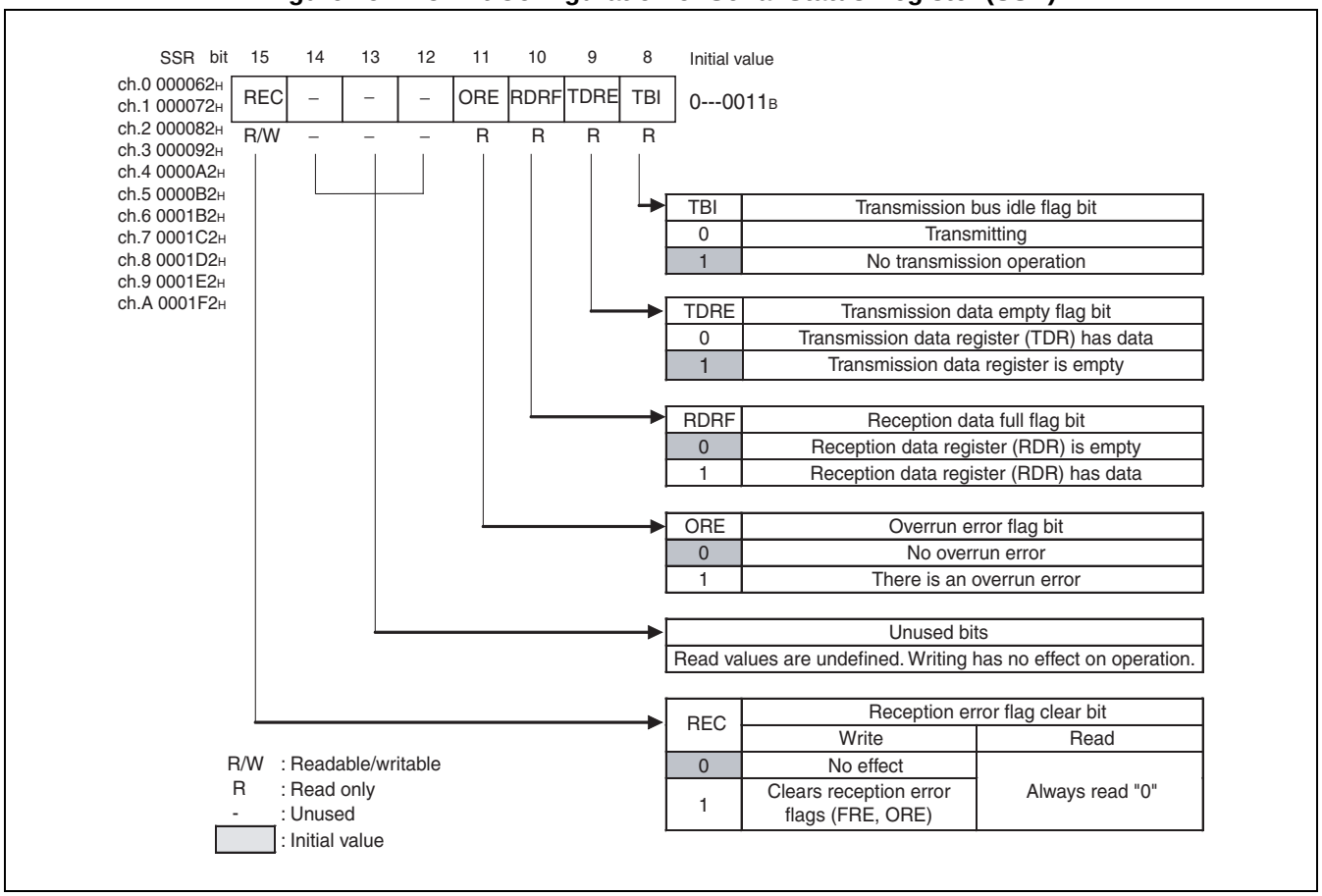


Table 15.11-5 Function Description of Each Bit in the Serial Status Register (SSR)

Bit name		Function
bit15	REC: Reception error flag clear bit	Clears ORE flag in the serial status register (SSR). • Writing "1" to this bit clears the error flags. • Writing "0" has no effect. When reading, "0" is always read.
bit14 to bit12	Unused bits	When reading: Values are undefined. When writing: No effect.
bit11	ORE: Overrun error flag bit	• When an overrun error occurs while receiving, this bit is set to "1". When you write "1" to REC bit in the serial status register (SSR), this bit is cleared. • A reception interrupt request is output when ORE bit and RIE bit are "1". • Data in the reception data register (RDR) is invalid if this flag is set. • When this bit is set while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the reception data will not be stored into the reception FIFO.
bit10	RDRF: Reception data full flag bit	• Indicates the status of the reception data register (RDR). • When the reception data is loaded to RDR, this bit is set to "1". If the reception data register (RDR) is read, this bit is cleared to "0". • A reception interrupt request is output when RDRF bit and RIE bit are "1". • When using the reception FIFO, RDRF is set to "1" if the reception FIFO has received a predefined number of data. • When using the reception FIFO, if the reception idle state continues over 8 clocks of the baud rate clock (because the reception FIFO has not received a predefined number of data and some data still remains in the reception FIFO), RDRF is set to "1". If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. • When using the reception FIFO, this bit is cleared to "0" if the reception FIFO gets empty.
bit9	TDRE: Transmission data empty flag bit	• Indicates the status of the transmission data register (TDR). • When you write a transmission data to TDR, this bit becomes "0" to indicate that there is some valid data. When the data is loaded to the transmission shift register to start transmission, this bit becomes "1" to indicate that there is no valid data in TDR. • A transmission interrupt request is output when TIE bit and the TDRE bit are "1". • If you set UPCL bit in the serial control register (SCR) to "1", TDRE bit becomes "1". • For more information about set/reset timing of TDRE bit when using the transmission FIFO, see "15.4.4 Interrupt Generation and Flag Set Timing When Using Transmission FIFO".
bit8	TBI: Transmission bus idle flag bit	• Indicates that CSIO is not processing the transmission operation. • This bit becomes "0" when data is written to the transmission data register (TDR). • This bit becomes "1" when the transmission data register (TDR) is empty (TDRE = 1) and the transmission operation is not in progress. • If you set UPCL bit in the serial control register (SCR) to "1", TDRE bit becomes "1". • A transmission interrupt request is output when this bit is "1" and the transmission bus idle interrupt is enabled (SCR:TBIE = 1).

15.11.4 Extended Communication Control Register (ESCR)

Extended communication control register (ESCR) can set the transmission/reception data length and fix the serial output to "H".

■ Bit Configuration of Extended Communication Control Register (ESCR)

Figure 15.11-4 shows the bit configuration of the extended communication control register (ESCR), and Table 15.11-6 shows the function of each bit.

Figure 15.11-4 Bit Configuration of Extended Communication Control Register (ESCR)

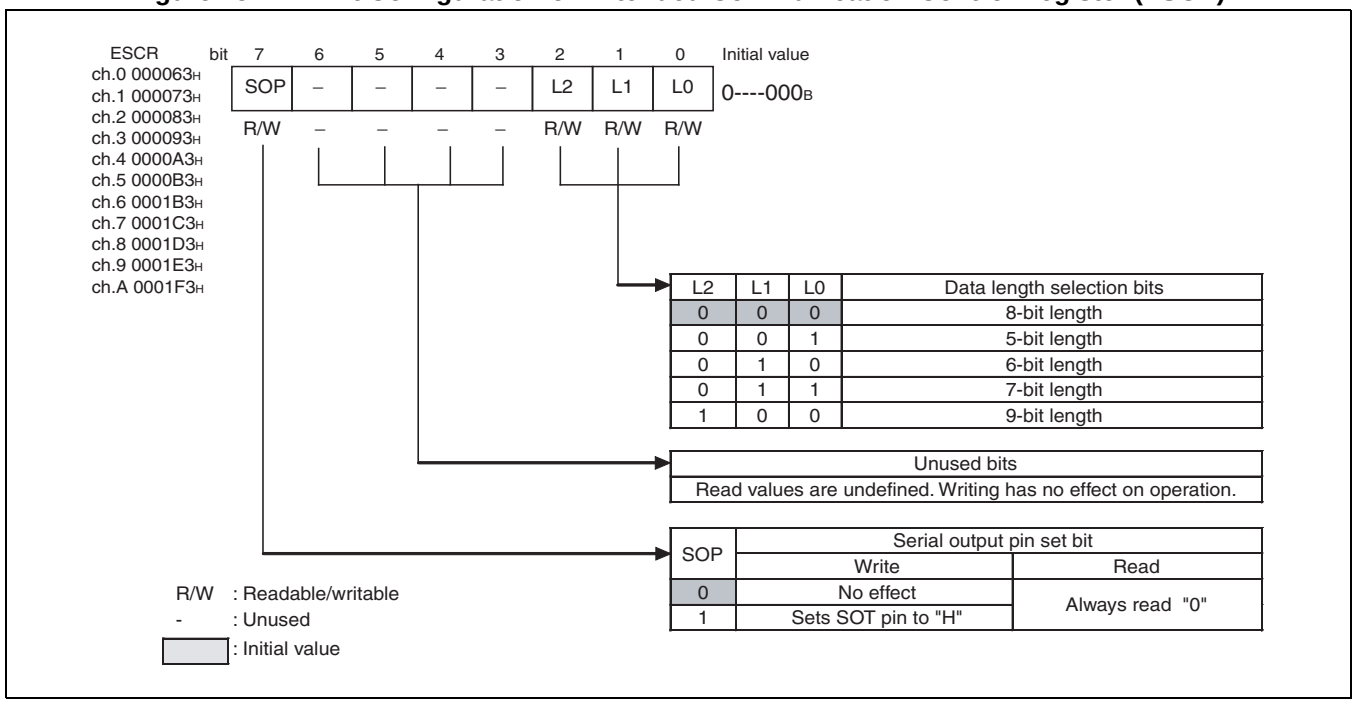


Table 15.11-6 Function Description of Each Bit in the Extended Communication Control Register (ESCR)

Bit name		Function
bit7	SOP: Serial output pin set bit	• Sets the serial output pin to "H" level. • When reading, "0" is always read. Note: Do not set this bit while transmitting the serial data. Configuration value for this bit is enabled only when TXE bit for serial control register (SCR) is "0".
bit6 to bit3	Unused bits	When reading : Values are undefined. When writing : No effect.
bit2 to bit0	L2 to L0: Data length selection bits	Specifies the data length of the transmission/reception data. • When "000_B" is set: Sets the data length to 8-bit. • When "001_B" is set: Sets the data length to 5-bit. • When "010_B" is set: Sets the data length to 6-bit. • When "011_B" is set: Sets the data length to 7-bit. • When "100_B" is set: Sets the data length to 9-bit. Note: Any setting other than those above is disabled.

15.11.5 Reception Data Register/Transmission Data Register (RDR/TDR)

Reception/transmission data registers are arranged in the same address. When reading, it functions as the reception data register and when writing, it functions as the transmission data register.

■ Reception Data Register (RDR)

Figure 15.11-5 shows the bit configuration of the reception data register (RDR).

Figure 15.11-5 Bit Configuration of the Reception Data Register (RDR)

RDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	8	7	6	5	4	3	2	1	0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	000000000_B
	R	R	R	R	R	R	R	R	R	

R: Read only

Reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal transmitted to the serial input pin (SIN pin) is converted at the shift register and then stored in this reception data register (RDR).
- As described below, data becomes "0" from the upper bits depending on the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9-bit	X	X	X	X	X	X	X	X	X
8-bit	0	X	X	X	X	X	X	X	X
7-bit	0	0	X	X	X	X	X	X	X
6-bit	0	0	0	X	X	X	X	X	X
5-bit	0	0	0	0	X	X	X	X	X

- When the received data is stored in the reception data register (RDR), the reception data full flag bit (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SSR:RIE = 1), a reception interrupt request occurs.
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". If the serial reception data register (RDR) is read, the reception data full flag bit (SSR:RDRF) is automatically cleared to "0".
- If a reception error occurs (SSR:ORE), the data in the reception data register (RDR) becomes invalid.
- For 9-bit length transfer, RDR reading is performed by 16-bit access.

Notes:

- When using the reception FIFO, RDRF is set to "1" if the reception FIFO has received a predefined number of data.
- When using the reception FIFO, RDRF is cleared to "0" if the reception FIFO gets empty.
- When a reception error occurs (SSR:ORE = 1) while using the reception FIFO, the enable bit in the reception FIFO will be cleared and the received data will not be stored into the reception FIFO.

■ Transmission Data Register (TDR)

Figure 15.11-6 shows the bit configuration of the transmission data register (TDR).

Figure 15.11-6 Bit Configuration of the Transmission Data Register (TDR)

TDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	8	7	6	4	5	3	2	1	0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	111111111_B
	W	W	W	W	W	W	W	W	W	

W: Write only

Transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If data to be transmitted is written to the transmission data register (TDR) when the transmission operation is enabled (SCR:TXE = 1), the data is transferred to the transmission shift register where the data is converted to serial data, and then transmitted from the serial data output pin (SOT pin).
- As described below, data becomes invalid from the upper bits depending on the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9-bit	X	X	X	X	X	X	X	X	X
8-bit	Invalid	X	X	X	X	X	X	X	X
7-bit	Invalid	Invalid	X	X	X	X	X	X	X
6-bit	Invalid	Invalid	Invalid	X	X	X	X	X	X
5-bit	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- A transmission data empty flag (SSR:TDRE) is cleared to "0" when the transmission data is written into the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) is set to "1" when the transmission data is transferred to the transmission shift register and the transmission is started.
- When the transmission data empty flag (SSR:TDRE) is "1", you can write next transmission data. If transmission interrupt is enabled, a transmission interrupt request occurs. Write next transmission data when a transmission interrupt occurs or when the transmission data empty flag (SSR:TDRE) is "1".
- You cannot write a transmission data to the transmission data register (TDR) when the transmission data empty flag (SSR:TDRE) is "0" and also the transmission FIFO is disabled or full.
- For 9-bit length transfer, writing to TDR is performed by 16-bit access.

Notes:

- The transmission data register (TDR) is a write only register and the reception data register (RDR) is a read only register. These registers are located at the same address, so the read value is different from the write value. Therefore an instruction that operates read-modify-write (RMW) instruction, such as INC/DEC, cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR:TDRE) when using the transmission FIFO, see "15.4.4 Interrupt Generation and Flag Set Timing When Using Transmission FIFO".

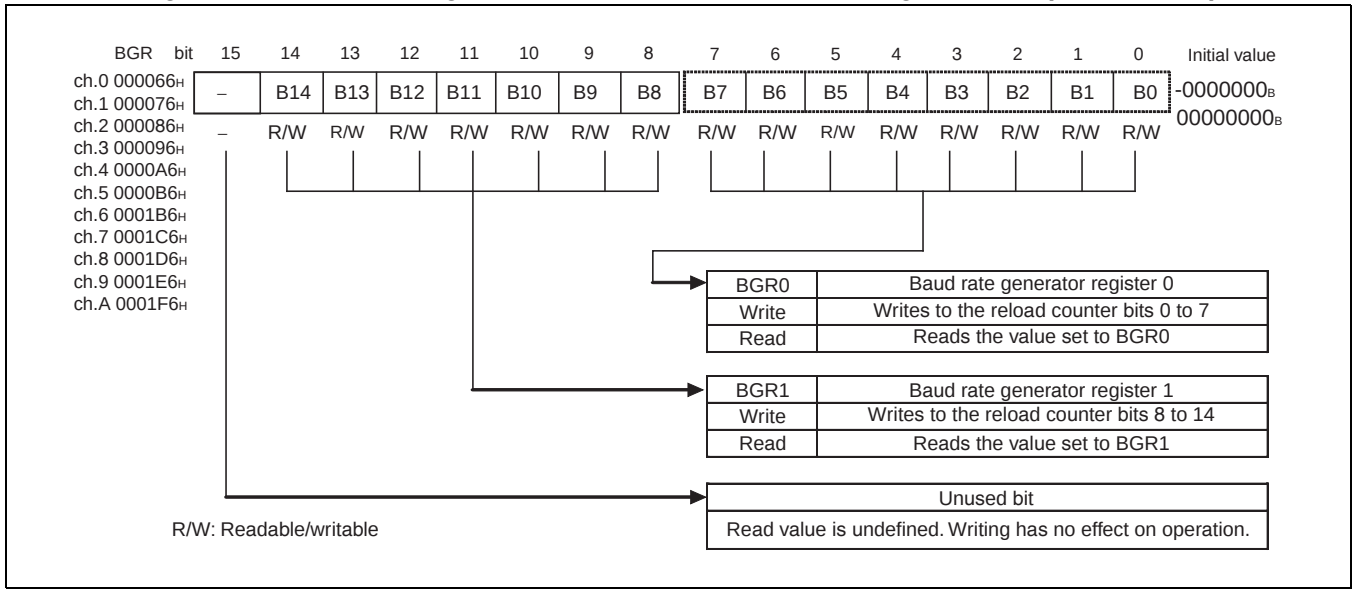
15.11.6 Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

The baud rate generator registers 1, 0 (BGR1/BGR0) sets the division ratio for the serial clock.

■ Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

Figure 15.11-7 shows the bit configuration of the baud rate generator registers 1, 0 (BGR1/BGR0).

Figure 15.11-7 Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)



- Set a value to the baud rate generator registers 1, 0 (BGR1, BGR0).
- BGR1 (supports the upper bits) and BGR0 (supports the lower bits) can write the reload value to count and read the value set to BGR1/BGR0.
- The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).

Notes:

- Write to the baud rate generator registers 1, 0 (BGR1/BGR0) by 16-bit access.
- If the reload value is an even number, "L" and "H" widths of the serial clock become as follows depending on the setting for SCINV bit: If it is an odd number, "H" and "L" widths of the serial clock have the same length.

  When SCINV = 0, "H" width of the serial clock is longer than "L" width by 1 cycle of the machine clock.

  When SCINV = 1, "L" width of the serial clock is longer than "H" width by 1 cycle of the machine clock.
- Be sure to set the reload value to 1 or more. However, if you use CSIO for the master and slave, set the reload value for the master CSIO to 3 or more.
- When you change the value set to the baud rate generator registers 1, 0 (BGR0/BGR1), new value is reloaded after the counter value reaches "0000000000000000_B". Therefore, if you want to reflect the change immediately, execute CSIO reset (UPCL) after you change the value set to BGR1/BGR0.
- When using the reception FIFO, you need to set a baud rate to BGR1/BGR0 if the reception FIFO idle detection enable bit is set to "1" (FCR1:FRIIE) to operate as the slave mode.

15.11.7 FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the test for FIFO, specifies the transmission/ reception FIFO, enables the transmission FIFO interrupt, and controls the interrupt flag.

■ Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 15.11-8 shows the bit configuration of the FIFO control register 1 (FCR1), and Table 15.11-7 shows the function of each bit.

Figure 15.11-8 Bit Configuration of FIFO Control Register 1 (FCR1)

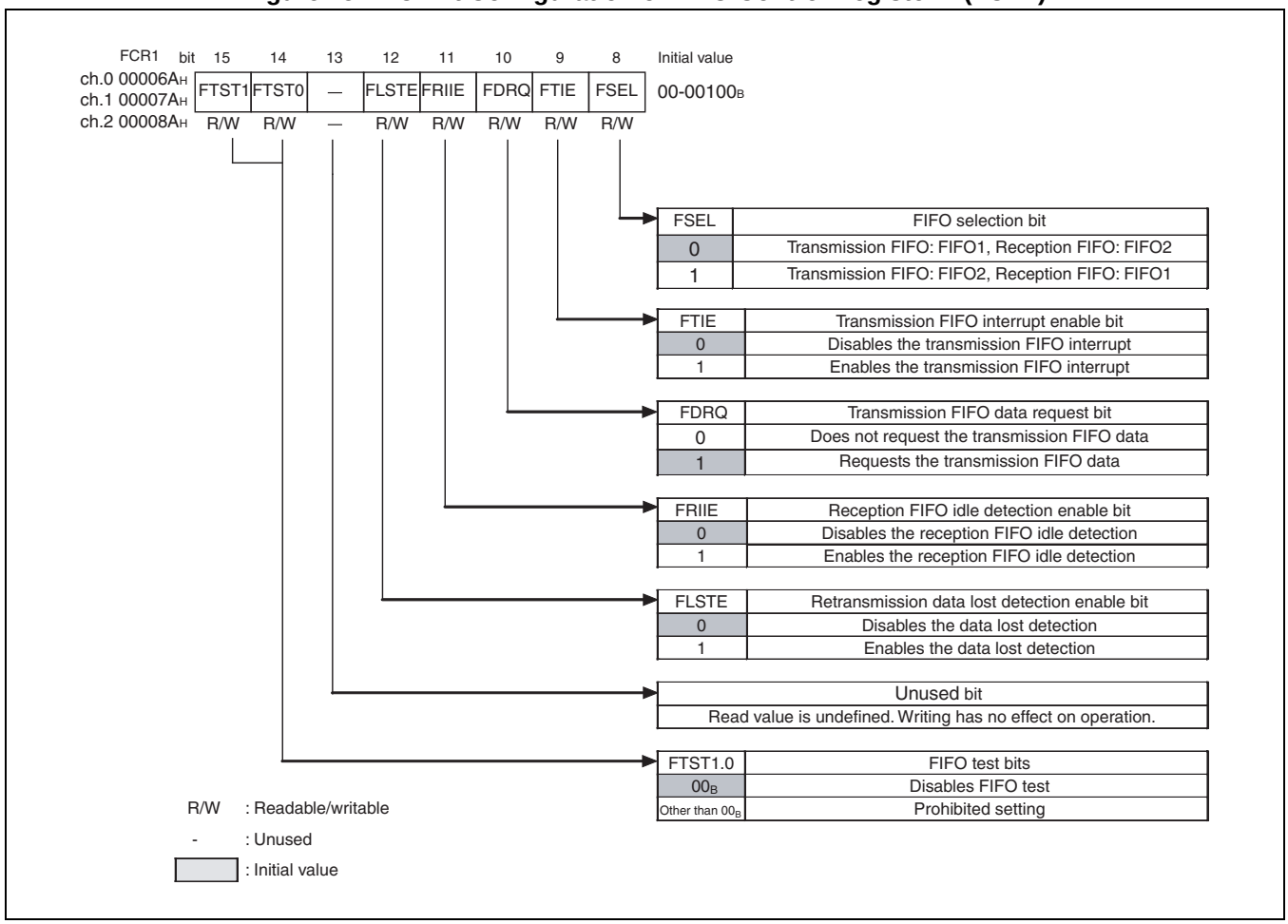


Table 15.11-7 Function Description of Each Bit in the FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	FTST1, FTST0: FIFO test bits	These are the test bits for FIFO. Be sure to set these bits to "0". Note: If you set these bits to "1", the test for FIFO will be executed.
bit13	Unused bit	When reading : Value is undefined. When writing : No effect.
bit12	FLSTE: Retransmission data lost detection enable bit	Enables data lost detection. When "0" is set: Disables data lost detection. When "1" is set: Enables data lost detection. Note: When you set this bit to "1", set FSET bit to "1" before doing so.
bit11	FRIIE: Reception FIFO idle detection enable bit	Specifies whether to detect the reception idle state that continues over 8-bit time with the reception FIFO holding valid data. If the reception interrupt is enabled (SCR:RIE = 1), a reception interrupt occurs when the reception idle state is detected. When "0" is set: Disables the reception idle state detection. When "1" is set: Enables the reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a data request bit for the transmission FIFO. When this bit is set to "1", it indicates that the transmission data is being requested. A transmission FIFO interrupt request is output when this bit is "1" and the transmission FIFO interrupt is enabled (FTIE = 1). FDRQ set condition • FBYTE (for transmission) = 0 (transmission FIFO is empty) • Reset the transmission FIFO FDRQ reset condition • When writing "0" to this bit • When the transmission FIFO gets full Notes: • When FBYTE (for transmission) = 0, writing "0" to this bit is disabled. • When this bit is set to "0", change of FSEL bit is disabled. • Setting "1" to this bit has no effect on the operations. • "1" is read by a read-modify-write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is an interrupt enable bit for the transmission FIFO. If this bit is set to "1", an interrupt occurs when FDRQ bit is "1".
bit8	FSEL: FIFO selection bit	Selects the transmission/reception FIFO. When "0" is set:Assigns the transmission FIFO:FIFO1 and the reception FIFO:FIFO2. When "1" is set:Assigns the transmission FIFO:FIFO2 and the reception FIFO:FIFO1. Notes: • This bit cannot be cleared by the FIFO reset (FCL2, FCL1 = 1). • When you want to change this bit, disable FIFO operation (FE2, FE1 = 0) and any transmission/reception (TXE = RXE = 0) first.

15.11.8 FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, performs FIFO reset, saves the read pointer, and sets the retransmission.

■ Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 15.11-9 shows the bit configuration of the FIFO control register 0 (FCR0), and Table 15.11-8 shows the function of each bit.

Figure 15.11-9 Bit Configuration of FIFO Control Register 0 (FCR0)

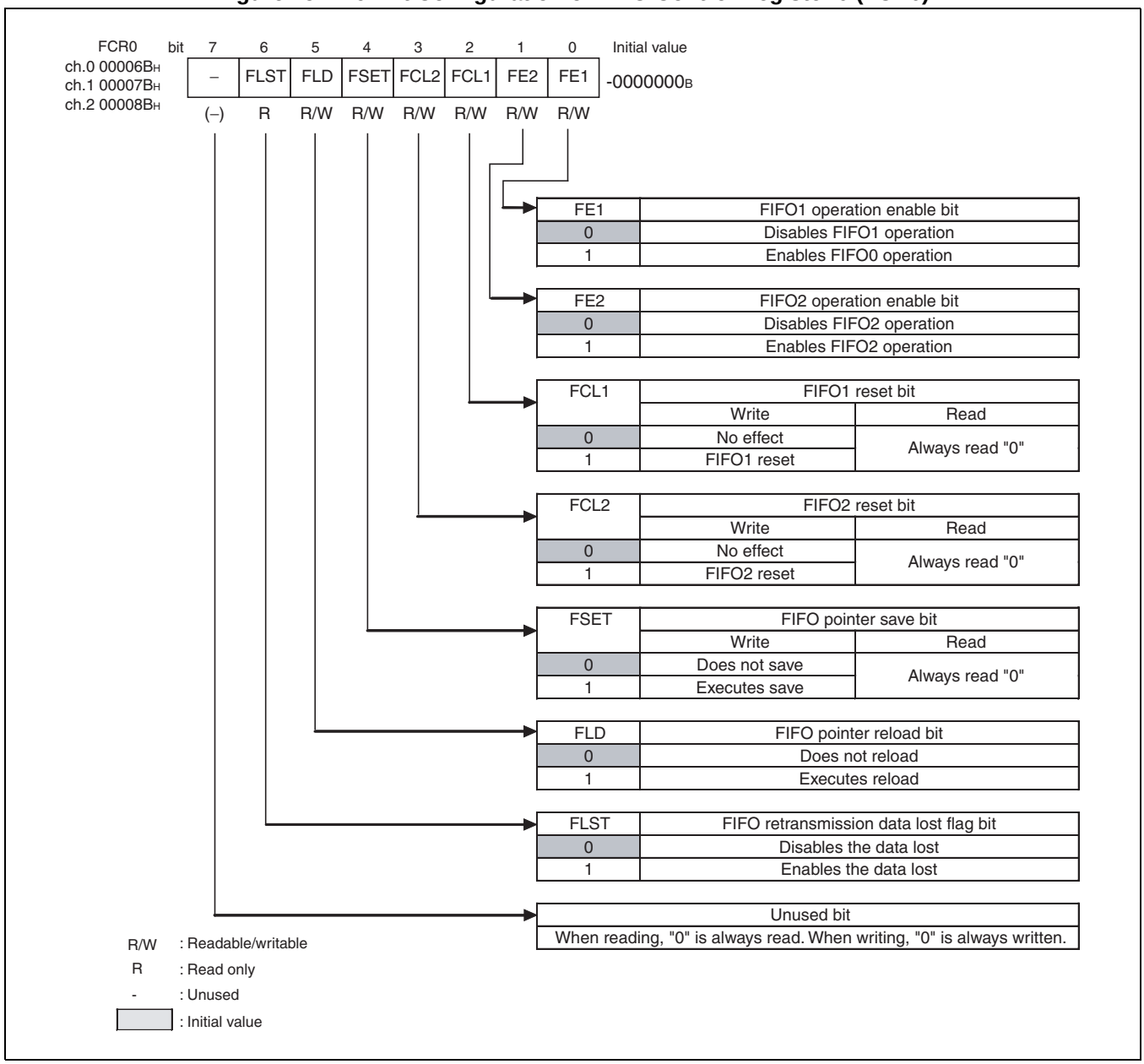


Table 15.11-8 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Unused bit	When reading : "0" is always read. When writing : Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	Indicates that retransmission data of the transmission FIFO has been lost. FLST set condition • Writing to FIFO when FLSTE bit in the FIFO control register 1 (FCR1) is "1" and also the write pointer of the transmission FIFO and the read pointer saved by FSET bit match each other. FLST reset condition • FIFO reset (writing "1" into FCL) • When writing "1" to FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer saved with FSET bit, therefore FLD bit cannot set the retransmission when an error occurs. If you retransmit with this bit set to "1", perform FIFO reset and write the data again into FIFO.
bit5	FLD: FIFO pointer reload bit	Reloads the data saved to the transmission FIFO by FSET bit to the read pointer. This bit is used for the retransmission due to a communication error. When a retransmission setting is completed, this bit becomes "0". Notes: • As long as this bit is set to "1", this bit is reloading to the read pointer. Therefore, do not write anything other than FIFO reset. • Setting this bit to "1" is disabled while transmitting or being in FIFO enabled state. • Set TIE and TBIE bits to "0" and then write "1" into this bit. After you enabled the transmission FIFO, set TIE and TBIE bits to "1".
bit4	FSET: FIFO pointer save bit	Saves the read pointer of the transmission FIFO. Once you save the read pointer before transmission, when FLST bit is "0" it is possible to retransmit in the case a communication error. When "1" is set: Retains the current value set to the read pointer. When "0" is set: No effect. Note: Set this bit to "1" when the number of bytes for transmission (FBYTE) indicates "0".
bit3	FCL2: FIFO2 reset bit	Resets FIFO2. If you set this bit to "1", the internal state of FIFO2 is initialized. Only the FLST2 bit in the FIFO control register 1 (FCR1) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables any transmission/reception before performing FIFO2 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE2 register becomes "0".
bit2	FCL1: FIFO1 reset bit	Resets FIFO1. If you set this bit to "1", the internal state of FIFO1 is initialized. Only the FLST1 bit in the FIFO control register 1 (FCR1) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables any transmission/reception before performing FIFO1 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE1 register becomes "0".

Table 15.11-8 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit1	FE2: FIFO2 operation enable bit	Enables/disables the operation of FIFO2. • When using FIFO2, set this bit to "1". • Set FIFO2 to the transmission FIFO (FCR1:FSEL = 1). When there is some data in FIFO2 and UART is enabled for transmission (TXE = 1), writing "1" into this bit starts transmitting immediately. At this time, set TIE and TBIE bits to "0" and then write "1" into this bit to make TIE and TBIE bits to "1". • If specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • When using as the transmission FIFO, set "1" or "0" to this bit when the transmission buffer is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception buffer is empty (RDRF = 0). • Even if FIFO2 is disabled, the state of FIFO2 is still retained.
bit0	FE1: FIFO1 operation enable bit	Enables/disables the operation of FIFO1. • When using FIFO1, set this bit to "1". • Set FIFO1 to the transmission FIFO (FCR1:FSEL = 0). When there is some data in FIFO1 and UART is enabled for transmission (TXE = 1), writing "1" into this bit starts transmitting immediately. At this time, set TIE and TBIE bits to "0" and then write "1" into this bit to make TIE and TBIE bits to "1". • If specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • When using as the transmission FIFO, set "1" or "0" to this bit when the transmission buffer is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception buffer is empty (RDRF = 0). • Even if FIFO1 is disabled, the state of FIFO1 is still retained.

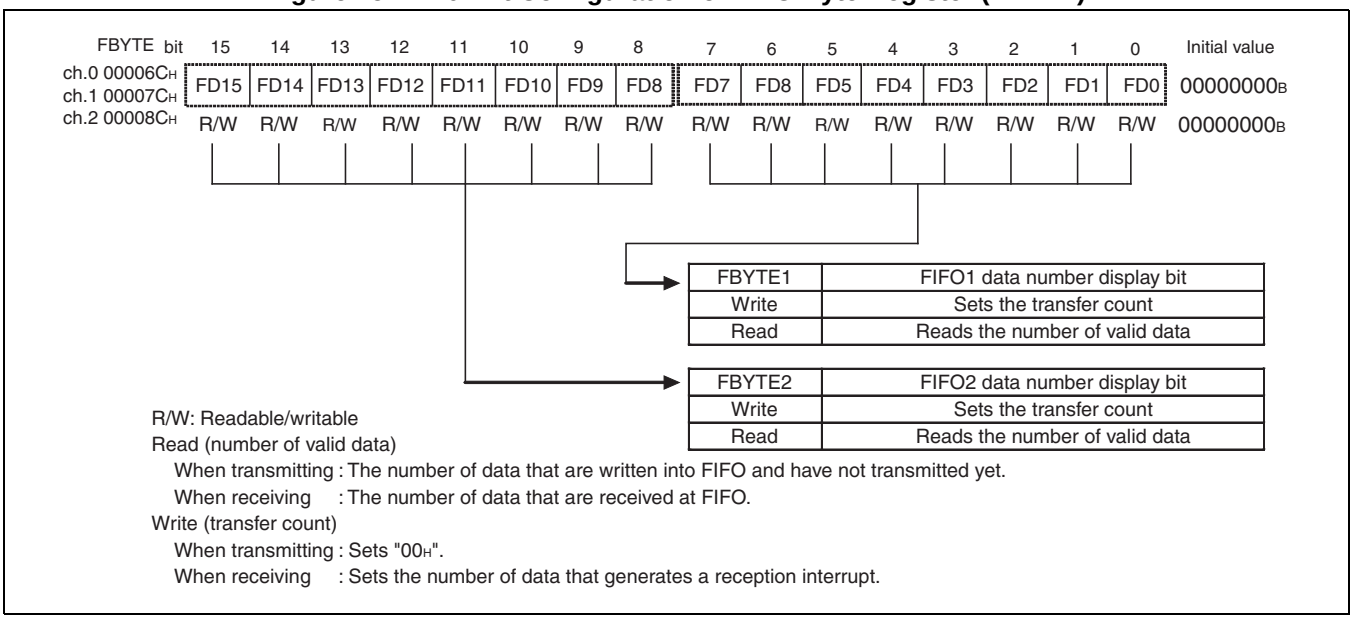
15.11.9 FIFO Byte Register (FBYTE)

FIFO byte register (FBYTE) indicates the number of valid data for FIFO.

■ Bit Configuration of FIFO Byte Register (FBYTE)

Figure 15.11-10 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 15.11-10 Bit Configuration of FIFO Byte Register (FBYTE)



FIFO byte register (FBYTE) indicates the number of valid data of FIFO. The number varies as follows, depending on the setting of FSEL bit in the FCR1 register.

Table 15.11-9 The Number of Data Displayed

FSEL	FIFO selection	The number of byte displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value of the transfer count of the FIFO byte register (FBYTE) is "08_H".
- Set the number of data to generate a reception interrupt flag to the FBYTE in the reception FIFO. An interrupt flag (RDRF) is set to "1" when the defined transfer count matches with the number of data displayed in the FIFO byte register (FBYTE).
- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.

- To receive the data with the master operation (master reception), set TIE and TBIE bits to "0", specify the number of data to receive in the FIFO byte register (FBYTE) of the transmission FIFO, and write "0" to FDRQ bit. Then, the specified number of data for serial clock can be output when TXE bit is "1" to receive the specified number of data. If you want to set TIE and TBIE bits to "1", wait until FDRQ bit becomes "1".

Notes:

- In the master operation, set "00_H" to FBYTE of the transmission FIFO except that data is received.
- When receiving data with the master operation, set the number of transmission data when the transmission FIFO is empty and also TIE and TBIE bits are set to "0".
- If you want to disable the reception (RXE=0) while receiving the data with the master operation, disable the transmission FIFO before disabling the transmission and reception.
- Set a data more than "1" to the FBYTE in the reception FIFO.
- Disable the reception before changing the FBYTE in the reception FIFO.
- You cannot use any read-modify-write (RMW) instruction to this register.
- The setting that exceeds the FIFO size is disabled.

15.12 Interrupts of CSIO (Clock Synchronous Multi Function Serial Interface)

There are reception and transmission interrupts for CSIO (clock synchronous multi function serial interface) and an interrupt request can be generated by the following sources:

- **When the received data is set in the reception data register (RDR), or a reception error occurs**
- **When the transmission data is transferred from the transmission data register (TDR) to the transmission shift register and the transmission has started**
- **Transmission bus idle (no transmission operation)**
- **Transmission FIFO data request**

■ Interrupts of CSIO

Table 15.12-1 shows the CSIO interrupt control bit and the interrupt source.

Table 15.12-1 CSIO Interrupt Control Bit and Interrupt Source

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	How to clear the interrupt request flag
Reception	RDRF	SSR	Receive 1 byte	SCR:RIE	Read reception data (RDR)
			The value set to FBYTE is received		Read reception data (RDR) until the reception FIFO gets empty
			Detect the reception idle state that continues over 8-bit time with the reception FIFO holding valid data when FRIIE bit is "1"		
	ORE	SSR	Overrun error		Write "1" to the reception error flag clear bit (SSR:REC)
Transmis-sion	TDRE	SSR	Transmission register is empty	SCR:TIE	Write to the transmission data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)*
	TBI	SSR	No transmission operation	SCR:TBIE	Write to the transmission data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)*
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Write "0" to the FIFO transmission data request bit (FCR1:FDRQ), or the transmission FIFO is full

*: Set TIE bit to "1" after TDRE bit has become "0".

15.12.1 Reception Interrupt Generation and Flag Set Timing

A reception completion (RDRF bit in SSR) and a reception error occurrence (ORE bit in SSR) are the sources of the reception interrupt.

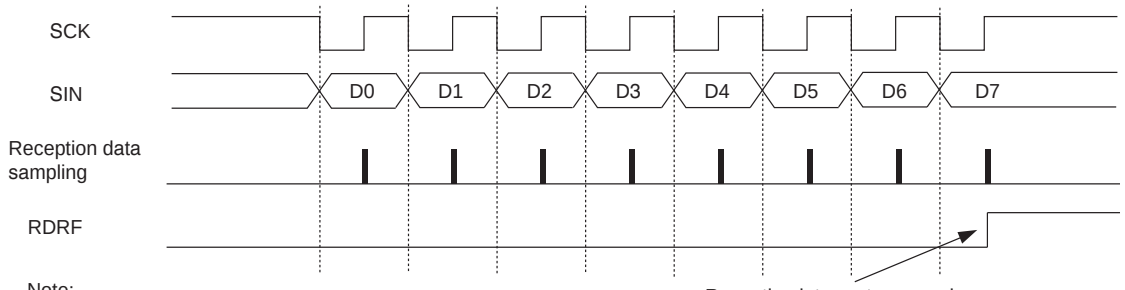
■ Reception Interrupt Generation and Flag Set Timing

Detection of the last data bit stores the reception data to the reception data register (RDR). When a reception is completed (SSR:RDRF = 1) or a reception error occurs (SSR:ORE = 1), each flag is set. If reception interrupt is enabled (SSR:RIE = 1) at this point, a reception interrupt occurs.

Note:

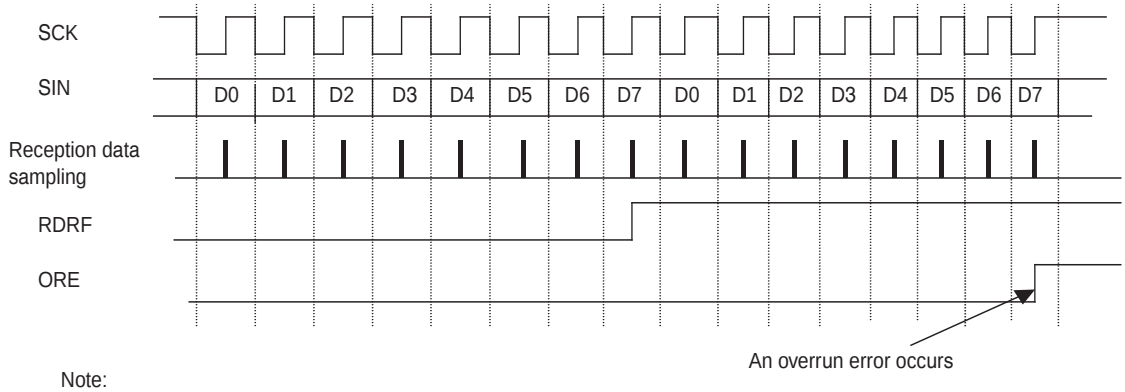
If a reception error occurs, the data in the reception data register (RDR) becomes invalid.

Figure 15.12-1 Reception Operation and Flag Set Timing



Note:
This figure shows the timing under the following conditions:
SCR: MS=1, SPI=0
ESCR: L2 to L0=000_B
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

Figure 15.12-2 ORE (Overrun Error) Flag Set Timing



Note:
This figure shows the timing under the following conditions:
SCR: MS=1, SPI=0
ESCR: L2 to L0=000_B
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

An overrun error occurs when next data is transferred before the reception data is read (RDRF = 1).

15.12.2 Interrupt Generation and Flag Set Timing When Using Reception FIFO

When using the reception FIFO, an interrupt is generated when the same value set to the FIFO byte register (FBYTE) has been received.

■ Reception Interrupt Generation and Flag Set Timing When Using Reception FIFO

When using the reception FIFO, the generation of an interrupt depends on the value set to the FIFO byte register (FBYTE).

- When the amount of data set to the FIFO byte register (FBYTE) as the transfer count has been received, the reception data full flag bit in the serial status register (RDRF bit in SSR) is set to "1". If reception interrupt is enabled (SCR:RIE) at this point, a reception interrupt occurs.
- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.
- When the reception data (RDR) is read until the reception FIFO gets empty, the reception data full flag bit (SSR:RDRF) is cleared.
- If next data is received with the FIFO size displayed as the number of data that can be received, an overrun error occurs (SSR:ORE = 1).

Figure 15.12-3 Reception Interrupt Generation Timing When Using Reception FIFO

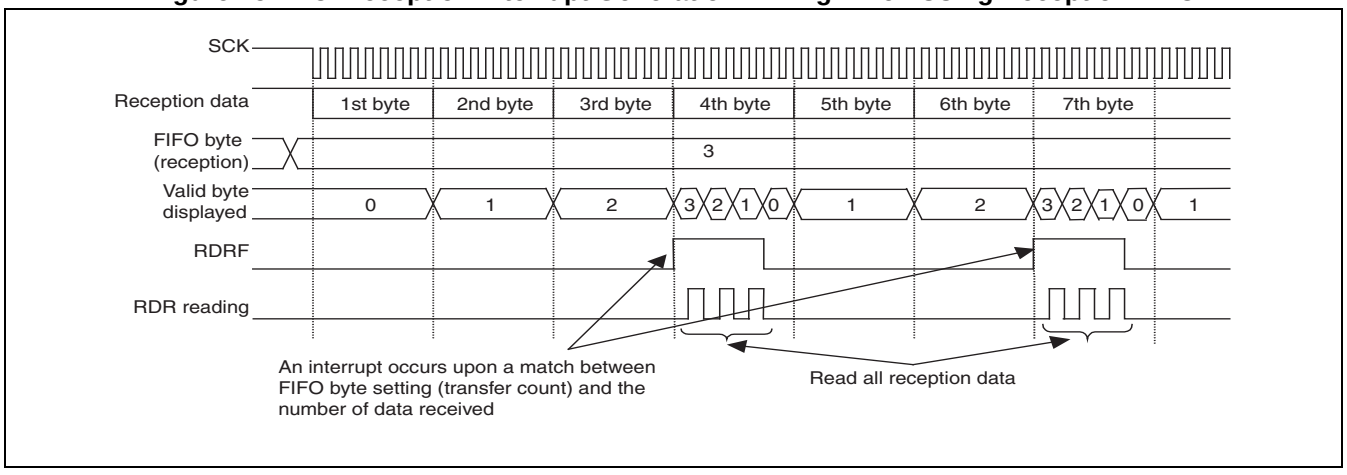
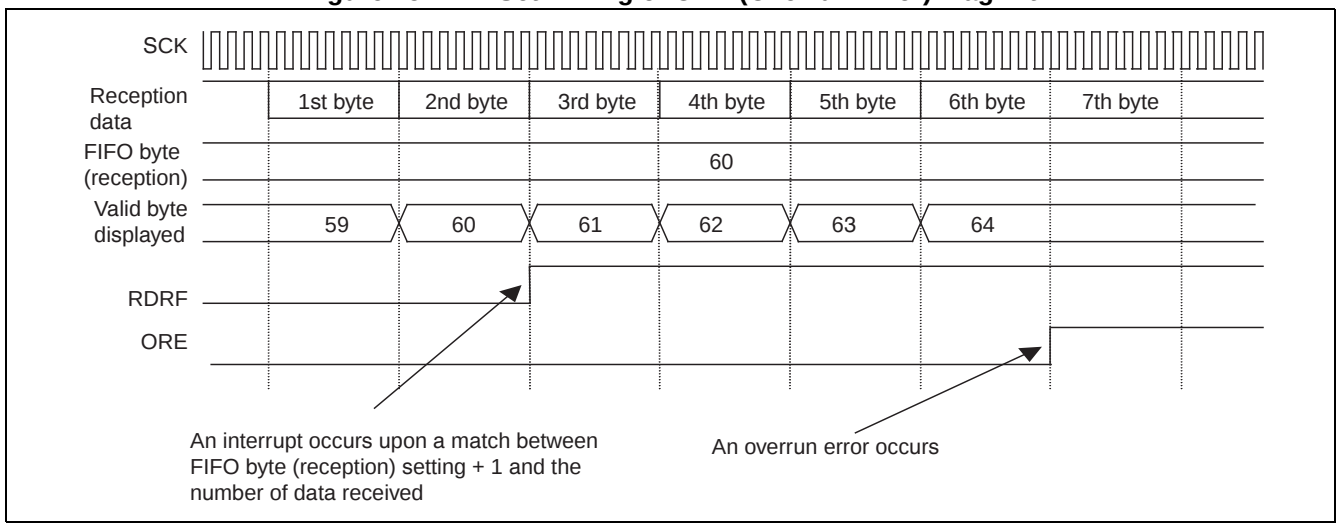


Figure 15.12-4 Set Timing of ORE (Overrun Error) Flag Bit



An overrun error occurs when next data is received with the FIFO size displayed in the FIFO. The figure shows an example where a FIFO size of 64 bytes is used.

15.12.3 Transmission Interrupt Generation and Flag Set Timing

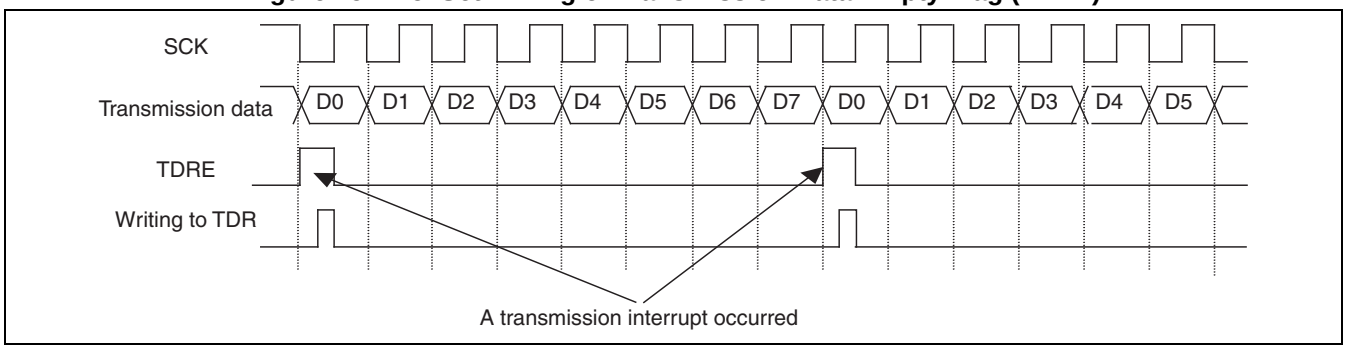
A transmission interrupt occurs when the transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and the transmission has started, or when the transmission operation is not in progress (SSR:TBI = 1).

■ Transmission Interrupt Generation and Flag Set Timing

● Set timing of the transmission data empty flag (TDRE)

Once the data written into the transmission data register (TDR) is transferred to the transmission shift register, next data can be written into TDR (SSR:TDRE = 1). If transmission interrupt is enabled (SCR:TIE = 1) at this point, a transmission interrupt occurs. As TDRE bit is read only, it is cleared to "0" by writing data into the transmission data register (TDR).

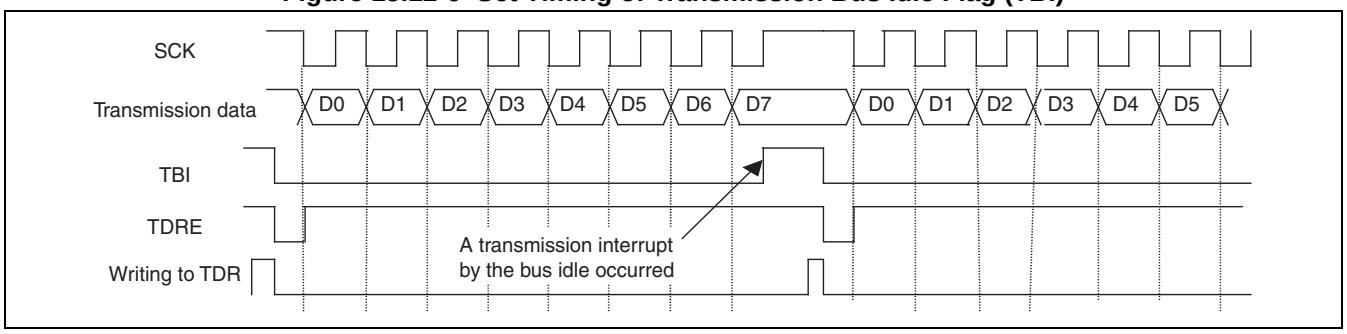
Figure 15.12-5 Set Timing of Transmission Data Empty Flag (TDRE)



● Set timing of the transmission bus idle flag (TBI)

TBI bit in the serial status register (SSR) is set to "1" when the transmission data register is empty (TDRE = 1) and the transmission operation is not in progress. If transmission bus idle interrupt is enabled (SCR:TBIE = 1) at this point, a transmission interrupt occurs. TBI bit and the transmission interrupt request are cleared when a transmission data is set to the transmission data register (TDR).

Figure 15.12-6 Set Timing of Transmission Bus Idle Flag (TBI)



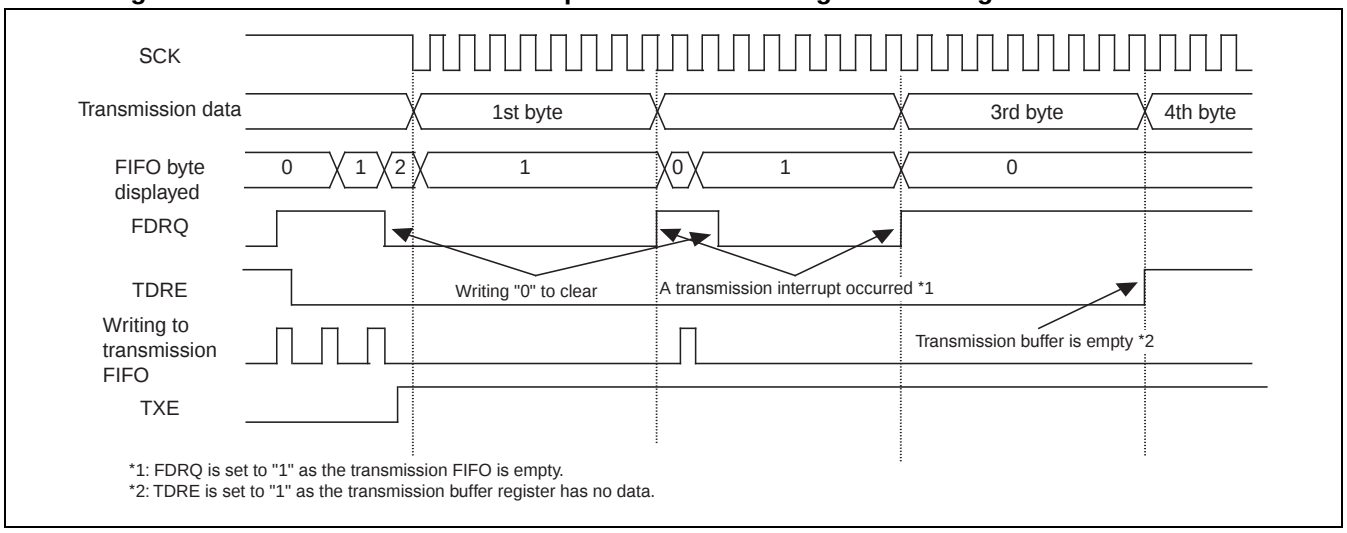
15.12.4 Interrupt Generation and Flag Set Timing When Using Transmission FIFO

When using the transmission FIFO, an interrupt occurs when the transmission FIFO has no data.

■ Transmission Interrupt Generation and Flag Set Timing When Using Transmission FIFO

- When the transmission FIFO has no data, FIFO transmission data request bit (FCR1:FDRQ) is set to "1".
  If FIFO transmission interrupt is enabled (FCR1:FTIE = 1) at this point, a transmission interrupt occurs.
- Once the transmission interrupt has been generated and you have written the required data into the transmission FIFO, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- The FIFO transmission data request bit (FCR1:FDRQ) becomes "0" when the transmission FIFO gets full.
- You can verify if the transmission FIFO has data or not by reading the FIFO byte register (FBYTE). FBYTE = 00_H indicates that the transmission FIFO has no data.

Figure 15.12-7 Transmission Interrupt Generation Timing When Using Transmission FIFO



15.13 Operations of CSIO (Clock Synchronous Multi Function Serial Interface)

The clock synchronous method is employed as the transfer method.

■ Operations of CSIO (Clock Synchronous Multi Function Serial Interface)

(1) Normal transfer (I)

● Features

	Item	Description
1	Mark level of the serial clock (SCK)	"H"
2	Transmission data output timing	Falling edge of SCK
3	Reception data sampling	Rising edge of SCK
4	Data length	5 to 9 bits

● Register setting

The following table shows the register setting required for the normal transfer (I):

Table 15.13-1 Normal Transfer (I) Register Setting

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	0	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set "1"

0: Set "0"

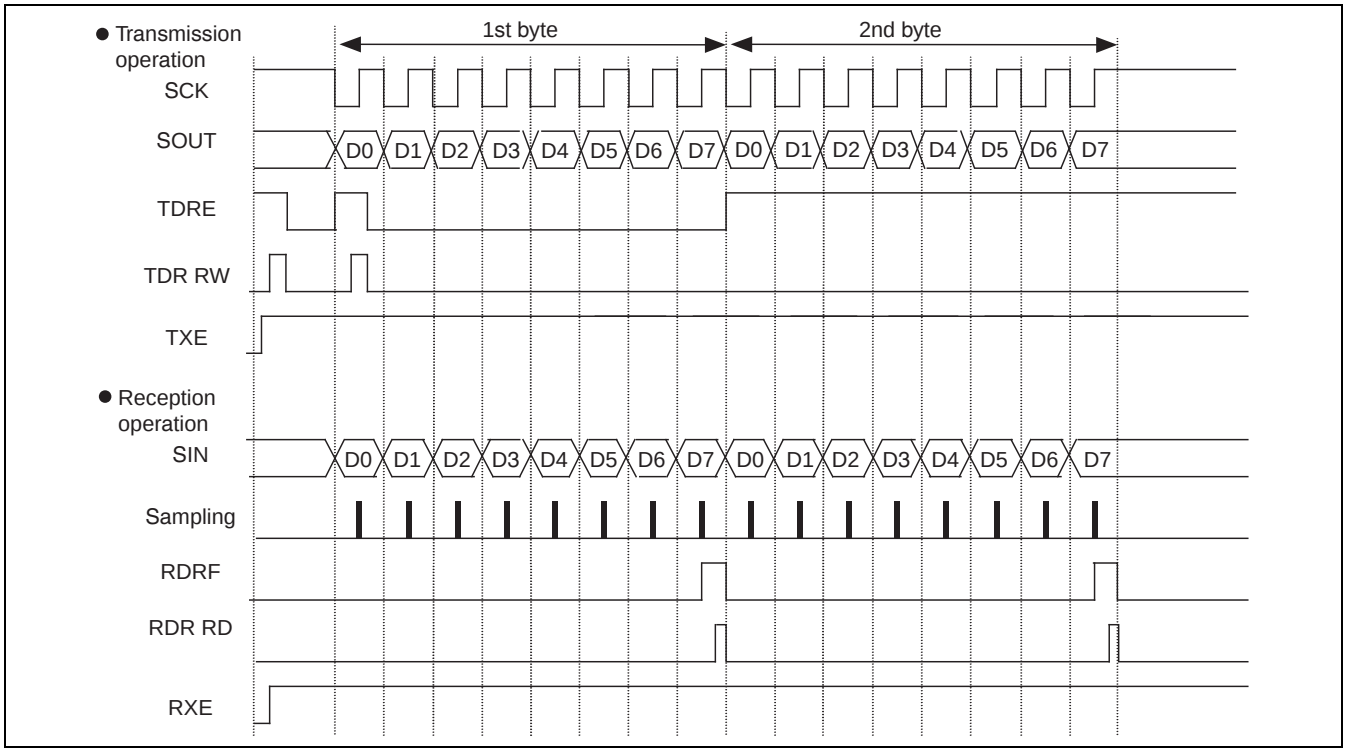
*: Setting defined by the user

Note:

The values set for the bits above (1/0) varies depending on the master operation and slave operation. Set as follows:

Master transmission : SCR: MS = 0, SMR: SCKE = 1, SOE = 1
Master reception : SCR: MS = 0, SMR: SCKE = 1, SOE = 0
Slave transmission : SCR: MS = 1, SMR: SCKE = 0, SOE = 1
Slave reception : SCR: MS = 1, SMR: SCKE = 0, SOE = 0

● Normal Transfer (I) Timing Chart



● Operating explanation

[1] Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1), and disable the reception operation (SCR:RXE = 0). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the falling edge of the serial clock (SCK) output.

  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) Disable the serial data output (SMR:SOE = 0), and enable the transmission operation (SCR:TXE = 1) and the reception operation (SCR:RXE = 1). Then, when you write a dummy data to TDR, the reception data is sampled in synchronization with the rising edge of the serial clock output (SCK).

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

Notes:

- If you want the reception operation only, write a dummy data to TDR in order to output the serial clock (SCK).
- When the transmission and reception FIFO are enabled, setting the number of frames you want to transfer to the FIFO byte register (FBYTE) outputs the serial clock (SCK) of the desired number of frames.

[2] Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the falling edge of the serial clock (SCK) input.

  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) If you disable the serial data output (SMR:SOE = 0) and enable the reception operation (SCR:RXE = 1), the reception data is sampled in synchronization with the rising edge of the serial clock (SCK) input.

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

(2) Normal transfer (II)

● Features

	Item	Description
1	Mark level of the serial clock (SCK)	"L"
2	Transmission data output timing	Rising edge of SCK
3	Reception data sampling	Falling edge of SCK
4	Data length	5 to 9 bits

● Register setting

The following table shows the register setting required for the normal transfer (II):

Table 15.13-2 Normal Transfer (II) Register Setting

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	1	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set "1"

0: Set "0"

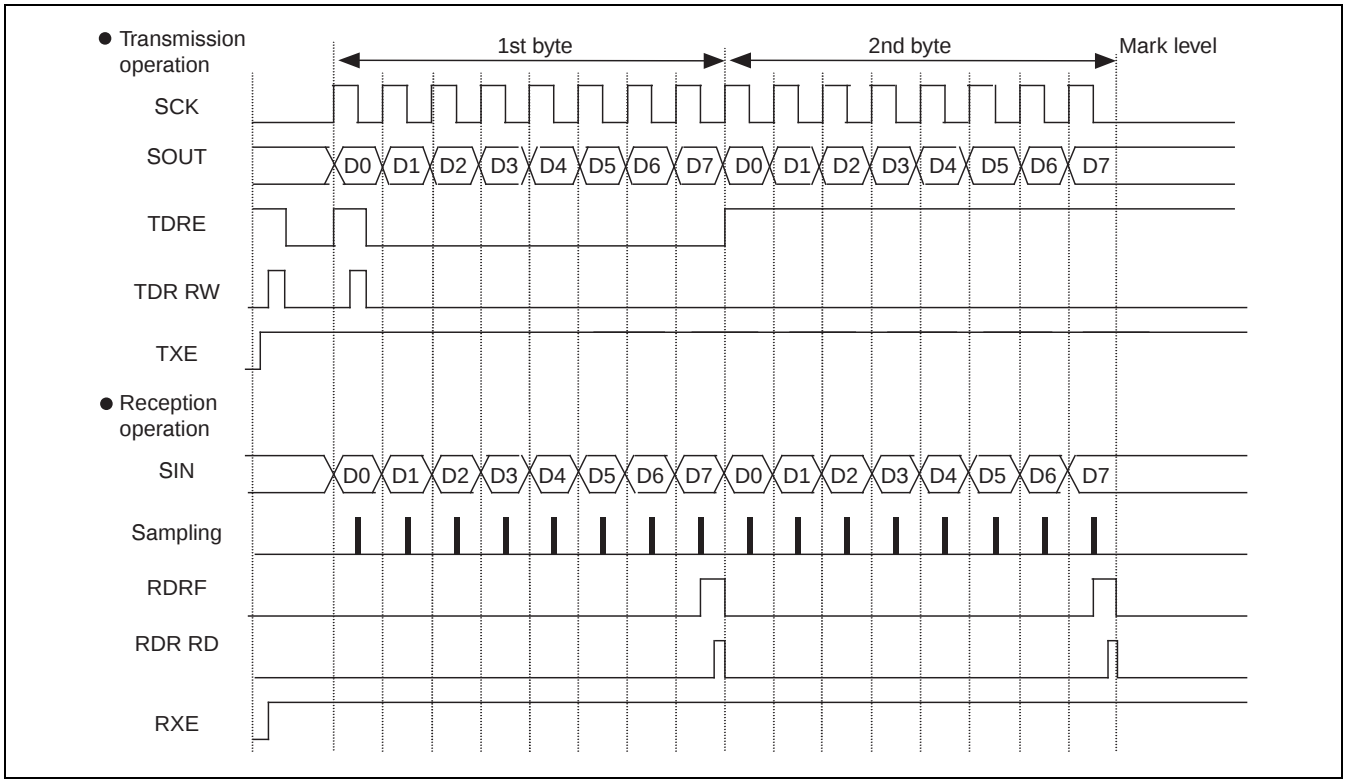
*: Setting defined by the user

Note:

The values set for the bits above (1/0) varies depending on the master operation and slave operation. Set as follows:

Master transmission : SCR: MS = 0, SMR: SCKE = 1, SOE = 1
Master reception : SCR: MS = 0, SMR: SCKE = 1, SOE = 0
Slave transmission : SCR: MS = 1, SMR: SCKE = 0, SOE = 1
Slave reception : SCR: MS = 1, SMR: SCKE = 0, SOE = 0

● Normal Transfer (II) Timing Chart



● Operating explanation

[1] Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation
  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1), and disable the reception operation (SCR:RXE = 0). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the rising edge of the serial clock (SCK) output.
  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.
- Reception operation
  (1) Disable the serial data output (SMR:SOE = 0), and enable the transmission operation (SCR:TXE = 1) and the reception operation (SCR:RXE = 1). Then, when you write a dummy data to TDR, the reception data is sampled in synchronization with the falling edge of the serial clock output (SCK).
  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.
  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

Notes:

- If you want the reception operation only, write a dummy data to TDR in order to output the serial clock (SCK).
- When the transmission and reception FIFO are enabled, setting the number of frames you want to transfer to the FIFO byte register (FBYTE) outputs the serial clock (SCK) of the desired number of frames.

[2] Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the rising edge of the serial clock (SCK) input.

  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) If you disable the serial data output (SMR:SOE = 0) and enable the reception operation (SCR:RXE = 1), the reception data is sampled in synchronization with the falling edge of the serial clock (SCK) input.

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

(3) SPI transfer (I)

● Features

	Item	Description
1	Mark level of the serial clock (SCK)	"H"
2	Transmission data output timing	Rising edge of SCK
3	Reception data sampling	Falling edge of SCK
4	Data length	5 to 9 bits

● Register setting

The following table shows the register setting required for the SPI transfer (I):

Table 15.13-3 SPI Transfer (I) Register Setting

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	0	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

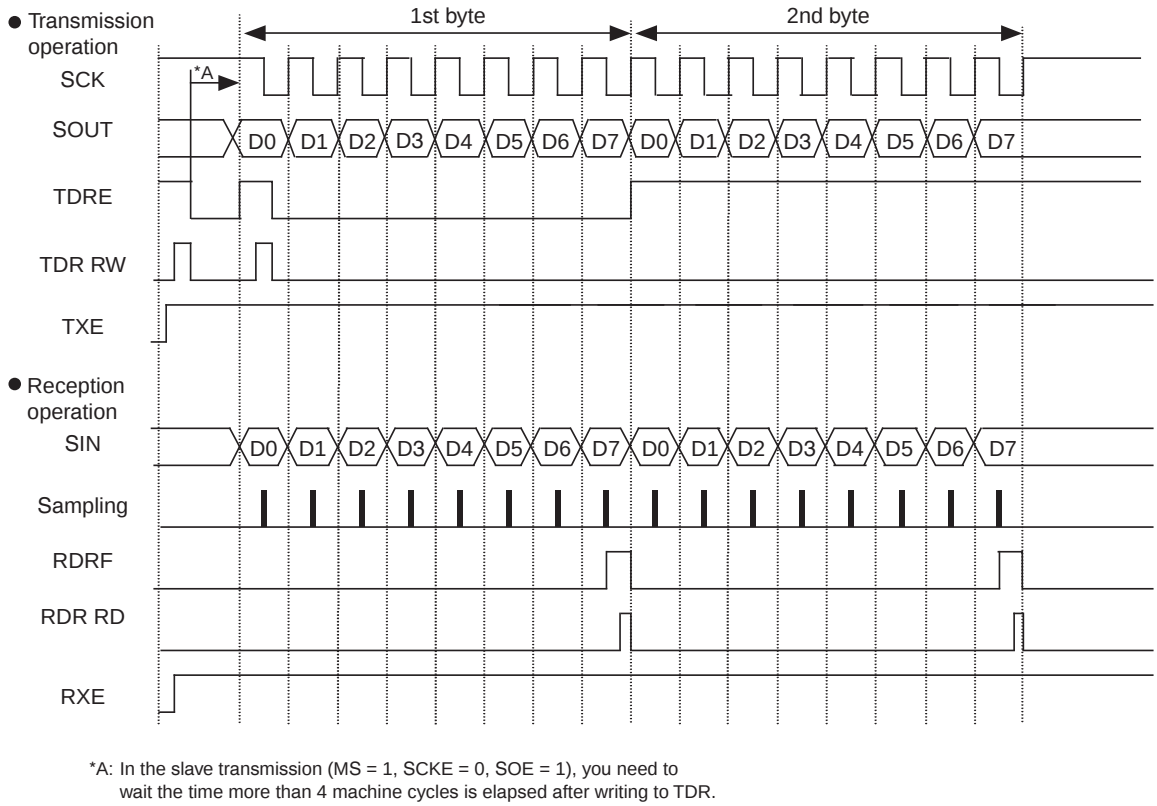
1: Set "1"
0: Set "0"
*: Setting defined by the user

Note:

The values set for the bits above (1/0) varies depending on the master operation and slave operation. Set as follows:

Master transmission : SCR: MS = 0, SMR: SCKE = 1, SOE = 1
Master reception : SCR: MS = 0, SMR: SCKE = 1, SOE = 0
Slave transmission : SCR: MS = 1, SMR: SCKE = 0, SOE = 1
Slave reception : SCR: MS = 1, SMR: SCKE = 0, SOE = 0

● SPI Transfer (I) Timing Chart



*A: In the slave transmission (MS = 1, SCKE = 0, SOE = 1), you need to wait the time more than 4 machine cycles is elapsed after writing to TDR.

● Operating explanation

[1] Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1), and disable the reception operation (SCR:RXE = 0). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the 1st bit. Then, the transmission data is output in synchronization with the rising edge of the serial clock (SCK) output.

  (2) TDRE bit in the serial status register (SSR) becomes "1" before the half cycle of the falling edge of the first serial clock. A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) Disable the serial data output (SMR:SOE = 0), and enable the transmission operation (SCR:TXE = 1) and the reception operation (SCR:RXE = 1). Then, when you write a dummy data to TDR, the reception data is sampled in synchronization with the falling edge of the serial clock output (SCK).

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

Notes:

- If you want the reception operation only, write a dummy data to TDR in order to output the serial clock (SCK).
- When the transmission and reception FIFO are enabled, setting the number of frames you want to transfer to the FIFO byte register (FBYTE) outputs the serial clock (SCK) of the desired number of frames.

[2] Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the 1st bit. Then, the transmission data is output in synchronization with the rising edge of the serial clock (SCK) output.

  (2) TDRE bit in the serial status register (SSR) becomes "1" before the half cycle of the falling edge of the first serial clock. A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) If you disable the serial data output (SMR:SOE = 0) and enable the reception operation (SCR:RXE = 1), the reception data is sampled in synchronization with the falling edge of the serial clock (SCK) input.

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

(4) SPI transfer (II)

● Features

	Item	Description
1	Mark level of the serial clock (SCK)	"L"
2	Transmission data output timing	Falling edge of SCK
3	Reception data sampling	Rising edge of SCK
4	Data length	5 to 9 bits

● Register setting

The following table shows the register setting required for the SPI transfer (II):

Table 15.13-4 SPI Transfer (II) Register Setting

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

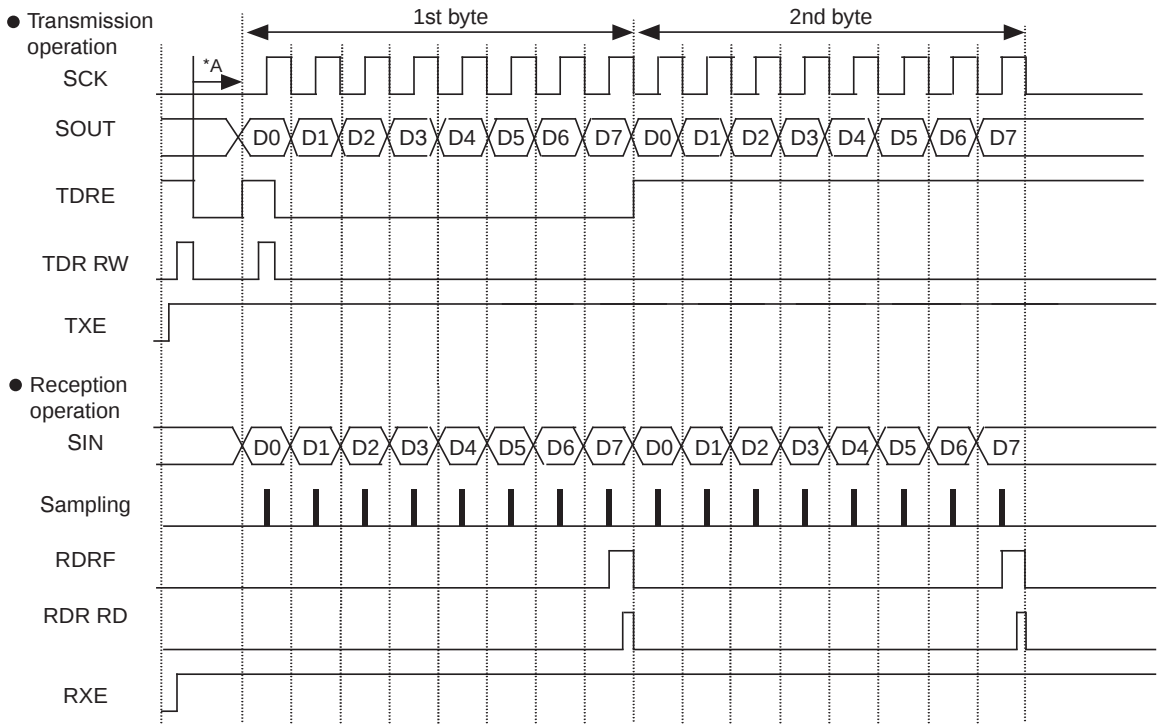
1: Set "1"
0: Set "0"
*: Setting defined by the user

Note:

The values set for the bits above (1/0) varies depending on the master operation and slave operation. Set as follows:

Master transmission : SCR: MS = 0, SMR: SCKE = 1, SOE = 1
Master reception : SCR: MS = 0, SMR: SCKE = 1, SOE = 0
Slave transmission : SCR: MS = 1, SMR: SCKE = 0, SOE = 1
Slave reception : SCR: MS = 1, SMR: SCKE = 0, SOE = 0

● SPI Transfer (II) Timing Chart



*A: In the slave transmission (MS = 1, SCKE = 0, SOE = 1), you need to wait the time more than 4 machine cycles is elapsed after writing to TDR.

● Operating explanation

[1] Master operation (SCR: MS = 0, SMR: SCKE = 1)

- Transmission operation

  (1) Enable the serial data output (SMR: SOE = 1) and the transmission operation (SCR:TXE = 1), and disable the reception operation (SCR:RXE = 0). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the falling edge of the serial clock (SCK) output.

  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) Disable the serial data output (SMR:SOE = 0), and enable the transmission operation (SCR:TXE = 1) and the reception operation (SCR:RXE = 1). Then, when you write a dummy data to TDR, the reception data is sampled in synchronization with the rising edge of the serial clock output (SCK).

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

Notes:

- If you want the reception operation only, write a dummy data to TDR in order to output the serial clock (SCK).
- When the transmission and reception FIFO are enabled, setting the number of frames you want to transfer to the FIFO byte register (FBYTE) outputs the serial clock (SCK) of the desired number of frames.

[2] Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) Enable the serial data output (SMR:SOE = 1) and the transmission operation (SCR:TXE = 1). Then, when you write a transmission data to TDR, TDRE bit in the serial status register (SSR) becomes "0" to output the transmission data in synchronization with the falling edge of the serial clock (SCK) output.

  (2) When the 1st bit of the transmission data is output, TDRE bit in the serial status register (SSR) becomes "1". A transmission interrupt request is also output when the transmission interrupt is enabled (SCR:TIE = 1). At this point, 2nd byte of the transmission data can be written.

- Reception operation

  (1) If you disable the serial data output (SMR:SOE = 0) and enable the reception operation (SCR:RXE = 1), the reception data is sampled in synchronization with the rising edge of the serial clock (SCK) input.

  (2) Receiving the last bit sets RDRF bit in the serial status register (SSR) to "1". A reception interrupt request is also output when the reception interrupt is enabled (SCR:RIE = 1).
  At this point, the reception data (RDR) can be read.

  (3) If the reception data register (RDR) is read, the RDRF bit in the serial status register (SSR) is cleared to "0".

15.14 CSIO Dedicated Baud Rate Generator

The CSIO dedicated baud rate generator only functions in the master operation. However, if you use the reception FIFO, set the dedicated baud rate generator even in the slave operation.

■ CSIO (Clock Synchronous Multi Function Serial Interface) Baud Rate Selection

The setting for the dedicated baud rate generator varies depending on the master operation and slave operation.

● Master operation

Select the baud rate by dividing the internal clock by the dedicated baud rate generator.

- There are 2 internal reload counters that correspond to transmission and reception serial clocks. Baud rate can be specified by setting the reload value for 15 bits with the baud rate generator registers 1, 0 (BGR1/BGR0).
- The reload counter divides the internal clock by the set value.

● Slave operation

The dedicated baud rate generator does not function in the slave operation (SCR:MS = 1).

(The external clock input from the clock input pin SCK is directly used)

Note:

If you use the reception FIFO, set the dedicated baud rate generator even in the slave operation.

15.14.1 Baud Rate Setting

This section shows the setting for the baud rate. The calculation result of serial clock frequency is also described.

■ Calculating Baud Rate

2 of 15-bit reload counters are set using the baud rate generator registers 1, 0 (BGR1/BGR0).

The equation to calculate the baud rate is shown below:

(1) Reload value

$V = \phi / b - 1$

V: Reload value, b: Baud rate, ϕ: Machine clock frequency

(2) Example of calculation

Where the machine clock is 16 MHz, the internal clock is enabled, and the baud rate is set to 19200 bps, the reload value is calculated as follows:
Reload value:

$V = (16 \times 1000000) / 19200 - 1 = 832$

Therefore, the baud rate is:

$b = (16 \times 1000000) / (832 + 1) = 19208$ (bps)

(3) Baud rate error

Baud rate error can be calculated by the following equation:

Error (%) = (calculated value - desired value) / desired value $\times$ 100

Ex.) Where the machine clock is 20 MHz and the desired baud rate is set to 153600 bps:

Reload value = $(20 \times 1000000)/153600 - 1 = 129$

Baud rate (calculated value) = $(20 \times 1000000)/(129+1) = 153846$ (bps)

Error (%) = $(153846 - 153600)/153600 \times 100 = 0.16$ (%)

Notes:

- If the reload value is set to "0", the reload counter will stop.
- If the reload value is an even number, "L" and "H" widths of the serial clock become as follows depending on the setting for SCINV bit:If it is an odd number, "H" and "L" widths of the serial clock have the same length.
  When SCINV = 0, "H" width of the serial clock is longer than "L" width by 1 cycle of the machine clock.
  When SCINV = 1, "L" width of the serial clock is longer than "H" width by 1 cycle of the machine clock.
- Be sure to set the reload value to 3 or more.

■ Reload Value and Baud Rate for Each Machine Clock Frequency

Table 15.14-1 Reload Value And Baud Rate

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8 M	-	-	-	-	-	-	-	-	-	-	3	0
6 M	-	-	-	-	-	-	-	-	3	0	-	-
5 M	-	-	-	-	-	-	3	0	-	-	-	-
4 M	-	-	-	-	3	0	4	0	5	0	7	0
2.5 M	-	-	3	0	-	-	-	-	-	-	-	-
2 M	3	0	4	0	7	0	9	0	11	0	15	0
1 M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

• Value : Value set to BGR1/BGR0 registers

• ERR : Baud rate error (%)

■ Functions of the Reload Counter

The reload counter has the transmission and reception reload counters that function as the dedicated baud rate generator. The reload counter consists of 15-bit register and generates a transmission and reception clock from the internal clock.

■ Start of a Count

The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).

■ Restart

The reload counters can be restarted for the following conditions:

- For both transmission/reception reload counters

  Programmable reset (SCR: UPCL bit)

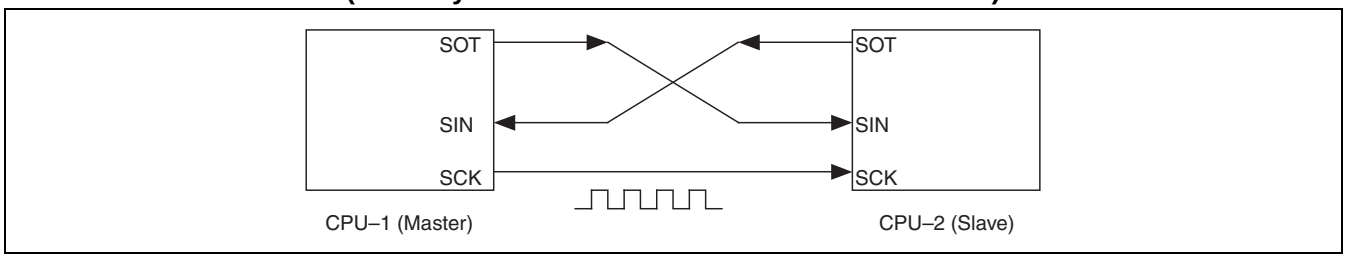
15.15 Setting Procedure and Program Flow for CSIO (Clock Synchronous Multi Function Serial Interface)

In CSIO (clock synchronous multi function serial interface), a synchronous serial bidirectional communication is available.

■ Inter-CPU Connection

The bidirectional communication is selected in CSIO (clock synchronous multi function serial interface). As shown in Figure 15.15-1 CPUs are connected each other.

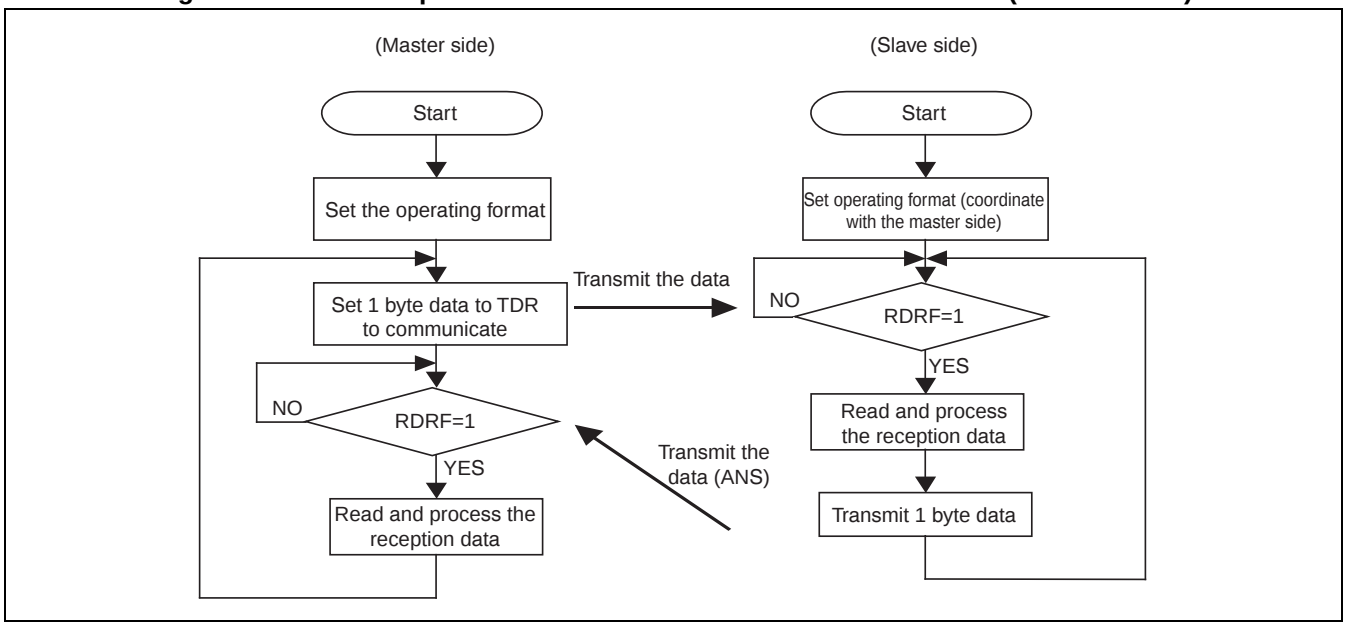
Figure 15.15-1 Example of Bidirectional Communication Connection in CSIO (Clock Synchronous Multi Function Serial Interface)



■ Flowchart

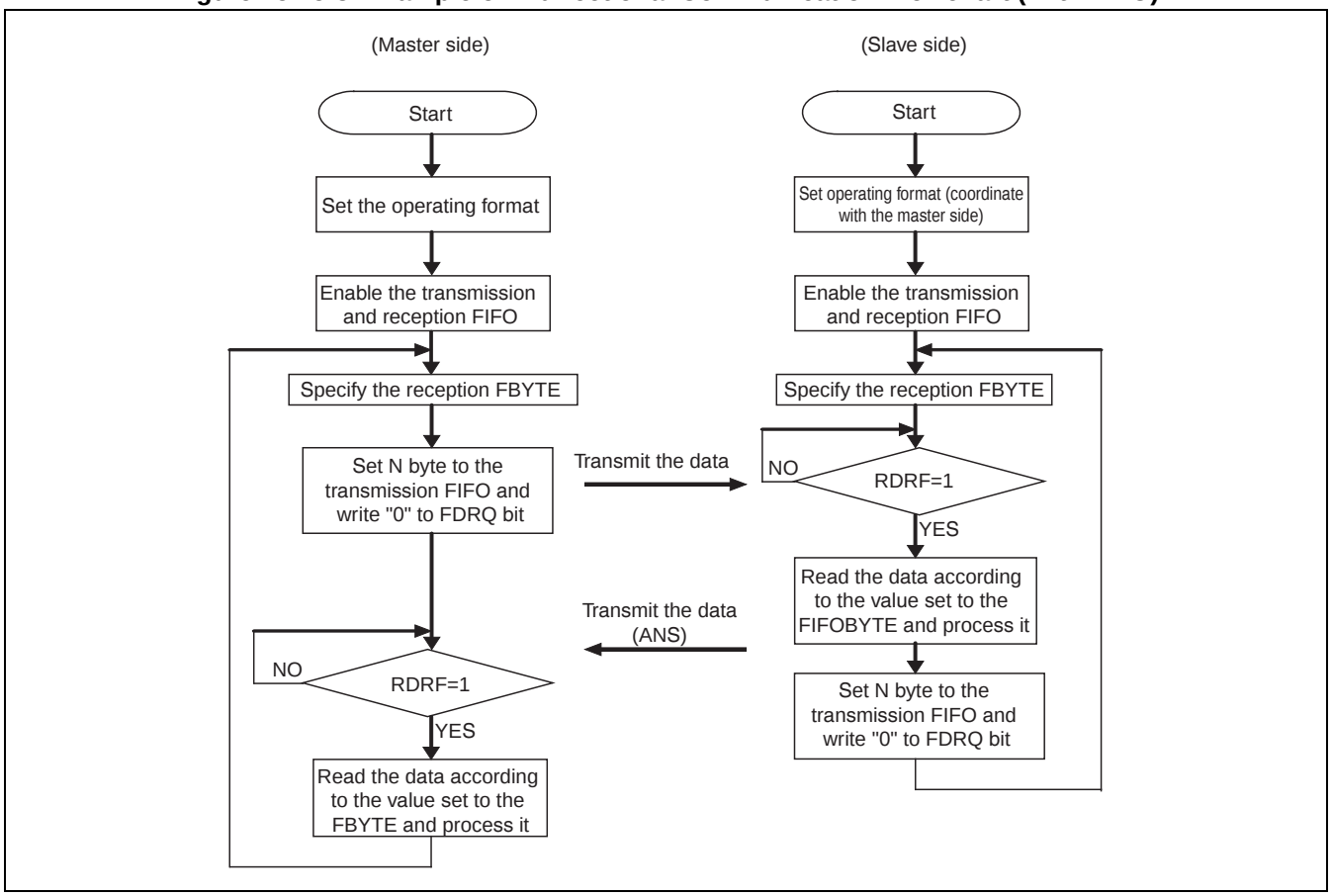
● Without FIFO

Figure 15.15-2 Example of Bidirectional Communication Flowchart (Without FIFO)



● With FIFO

Figure 15.15-3 Example of Bidirectional Communication Flowchart (With FIFO)



15.16 Notes on CSIO Mode

The notes for when you use the CSIO mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.

15.17 Overview of the I2C Interface

I2C interface that supports the Inter IC bus operates as the master/slave device on the I2C bus. In addition, it has the transmission/reception FIFO (Maximum 16-byte each).

■ Functions of the I2C Interface

I2C interface has the following functions:

- Master/slave transmission and reception function
- Arbitration function
- Clock synchronization function
- Transfer direction detection function
- Function that generates and detects a repetitive start condition
- Bus error detection function
- General call addressing function
- 7-bit addressing as master and slave
- Capable of generating an interrupt upon the transfer and bus error
- 10-bit addressing function can be supported by program
- Noise filter provided

■ Functions of FIFO

FIFO has the following functions:

- Transmission/reception FIFO provided (maximum size: transmission FIFO 16-byte, reception FIFO 16-byte)
- Transmission FIFO and reception FIFO can be selected
- Capable of retransmitting the transmission data
- Reception FIFO interrupt timing can be changed from the software
- Independent FIFO reset supported

MB91313A Series

15.18 Registers of the I2C Interface

This section shows the register list of I2C interface.

■ Register List of the I2C Interface

Table 15.18-1 Register List of the I2C Interface

	Address		bit15 bit8	bit7 bit0
I2C	$00X0_H$	$00X1_H$	IBCR (I2C bus control register)	SMR (serial mode register)
	$00X2_H$	$00X3_H$	SSR (serial status register)	IBSR (I2C bus status register)
	$00X4_H$	$00X5_H$	-	RDR/TDR (transmission and reception data register)
	$00X6_H$	$00X7_H$	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	$00X8_H$	$00X9_H$	ISMK (7-bit slave address mask register)	IBSA (7-bit slave address register)
FIFO	$00YA_H$	$00YB_H$	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)
	$00YC_H$	$00YD_H$	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)

(X = 06_H, 07_H, 08_H, 09_H, $0A_H$, $0B_H$, $1B_H$, $1C_H$, $1D_H$, $1E_H$, $1F_H$, Y = 06_H, 07_H, 08_H)

Table 15.18-2 Bit Arrangement of the I2C Interface

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	-	RIE	TIE	ITST1	ITST0
SSR/ IBSR	REC	TSET	-	-	ORE	RDRF	TDRE	-	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
TDR1/ TDR0	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	FTST1	FTST0	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

15.18.1 I2C Bus Control Register (IBCR)

I2C bus control register (IBCR) can select master/slave mode, generate the repetitive start condition, enable the acknowledgment and interrupt, and display the interrupt flag.

■ I2C Bus Control Register (IBCR)

Figure 15.18-1 shows the bit configuration of the I2C bus control register (IBCR), and Table 15.18-3 shows the function of each bit.

Figure 15.18-1 Bit Configuration of I2C Bus Control Register (IBCR)

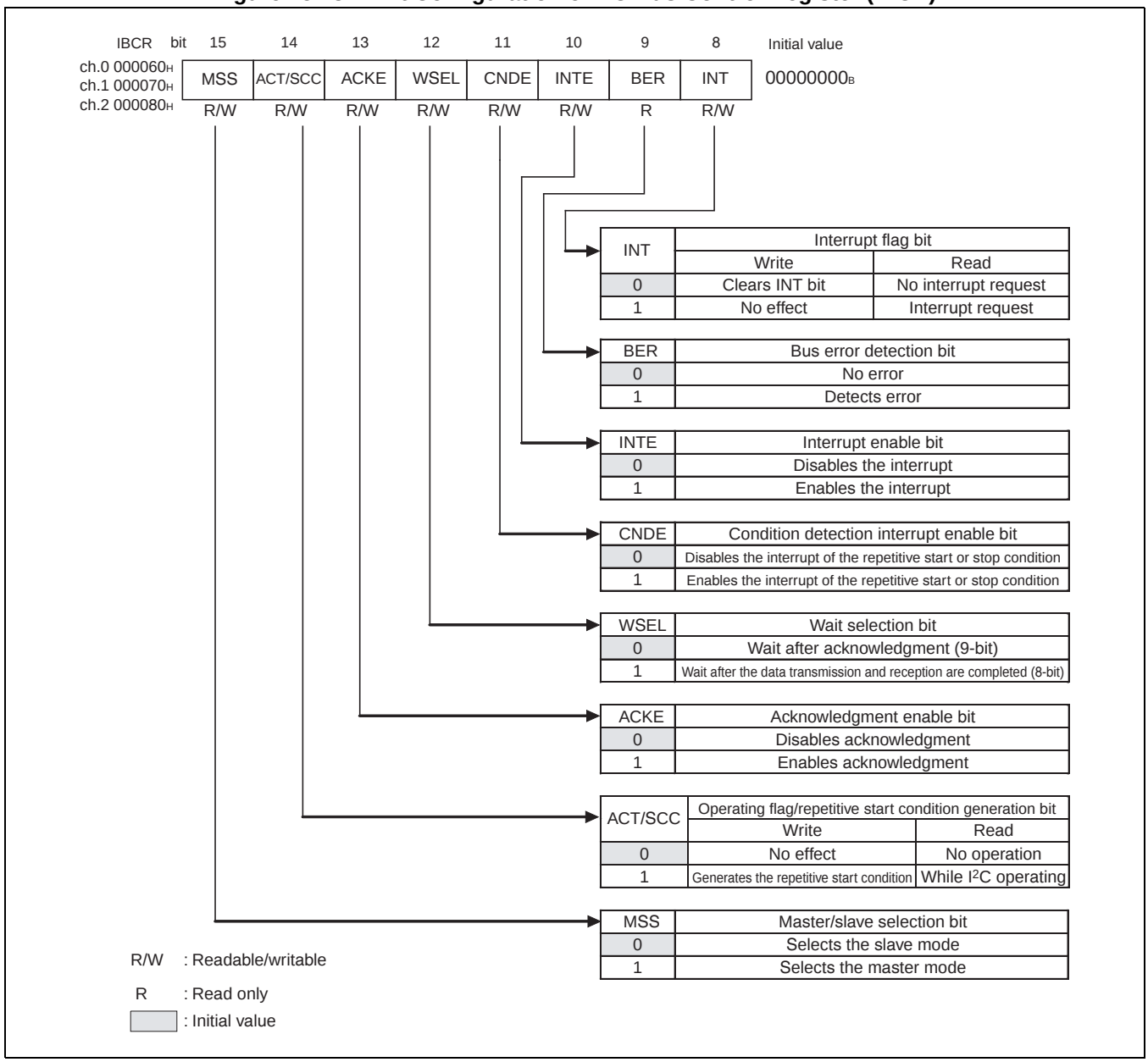


Table 15.18-3 Function Description of Each Bit in the I2C Bus Control Register (IBCR) (1 / 4)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit15</td><td>MSS:
Master/slave selection bit</td><td>
• If this bit is set to "1", the master mode is selected when I2C bus is in idle state (EN = 1, BB = 0).
• If BB bit in the IBSR register is "1", generation of the start condition is waited until BB bit becomes "0" when "1" is set to this bit. While waiting, if it operates as a slave by matching the slave address, this bit becomes "0" and AL bit in the IBSR register becomes "1".
• If the interrupt flag (INT) is "1" during the master operation (MSS = 1, ACT = 1), a stop condition is generated when "0" is written to this bit.

MSS bit is cleared when:
(1) I2C interface is disabled (EN bit = 0)
(2) Arbitration lost is generated
(3) Bus error is detected (BER bit = 1)
(4) Write "0" to MSS bit when INT = 1.
The relationship between the MSS and ACT bits is shown below:
<table>
<tr><th>MSS bit</th><th>ACT bit</th><th>State</th></tr>
<tr><td>0</td><td>0</td><td>Idle</td></tr>
<tr><td>0</td><td>1</td><td>ACK response* to the reserved address or the match with slave address, and slave operation is in progress (slave mode)</td></tr>
<tr><td>1</td><td>0</td><td>Waiting for master operation</td></tr>
<tr><td>1</td><td>1</td><td>Master operation is in progress (master mode)</td></tr>
</table>
*: ACK response: Indicates that SDA in the I2C bus is "L" while the acknowledgment period.

Notes:
• If MSS bit is set to "1", change MSS bit to "0" when MSS bit is "1" and INT bit is "1". If "0" is written to MSS bit when ACT bit is "1", INT bit is also cleared to "0".
• During master operation, "1" is read as long as ACT bit is "1", even if "0" is written to MSS bit.
</td></tr>
</table>

Table 15.18-3 Function Description of Each Bit in the I2C Bus Control Register (IBCR) (2 / 4)

Bit name		Function
bit14	ACT/SCC: Operating flag/repetitive start condition generation bit	This bit has a different role when reading and writing. Read: ACT bit / Write: SCC bit ACT bit indicates that the IBCR register is operating as master or slave mode. Set conditions for ACT bit: (1) When the start condition is output to I2C bus (master mode) (2) When the slave address matches with the address transmitted from the master (slave mode) (3) When a reserved address is detected and an ACK response is performed to the address (slave mode is selected when MSS = 0) Reset conditions for ACT bit: <Master mode> (1) Stop condition detected (2) Arbitration lost detected (3) Bus error detection (4) I2C interface disabled (EN bit = 0) <Slave mode> (1) (Repetitive) start condition detected (2) Stop condition detected (3) When a reserved address is detected (RSA = 1) but an ACK response is not performed (4) I2C interface disabled (EN bit = 0) (5) Bus error generated (BER bit = 1) During master mode, a repetitive start is executed when "1" is written to this bit. Writing "0" to this bit is ignored. Notes: • Write "1" to SCC bit while master mode is interrupted (MSS = 1, ACT = 1, INT = 1). If ACT bit is "1", INT bit is cleared to "0" when "1" is written to SCC bit. • Writing "1" to this bit is disabled during slave mode (MSS = 0, ACT = 1). • If you write "1" to SCC bit and "0" to MSS bit, MSS bit is prioritized. • SCC bit is read by a read-modify-write (RMW) instruction.

Table 15.18-3 Function Description of Each Bit in the I2C Bus Control Register (IBCR) (3 / 4)

Bit name		Function
bit13	ACKE: Data byte acknowledgment enable bit	• Setting this bit to "1" will output "L" at an acknowledgment timing. • If ACT = 1, change this bit when INT bit is "1". This bit becomes invalid when: (1) An acknowledgment is performed to an address field other than the reserved address (auto-generation) (2) Data is transmitted (RSA = 0, TRX = 1, FBT = 0) (3) ACK response is always performed when the reception FIFO and slave reception are enabled (FE = 1, MSS = 0, ACT = 1) (4) If WSEL bit is "0", and the reception FIFO and master reception are enabled (FE = 1, MSS = 1, ACT = 1, WSEL = 0), ACK response is performed when TDRE bit is "0" and NACK response is performed when TDRE bit is "1" (5) When the reception FIFO is enabled, WSEL = 0, a reserved address is detected, and slave transmission is enabled (RSA = 1, TRX = 1, FBT = 1), ACK response is always performed. If you want to perform NACK response, disables the reception FIFO and set ACKE bit to "0" when interrupting after a reserved address is detected. (5) The reception FIFO is enabled, WSEL bit is "1", master reception is enabled, and the transmission data register has data (FE = 1, MSS = 1, ACT = 1, WSEL = 1, TDRE = 0)
bit12	WSEL: Wait selection bit	• This bit generates an interrupt (INT = 1) either before or after the acknowledgment, and selects whether to have I2C bus waited. • WSEL bit becomes invalid when: (1) An interrupt for the 1st byte*1 is generated (INT = 1) (2) A reserved address is detected (FBT = 1, RSA = 1) (3) NACK respond*2 is detected while transmitting the data when FIFO is used (FE = 1, RACK = 1, ACT = 1) (4) Reception FIFO gets full when using the reception FIFO *1: 1st byte: Indicates the data after the (repetitive) start condition. *2: NACK response: Indicates that SDA in the I2C bus is "H" while the acknowledgment period.
bit11	CNDE: Condition detection interrupt enable bit	Enables the generation of an interrupt when the stop conditions or the repetitive start conditions are detected in master or slave mode (ACT = 1). An interrupt is generated when RSC or SPC bit in the I2C bus status register (IBSR) is "1" and also this bit is "1".
bit10	INTE: Interrupt enable bit	Enables an interrupt for the data transmission/reception and the bus error (INT = 0) when in master or slave mode.
bit9	BER: Bus error detection bit	Indicates that an error is detected on the I2C bus. Set conditions for BER bit: (1) A start or stop condition is detected while transmitting the 1st byte * (2) A (repetitive) start or stop condition is detected at 2nd to 9th bit (acknowledgment) of the data in the 2nd or subsequent byte Reset conditions for BER bit: (1) Write "0" to INT bit when BER = 1 (2) I2C interface disabled (EN = 0) *: 1st byte: Indicates the data after the (repetitive) start condition. Note: Check this bit when the interrupt flag (INT bit) becomes "1" so that you can implement appropriate measures including retransmission (if it is "1", data is not transmitted or received successfully).

Table 15.18-3 Function Description of Each Bit in the I2C Bus Control Register (IBCR) (4 / 4)

Bit name		Function
bit8	INT: Interrupt flag bit	Sets this flag to "1" when in master or slave mode, after 8 or 9 bit (ACK) of the data transmission/reception, or upon a bus error. When other than the bus error, SCL is set to "L" when INT = 1, and SCL released from "L" when INT = 0. Set conditions for INT bit: <8th bit> (1) The reserved address is detected with 1st byte (2) WSEL bit is "1" and an arbitration lost is detected in the 2nd or subsequent byte (3) WSEL bit is "1", and TDRE bit is "1" in the 2nd or subsequent byte during master operation (4) WSEL bit is "1", reception FIFO is disabled, and TDRE bit is "1" in the 2nd or subsequent byte during slave operation (5) WSEL bit is "1", and TDRE bit is "1" in the 2nd or subsequent byte during slave operation <9th bit> (1) An arbitration lost is detected with 1st byte (2) NACK is received when other than the stop condition output setting (writing "0" to MSS bit during master operation) (3) The reserved address is not detected with 1st byte, and TDRE bit is "1" in the transmission direction of master or slave mode (TRX = 1) (4) The reserved address is not detected with 1st byte, and reception FIFO has data in the reception direction of master or slave mode (TRX = 0) when the reception FIFO is enabled (5) The reserved address is not detected with 1st byte, and TDRE bit is "1" in the reception direction of master or slave mode (TRX = 0) when the reception FIFO is disabled (6) WSEL bit is set to "0" and an arbitration lost is detected in the 2nd or subsequent byte (7) WSEL bit is set to "0", and TDRE bit is "1" in the 2nd or subsequent byte during master operation (8) WSEL bit is set to "0", and TDRE bit is "1" in the 2nd or subsequent byte while slave transmission (9) WSEL bit is set to "0" and slave reception is enabled while reception FIFO is disabled. However, an interrupt is not generated at 9th bit in the slave reception with 1st byte that detects the reserved address (10)Reception FIFO gets full in the slave reception when the reception FIFO is enabled <Others> (1) Bus error detection Reset conditions for INT bit: (1) Writing "0" to INT bit (2) Writing "0" to MSS bit when INT = 1 and ACT = 1 (3) Writing "1" to SCC bit when INT = 1 and ACT = 1 Writing "1" to INT bit is ignored. Notes: • If you set EN bit to "0", RDRF and INT bits may become "1" depending on the reception timing. In this case, read the reception data to clear INT bit. • "1" is read by a read-modify-write (RMW) instruction. • If the reception FIFO is enabled, INT bit is not set to "1" even when the reception FIFO gets full by the master reception operation.

15.18.2 Serial Mode Register (SMR)

The serial mode register (SMR) is used to set the operating mode, enable/disable the transmission and reception interrupts.

■ Serial Mode Register (SMR)

Figure 15.18-2 shows the bit configuration of the serial mode register (SMR), and Table 15.18-4 shows the function of each bit.

Figure 15.18-2 Bit Configuration of Serial Mode Register (SMR)

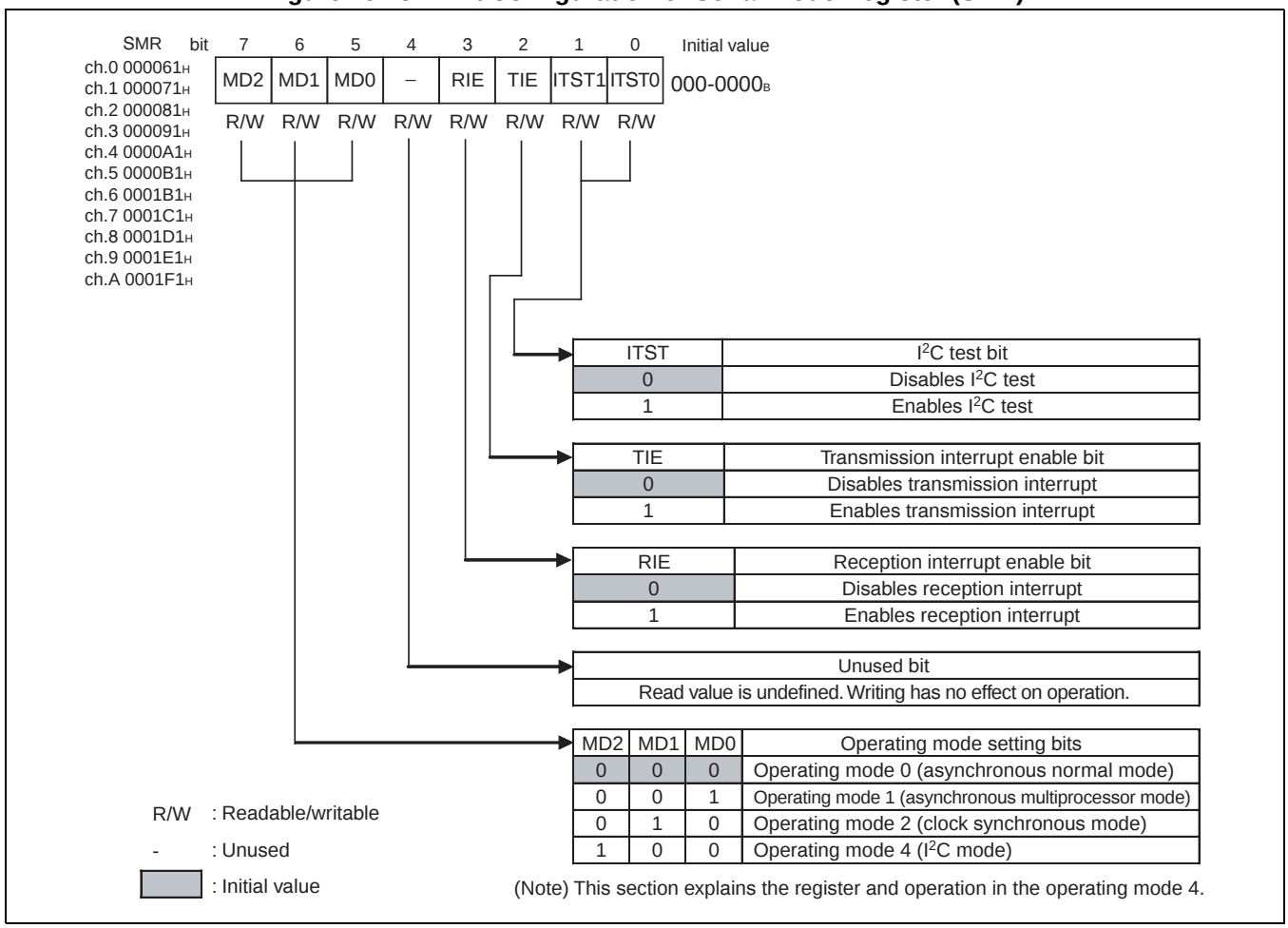


Table 15.18-4 Function Description of Each Bit in the Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operating mode setting bits	Sets the operating mode. "000_B": Sets to the operating mode 0 (asynchronous normal mode) "001_B": Sets to the operating mode 1 (asynchronous multiprocessor mode) "010_B": Sets to the operating mode 2 (clock synchronous mode) "100_B": Sets to the operating mode 4 (I^2C mode) Notes: • Any setting other than those above is disabled. • When switching the operating mode, disables I^2C (ISMK:EN = 0) before switching it. • Set the operating mode, and then set each register.
bit4	Unused bit	When reading : Values are undefined. When writing : No effect.
bit3	RIE: Reception interrupt enable bit	• Enables/disables the output of a reception interrupt request to the CPU. • A reception interrupt request is output when RIE bit and the reception data flag bit (RDRF) are "1" or when the error flag bit (ORE) is set to "1". Note: If INT bit in the I^2C bus control register (IBCR) is used to receive the data, set this bit to "0".
bit2	TIE: Transmission interrupt enable bit	• Enables/disables the output of a transmission interrupt request to the CPU. • A transmission interrupt request is output when TIE bit and the TDRE bit are "1". Note: If INT bit in the I^2C bus control register (IBCR) is used to transmit the data, set this bit to "0".
bit1, bit0	ITST1, ITST0: I^2C test bits	These are I^2C test bits. Be sure to set these bits to "0". Note: If you set these bits to "1", the test for I^2C will be executed.

Note:

Set the operating mode first because the other registers will be initialized once the operating mode has been changed. However, when IBCR and SMR are written at the same time by 16-bit writing, the written contents will be reflected on IBCR.

15.18.3 Serial Status Register (SSR)

The serial status register (SSR) is used to check transmission/reception status.

■ Serial Status Register (SSR)

Figure 15.18-3 shows the bit configuration of the serial status register (SSR), and Table 15.18-5 shows the function of each bit.

Figure 15.18-3 Bit Configuration of Serial Status Register (SSR)

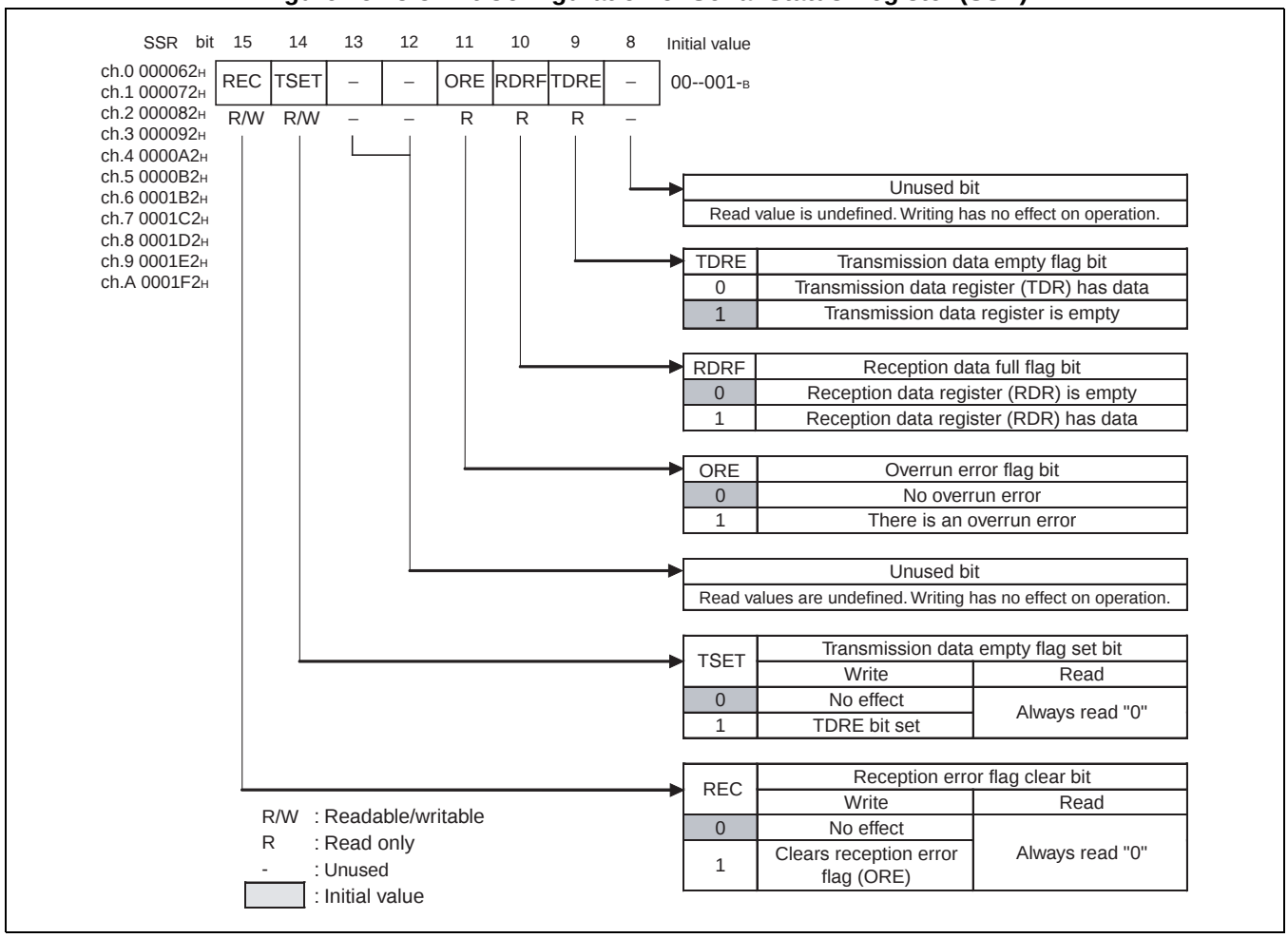


Table 15.18-5 Function Description of Each Bit in the Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	Clears ORE bit in the serial status register (SSR). • Writing "1" to this bit clears ORE bit. • Writing "0" has no effect. When reading, "0" is always read.
bit14	TSET: Transmission buffer empty flag set bit	Sets TDRE bit in the serial status register (SSR). • Writing "1" to this bit sets TDRE bit. • Writing "0" has no effect. When reading, "0" is always read.
bit13, bit12	Unused bits	When reading : Values are undefined. When writing : No effect.
bit11	ORE: Overrun error flag bit	• When an overrun error occurs while receiving, this bit is set to "1". When you write "1" to REC bit in the serial status register (SSR), this bit is cleared. • A reception interrupt request is output when ORE bit and RIE bit are "1". • Data in the reception data register (RDR) is invalid if this flag is set. • When this bit is set while using the reception FIFO, the reception data will not be stored into the reception FIFO.
bit10	RDRF: Reception data full flag bit	• Indicates the status of the reception data register (RDR). • A reception interrupt request is output when the reception data flag (RDRF) bit and RIE bit are "1". • When the reception data is loaded to RDR, this bit is set to "1". If the reception data register (RDR) is read, this bit is cleared to "0". • This bit is set at SCL falling timing of 8th bit of the data. • This bit is also set by NACK response*. • When using the reception FIFO, RDRF is set to "1" if the reception FIFO has received a predefined number of data. • When using the reception FIFO, this bit is cleared to "0" if the reception FIFO gets empty. • When using the reception FIFO, if the reception idle state continues over 8 clocks of the reception baud rate clock (because the reception FIFO has not received a predefined number of data and some data still remains in the reception FIFO), RDRF is set to "1" when BER bit is "0". If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. *: NACK response: Indicates that SDA in the I2C bus is "H" while the acknowledgment period.

Table 15.18-5 Function Description of Each Bit in the Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit9	TDRE: Transmission data empty flag bit	• Indicates the status of the transmission data register (TDR). • A transmission interrupt request is output when TIE bit and the TDRE bit are "1". • When you write a transmission data to TDR, this bit becomes "0" to indicate that there is some valid data. When the data is loaded to the transmission shift register to start transmission, this bit becomes "1" to indicate that there is no valid data in TDR. • This bit is set when "1" is written to TSET bit in the serial status register (SSR). When an arbitration lost or bus error is detected, use this bit to set TDRE bit to "1".
bit8	Unused bit	When reading : Values is undefined. When writing : No effect.

15.18.4 I2C Bus Status Register (IBSR)

I2C bus status register (IBSR) indicates that a repetitive start, acknowledgment, data direction, arbitration lost, stop condition, I2C bus state, and bus error are detected.

■ I2C Bus Status Register (IBSR)

Figure 15.18-4 shows the bit configuration of the I2C bus status register (IBSR), and Table 15.18-6 shows the function of each bit.

Figure 15.18-4 Bit Configuration of I2C Bus Status Register (IBSR)

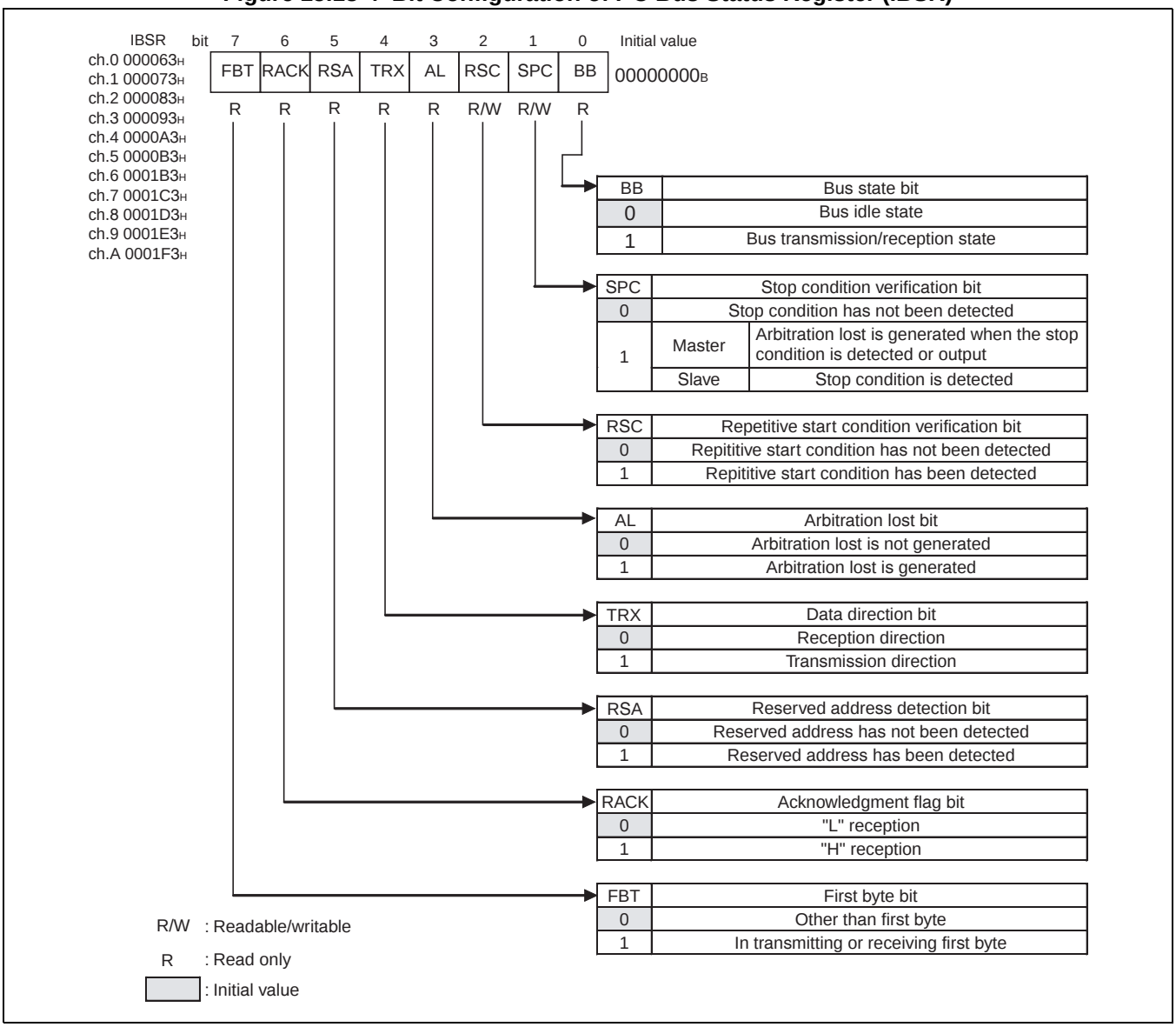


Table 15.18-6 Function Description of Each Bit in the I2C Bus Status Register (IBSR) (1 / 3)

Bit name		Function
bit7	FBT: First byte bit	Indicates First byte. Set conditions for FBT bit: (1) (Repetitive) start condition is detected Clear conditions for FBT bit: (1) Transmission and reception of 2nd byte (2) Stop condition is detected (3) I2C interface is disabled (EN bit = 0) (4) Bus error is detected (BER bit = 1)
bit6	RACK: Acknowledgment flag bit	The acknowledgment received with First byte or when in master or slave mode is indicated in this bit. Update conditions for RACK bit (1) An acknowledgment when 1st byte (2) An acknowledgment when in master or slave mode Clear conditions for RACK bit (RACK = 0) (1) (Repetitive) start condition is detected (2) I2C interface is disabled (EN bit = 0) (3) Bus error is detected (BER bit = 1)
bit5	RSA: Reserved address detection bit	Indicates that a reserved address is detected. Set conditions for RSA bit (RSA = 1) (1) 1st byte is ($0000xxxx_B$) or ($1111xxxx_B$). "x" represents "0" or "1". Reset conditions for RSA bit (RSA = 0) (1) (Repetitive) start condition is detected (2) Stop condition is detected (3) I2C interface is disabled (EN bit = 0) (4) Bus error is detected (BER bit = 1) When RSA bit becomes "1" with 1st byte, the interrupt flag (INT) is set to "1" to specify SCL to "L" at the falling of SCL of the 8th bit in the 1st byte, regardless of whether FIFO is enabled or disabled. If you want to read the reception data to operate as a slave at this point, set ACKE to "1" to clear the interrupt flag (INT) to "0". Then, if TRX bit is "0", the data is received as a slave. If you do not want to have the data received, set ACKE bit to "0". Subsequent data will not be received. Notes: • If ACKE bit is set to "0" while transmitting the data, setting ACKE bit to "1" is disabled until a stop or a repetitive start condition is detected. • Upon the interrupt by detecting the reserved address, the verification of the slave transmission causes an ACK response when the reception FIFO is enabled, so disable the reception FIFO and set ACKE bit to "0".

Table 15.18-6 Function Description of Each Bit in the I2C Bus Status Register (IBSR) (2 / 3)

Bit name		Function
bit4	TRX: Data direction bit	Indicates the direction of the data. Set conditions for TRX bit: (1) Transmits a (repetitive) start condition in master mode (2) When 8th bit in 1st byte is "1" in slave mode (transmission direction as a slave) Reset conditions for TRX bit: (1) Arbitration lost is generated (AL = 1) (2) When 8th bit in 1st byte is "0" in slave mode (reception direction as a slave) (3) When 8th bit in 1st byte is "1" in master mode (reception direction as a master) (4) Stop condition is detected (5) (Repetitive) start condition is detected in other than master mode (6) I2C interface is disabled (EN bit = 0) (7) Bus error is detected (BER bit = 1)
bit3	AL: Arbitration lost bit	Indicates the arbitration lost. Set conditions for AL bit: (1) The output data and received data are different when in master mode (2) MSS bit is set to "1" but, the register is operating as a slave (3) A repetitive start condition is detected at 1st bit of the data in the 2nd or subsequent byte when in master mode (4) A stop condition is detected at 1st bit of the data in the 2nd or subsequent byte when in master mode (5) Trying to generate a (repetitive) start condition but cannot do so when in master mode (6) Trying to generate a stop condition but cannot do so when in master mode Reset conditions for AL bit: (1) Writing "1" to MSS bit (2) Writing "0" to INT bit (3) Writing "0" to SPC bit when AL bit = 1 and SPC bit = 1 (4) I2C interface is disabled (EN bit = 0) (5) Bus error is detected (BER bit = 1)
bit2	RSC: Repetitive start condition verification bit	Indicates that a repetitive start condition is detected when in master or slave mode. Set conditions for RSC bit: (1) A repetitive start condition is detected after the acknowledgment during the slave or master mode operation Reset conditions for RSC bit: (1) Writing "0" to RSC bit (2) Writing "1" to MSS bit (3) I2C interface is disabled (EN bit = 0) Writing "1" to this bit is ignored. Notes: • While the reception operation is in progress as slave mode by detecting the reserved address, if the ACK response is not performed, the slave mode is terminated so that this bit will not be set to "1" even when next repetitive start condition is detected. • "1" is read by a read-modify-write (RMW) instruction.

Table 15.18-6 Function Description of Each Bit in the I2C Bus Status Register (IBSR) (3 / 3)

Bit name		Function
bit1	SPC: Stop condition verification bit	Indicates that a stop condition is detected when in master or slave mode. Set conditions for SPC bit: (1) A stop condition is detected during the slave or master mode operation (2) An arbitration lost is generated by the operation to generate a stop condition when in master mode Reset conditions for SPC bit: (1) Writing "0" to this bit (2) Writing "1" to MSS bit (3) I2C interface is disabled (EN bit = 0) Writing "1" to this bit is ignored. Notes: • While the reception operation is in progress as slave mode by detecting the reserved address, if the ACK response is not performed, the slave mode is terminated so that this bit will not be set to "1" even when next stop condition is detected. • "1" is read by a read-modify-write (RMW) instruction.
bit0	BB: Bus state bit	Indicates the state of the bus. Set conditions for BB bit: (1) "L" is detected in SDA or SCL in the I2C bus Reset conditions for BB bit: (1) Stop condition is detected (2) I2C interface is disabled (EN bit = 0) (3) Bus error is detected (BER bit = 1)

15.18.5 Reception/Transmission Data Registers (RDR/TDR)

Reception/transmission data registers are arranged in the same address. When reading, it functions as the reception data register and when writing, it functions as the transmission data register.

■ Reception Data Register (RDR)

Figure 15.18-5 shows the bit configuration of the reception data register (RDR).

Figure 15.18-5 Bit Configuration of the Reception Data Register (RDR)

RDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R	R	R	R	R	R	R	R	

R: Read only

Reception data register (RDR) is a data buffer register for serial data reception.

- A serial data signal transmitted to the serial data line (SDA pin) is converted at the shift register and then stored in this reception data register (RDR).
- When 1st byte* is received, the lowest bit (RDR:D0) becomes the data direction bit.
- When the received data is stored in the reception data register (RDR), the reception data full flag bit (SSR:RDRF) is set to "1".
- If the reception data register (RDR) is read, the reception data full flag bit (SSR:RDRF) is automatically cleared to "0".

*: 1st byte: Indicates the data after the (repetitive) start condition.

Notes:

- When using the reception FIFO, RDRF is set to "1" if the reception FIFO has received a predefined number of data.
- When using the reception FIFO, RDRF is cleared to "0" if the reception FIFO gets empty.

■ Transmission Data Register (TDR)

Figure 15.18-6 shows the bit configuration of the transmission data register (TDR).

Figure 15.18-6 Bit Configuration of the Transmission Data Register (TDR)

TDR

ch.0 000064_H ch.1 000074_H
ch.2 000084_H ch.3 000094_H
ch.4 $0000A4_H$ ch.5 $0000B4_H$
ch.6 $0001B4_H$ ch.7 $0001C4_H$
ch.8 $0001D4_H$ ch.9 $0001E4_H$
ch.A $0001F4_H$

bit15.....	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	11111111_B
	W	W	W	W	W	W	W	W	

W: Write only

Transmission data register (TDR) is a data buffer register for serial data transmission.

- Data is output to the serial data line (SDA pin) by MSB first of the value in the transmission data register (TDR).
- When 1st byte is transmitted, the lowest bit (TDR:D0) becomes the data direction bit.
- A transmission data empty flag (SSR:TDRE) is cleared to "0" when the transmission data is written into the transmission data register (TDR).
- Once the data is transferred to the transmission shift register, the transmission data empty flag (SSR:TDRE) bit is set to "1".
- Write next transmission data when:
  - Interrupt flag (INT bit) is "1"
  - Bus error is not generated (BER bit = 0)
  - Acknowledgment performs ACK response (receive "0" as an acknowledgment)
- If the transmission FIFO is disabled, you cannot write the transmission data to the transmission data register (TDR) when the data empty flag (SSR:TDRE) is "0".
- If the transmission FIFO is used, you can write the transmission data up to the size of the transmission FIFO even when the data empty flag (SSR:TDRE) is "0".

Note:

The transmission data register (TDR) is a write only register and the reception data register (RDR) is a read only register. These registers are located at the same address, so the read value is different from the write value. Therefore an instruction that operates read-modify-write (RMW) instruction, such as INC/DEC, cannot be used.

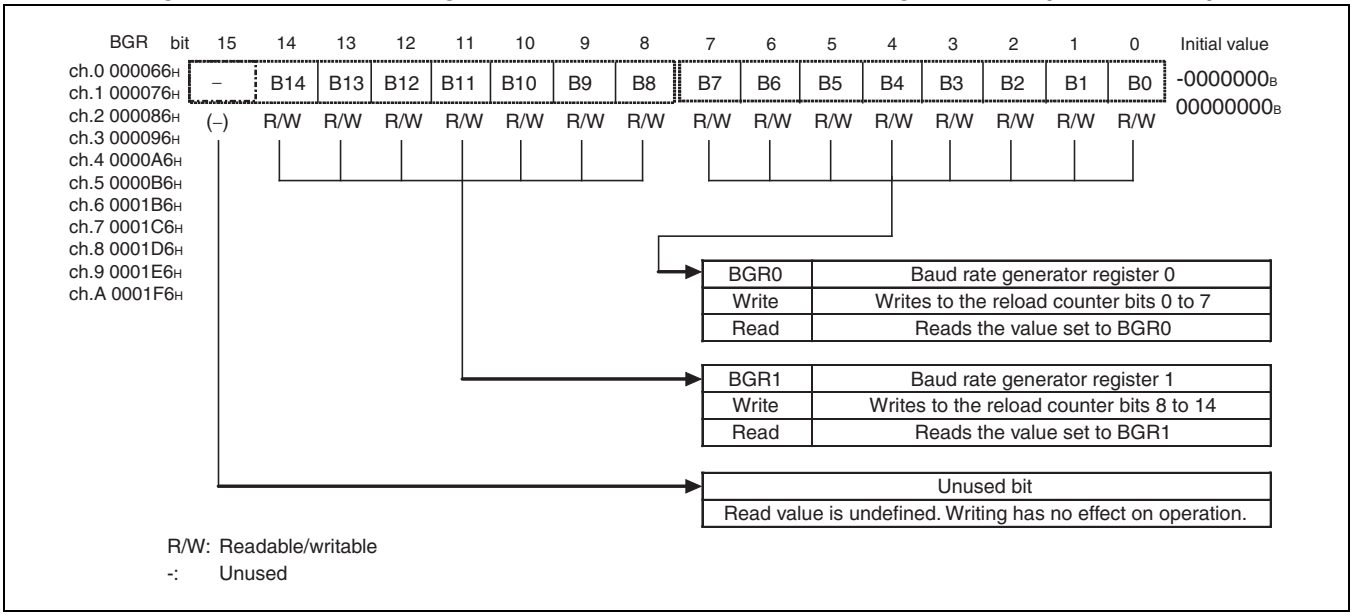
15.18.6 Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

The baud rate generator registers 1, 0 (BGR1/BGR0) sets the division ratio for the serial clock.

■ Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)

Figure 15.18-7 shows the bit configuration of the baud rate generator registers 1, 0 (BGR1/BGR0).

Figure 15.18-7 Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1/BGR0)



The baud rate generator registers (BGR) set the division ratio for the serial clock.

BGR1 (supports the upper bits) and BGR0 (supports the lower bits) can write the reload value to count and read the value set to BGR1/BGR0.

The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).

Notes:

- Write to the baud rate generator registers 1, 0 (BGR1/BGR0) by 16-bit access.
- Set the baud rate generator registers when EN bit in the ISMK register is "0".
- Baud rate should be set regardless of master or slave mode.
- In operating mode 4 (I^2C mode), the machine clock should be used for 8 MHz or more and setting over 400 kbps to the baud rate generator is disabled.

MB91313A Series

15.18.7 7-bit Slave Address Mask Register (ISMK)

7-bit slave address mask register (ISMK) is used to compare or set each bit in the slave address.

■ 7-bit Slave Address Mask Register (ISMK)

Figure 15.18-8 shows the bit configuration of the 7-bit slave address register (ISMK), and Table 15.18-7 shows the function of each bit.

Figure 15.18-8 Bit Configuration of 7-bit Slave Mask Register (ISMK)

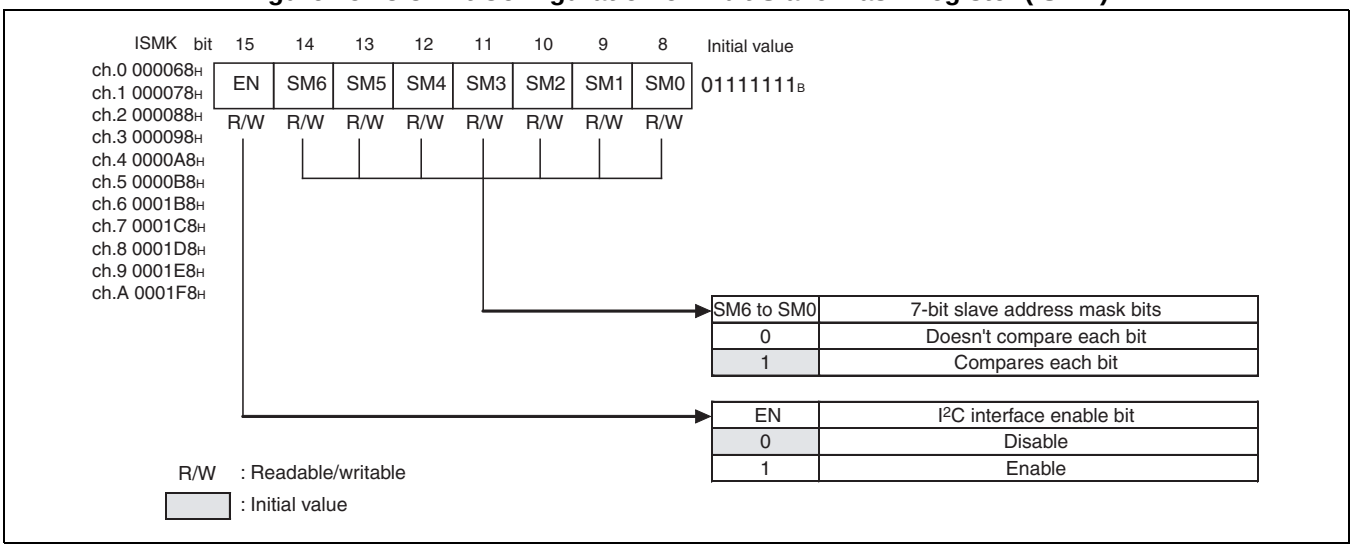


Table 15.18-7 Function Description of Each Bit in the 7-bit Slave Mask Register (ISMK)

Bit name		Function
bit15	EN: I2C interface enable bit	Enables/disables the operation of the I2C interface. When "0" is set: Disables the operation of the I2C interface. When "1" is set: Enables the operation of the I2C interface. Notes: • This bit is not cleared to "0" even if BER bit in the I2C bus status register (IBSR) is set to "1". • Set the baud rate generator when this bit is set to "0". • Set the 7-bit slave address and 7-bit slave mask registers when this bit is set to "0". • If EN bit is set to "0" while transmitting, a pulse can be generated in SDA/SCL in the I2C bus. • If FIFO is enabled, disable the FIFO then write "0" to EN bit.
bit14 to bit8	SM6 to SM0: Slave address mask bits	Specifies whether 7-bit slave address and the received address should be the subject of comparison or not. The bit set to "1": Compares The bit set to "0": Processed as if they match Note: Set this register when EN bit is "0".

15.18.8 7-bit Slave Address Register (ISBA)

7-bit slave address register (ISBA) is used to set the slave address.

■ 7-bit Slave Address Register (ISBA)

Figure 15.18-9 shows the bit configuration of the 7-bit slave address register (ISBA), and Table 15.18-8 shows the function of each bit.

Figure 15.18-9 Bit Configuration of 7-bit Slave Address Register (ISBA)

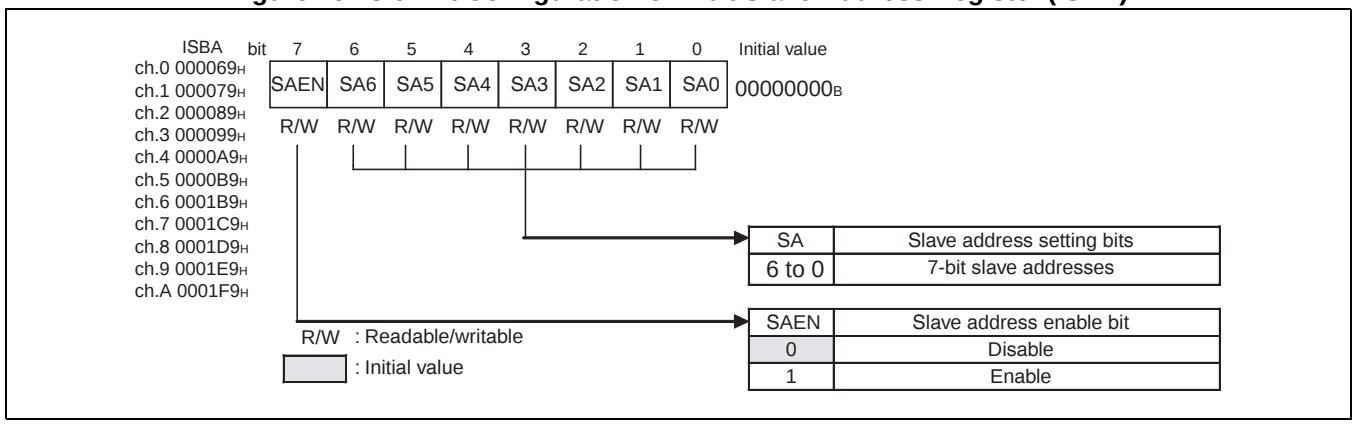


Table 15.18-8 Function Description of Each Bit in the 7-bit Slave Address Register (ISBA)

Bit name		Function
bit7	SAEN: Slave address enable bit	Enables the detection of the slave address. When "0" is set: Does not detect the slave address. When "1" is set: Compares between the setting of ISBA and ISMK and 1st byte of the received data.
bit6 to bit0	SA6 to SA0: 7-bit slave addresses	• If the detection of the slave address is enabled (SAEN = 1), 7-bit slave address register (ISBA) compares with the 7-bit of data received after the (repetitive) start condition is detected. If they match each other, this register operates as slave mode and outputs an ACK. At this point, the received slave address is set to this register (ACK is not output when SAEN = 0). • The address bit of which ISMK register is set to "0" is not the subject of comparison. Notes: • Setting the reserved address is disabled. • Set this register when EN bit in the ISMK register is "0".

15.18.9 FIFO Control Register 1 (FCR1)

FIFO control register 1 (FCR1) sets the test for FIFO, specifies the transmission/ reception FIFO, enables the transmission FIFO interrupt, and controls the interrupt flag.

■ Bit Configuration of FIFO Control Register 1 (FCR1)

Figure 15.18-10 shows the bit configuration of the FIFO control register 1 (FCR1), and Table 15.18-9 shows the function of each bit.

Figure 15.18-10 Bit Configuration of FIFO Control Register 1 (FCR1)

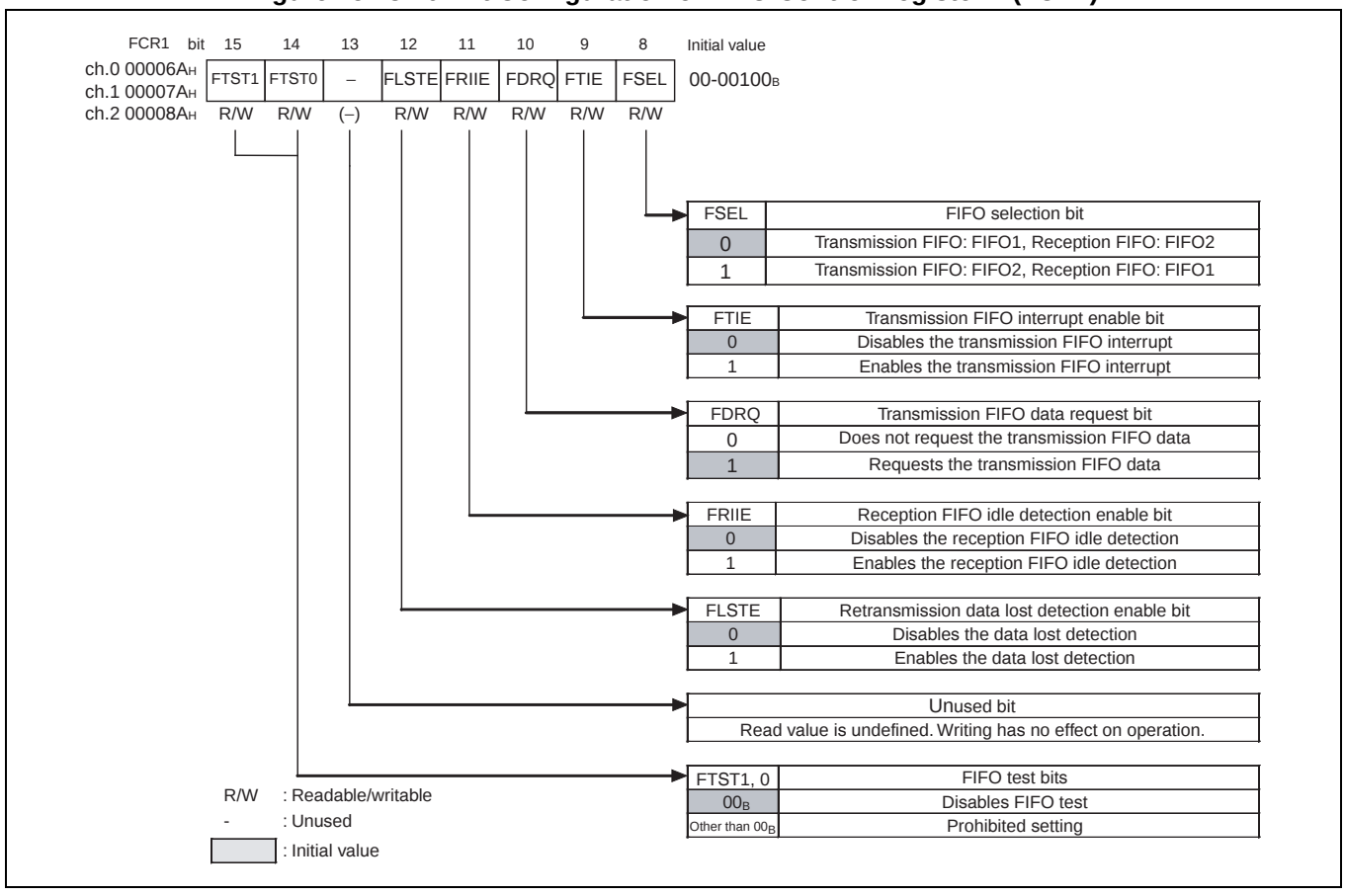


Table 15.18-9 Function Description of Each Bit in the FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	FTST1, FTST0: FIFO test bits	These are the test bits for FIFO. Be sure to set these bits to "0". Note: If you set these bits to "1", the test for FIFO will be executed.
bit13	Unused bit	When reading : Values is undefined. When writing : No effect.
bit12	FLSTE: Retransmission data lost detection enable bit	Enables data lost detection. When "0" is set: Disables data lost detection. When "1" is set: Enables data lost detection. Note: When you set this bit to "1", set FSET bit to "1" before doing so.
bit11	FRIIE: Reception FIFO idle detection enable bit Unused bit	Specifies whether to detect the reception idle state that continues over 8-bit time with the reception FIFO holding valid data. If the reception interrupt is enabled (SMR:RIE = 1), a reception interrupt occurs when the reception idle state is detected. When "0" is set: Disables the reception idle state detection. When "1" is set: Enables the reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a data request bit for the transmission FIFO. When this bit is set to "1", it indicates that the transmission data is being requested. A transmission FIFO interrupt request is output when this bit is "1" and the transmission interrupt is enabled (FTIE = 1). FDRQ set condition • FBYTE (for transmission) = 0 (transmission FIFO is empty) • Reset the transmission FIFO FDRQ reset condition • Writing "0" to this bit • The transmission FIFO gets full Notes: • When FBYTE (for transmission) = 0, writing "0" to this bit is disabled. • When this bit is set to "0", change of FSEL bit is disabled. • Setting "1" to this bit has no effect on the operations. • "1" is read by a read-modify-write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is an interrupt enable bit for the transmission FIFO. If this bit is set to "1", an interrupt occurs when FDRQ bit is "1".
bit8	FSEL: FIFO selection bit	Selects the transmission/reception FIFO. When "0" is set: Assigns the transmission FIFO:FIFO1 and the reception FIFO:FIFO2. When "1" is set: Assigns the transmission FIFO:FIFO2 and the reception FIFO:FIFO1. Notes: • This bit cannot be cleared by the FIFO reset (FCL2, FCL1 = 1). • When you want to change this bit, disable FIFO operation (FE2, FE1 = 0) and any transmission/reception (TXE = RXE = 0) first.

15.18.10 FIFO Control Register 0 (FCR0)

FIFO control register 0 (FCR0) enables/disables FIFO operation, performs FIFO reset, saves the read pointer, and sets the retransmission.

■ Bit Configuration of FIFO Control Register 0 (FCR0)

Figure 15.18-11 shows the bit configuration of the FIFO control register 0 (FCR0), and Table 15.18-10 shows the function of each bit.

Figure 15.18-11 Bit Configuration of FIFO Control Register 0 (FCR0)

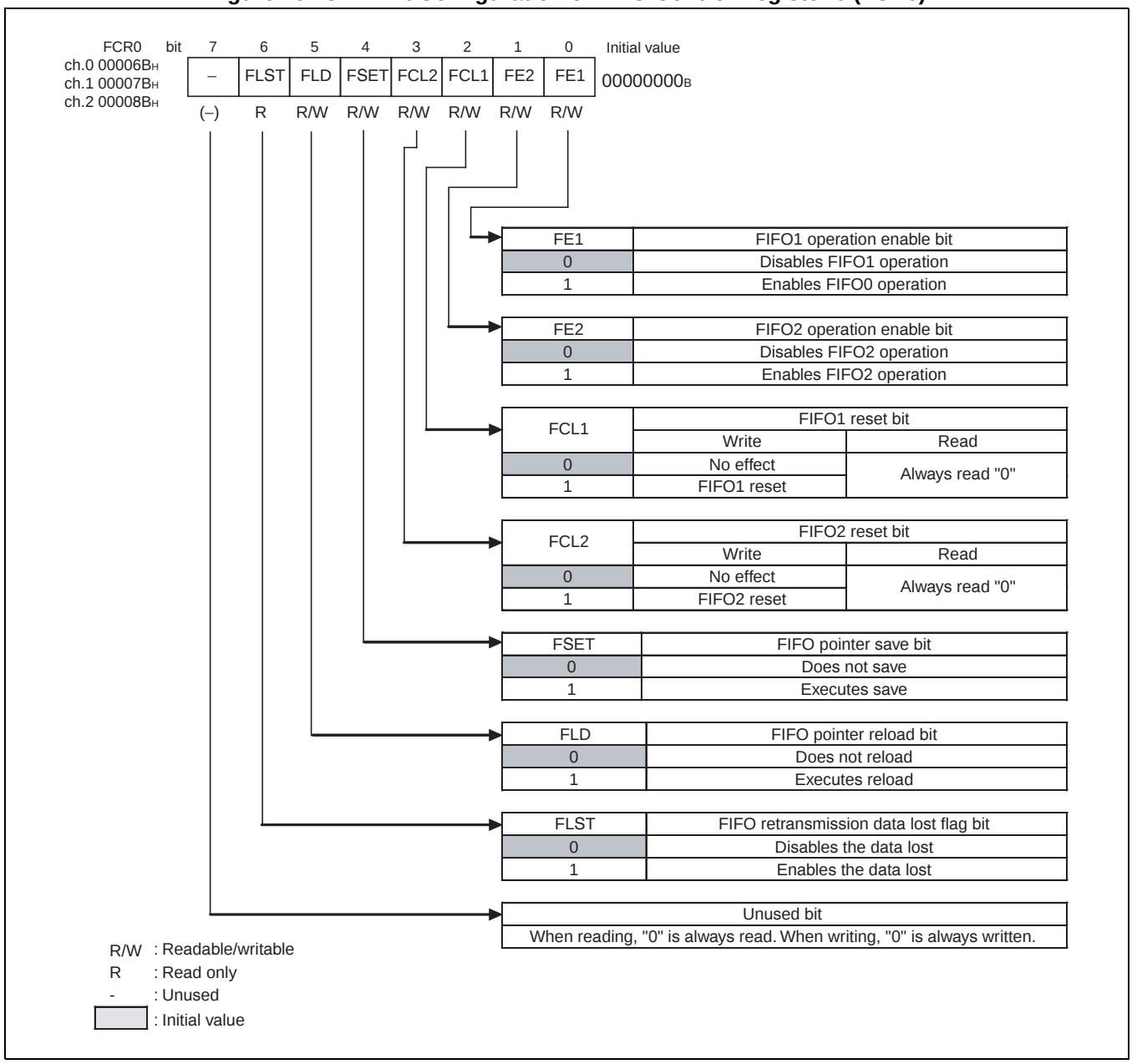


Table 15.18-10 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Unused bit	When reading : "0" is always read. When writing : Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	Indicates that retransmission data of the transmission FIFO has been lost. FLST set condition • Writing to FIFO when FLSTE bit in the FIFO control register 1 (FCR1) is "1" and also the write pointer of the transmission FIFO and the read pointer saved by FSET bit match each other. FLST reset condition • FIFO reset (writing "1" into FCL) • When writing "1" to FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer saved with FSET bit, therefore FLD bit cannot set the retransmission when an error occurs. If you retransmit with this bit set to "1", perform FIFO reset and write the data again into FIFO.
bit5	FLD: FIFO pointer reload bit	Reloads the data saved to the transmission FIFO by FSET bit to the read pointer. This bit is used for the retransmission due to a communication error. When a retransmission setting is completed, this bit becomes "0". Notes: • As long as this bit is set to "1", this bit is reloading to the read pointer. Therefore, do not write anything other than FIFO reset. • Setting this bit to "1" is disabled while transmitting or being in FIFO enabled state. • Set TIE bit to "0", and then write "1" into this bit. After you enabled the transmission FIFO, set TIE bit to "1".
bit4	FSET: FIFO pointer save bit	Saves the read pointer of the transmission FIFO. Once you save the read pointer before communication, when FLST bit is "0" it is possible to retransmit in the case a communication error. When "1" is set: Retains the current value set to the read pointer. When "0" is set: No effect. Note: Set this bit to "1" when the number of bytes for transmission (FBYTE) indicates "0".
bit3	FCL2: FIFO2 reset bit	Resets FIFO2. If you set this bit to "1", the internal state of FIFO2 is initialized. Only the FLST bit in the FIFO control register 0 (FCR0) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables FIFO2 before performing FIFO2 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE2 register becomes "0".
bit2	FCL1: FIFO1 reset bit	Resets FIFO1. If you set this bit to "1", the internal state of FIFO1 is initialized. Only the FLST bit in the FIFO control register 0 (FCR0) is initialized and the other bits in the FCR1/FCR0 registers remain unchanged. Notes: • Disables FIFO1 before performing FIFO1 reset. • Set the transmission FIFO interrupt enable bit to "0" first. • The number of the valid data in the FBYTE1 register becomes "0".

Table 15.18-10 Function Description of Each Bit in the FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit1	FE2: FIFO2 operation enable bit	Enables/disables the operation of FIFO2. • When using FIFO2, set this bit to "1". • When specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • If using as the transmission FIFO, set "1" or "0" to this bit when the transmission data is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception data is empty (RDRF = 0). • Even if FIFO2 is disabled, the state of FIFO2 is still retained. Notes: • Enable/disable the operation when BB bit is "0" or INT bit is "1". • When specified as a reception FIFO, the reserved address is detected, and it operates as a slave transmission, set this bit to "0" with the interrupt generated by detecting the reserved address and ACKE to "0". • If the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". • To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO2 contains data.
bit0	FE1: FIFO1 operation enable bit	Enables/disables the operation of FIFO1. • When using FIFO1, set this bit to "1". • When specified as a reception FIFO by FSEL bit, a reception error clears this bit to "0". You cannot set this bit to "1" until the reception error is cleared. • If using as the transmission FIFO, set "1" or "0" to this bit when the transmission data is empty (TDRE = 1). If using as the reception FIFO, set "1" or "0" to this bit when the reception data is empty (RDRF = 0). • Even if FIFO1 is disabled, the state of FIFO1 is still retained. Notes: • Enable/disable the operation when BB bit is "0" or INT bit is "1". • When specified as a reception FIFO, the reserved address is detected, and it operates as a slave transmission, set this bit to "0" with the interrupt generated by detecting the reserved address and ACKE to "0". • If the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". • To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO1 contains data.

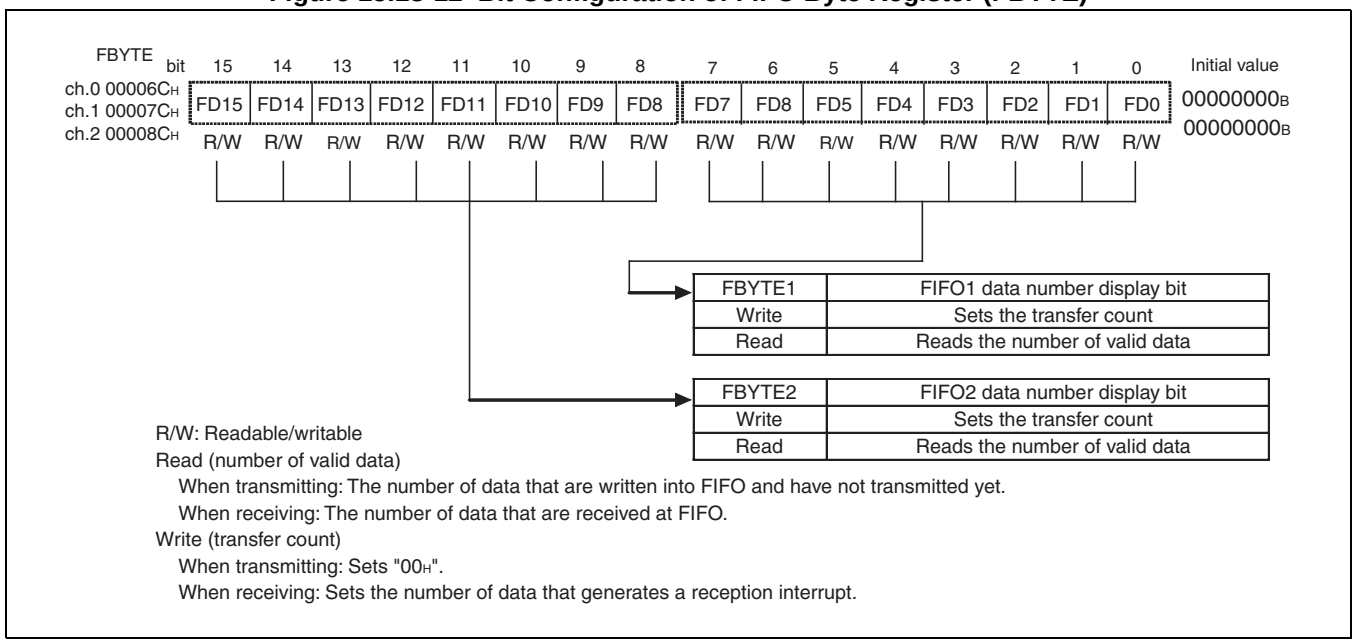
15.18.11 FIFO Byte Register (FBYTE)

FIFO byte register (FBYTE) indicates the number of valid data for FIFO. In addition, it can specify whether to generate a reception interrupt when the predefined number of data has been received at the reception FIFO.

■ Bit Configuration of FIFO Byte Register (FBYTE)

Figure 15.18-12 shows the bit configuration of the FIFO byte register (FBYTE).

Figure 15.18-12 Bit Configuration of FIFO Byte Register (FBYTE)



FIFO byte register (FBYTE) indicates the number of valid data of FIFO. The number varies as follows, depending on the setting of FSEL bit in the FCR1 register.

Table 15.18-11 The Number of Data Displayed

FSEL	FIFO selection	The number of data displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value of the transfer count of the FIFO byte register (FBYTE) is "08_H".
- Set the number of data to generate a reception interrupt flag to the FBYTE in the reception FIFO. An interrupt flag (RDRF) is set to "1" when the defined transfer count matches with the number of data displayed in the FIFO byte register (FBYTE).

- If the reception FIFO idle detection enable bit (FRIIE) is "1" and the number of data that exists in the reception FIFO has not reached the transfer count, the interrupt flag (RDRF) is set to "1" when the reception idle state continues over 8 clocks of the baud rate clock. If you read the RDR while counting 8 clocks, the counter is reset to "0" and start counting 8 clocks all over again. The counter is reset to "0" when the reception FIFO is disabled. When the reception FIFO that has still some data is enabled, it starts counting all over again.
- To receive the data with the master operation (master reception), set TIE bit to "0", specify the number of data to receive in the FIFO byte register (FBYTE) of the transmission FIFO, and write "0" to FDRQ bit. The specified number of data for SCL clocks are output, and then INT bit becomes "1". If you want to set TIE bit to "1", wait until FDRQ bit becomes "1".

Notes:

- In the master operation, set "00_H" to FBYTE of the transmission FIFO except that data is received.
- When receiving data with the master operation, set the number of transmission data when the transmission FIFO is empty and also TIE bit is set to "0".
- If you want to disable the I^2C interface (EN = 0) while receiving the data with the master operation, disable the transmission and reception FIFO before disabling the interface.
- Set a data more than "1" to the FBYTE in the reception FIFO.
- Disable the transmission and reception before you change the setting.
- You cannot use any read-modify-write (RMW) instruction to this register.
- The setting that exceeds the FIFO size is disabled.

15.19 Interrupts of the I2C Interface

An interrupt request of the I2C interface can be generated by the following sources:

- **After 1st byte/data is transmitted and received**
- **Stop conditions**
- **Repetitive start conditions**
- **FIFO transmission data request**
- **FIFO reception data is completed**

■ Interrupts of the I2C Interface

Table 15.19-1 shows the I2C interface interrupt control bit and the interrupt source.

Table 15.19-1 I2C Interface Interrupt Control Bit and Interrupt Source

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	How to clear the interrupt request flag
Reception	INT	IBCR	After 1st byte is transmitted and received*1	IBCR:INTE	Write "0" to the interrupt flag bit (IBCR:INT)
			After the data is transmitted and received*1		
			Bus error has been detected		
			Arbitration lost has been detected		
			The value set to FBYTE is received		Read the reception data (RDR) until the reception FIFO gets empty, and then write "0" to the interrupt flag bit (IBCR:INT)
			The reception idle state that continues over 8-bit time has been detected with the reception FIFO holding valid data when FRIIE bit is "1"		
	RDRF	SSR	Reserved address has been detected	SMR:RIE	Read reception data (RDR)
			After the data is received		
			The value set to FBYTE is received		Read reception data (RDR) until the reception FIFO gets empty
	ORE	SSR	Overrun error		Write "1" to the reception error flag bit (SSR:REC)
	SPC	IBSR	Stop conditions	IBCR:CNDE	Write "0" to the stop condition detection bit
	RSC	IBSR	Repetitive start conditions		Write "0" to the repetitive start detection flag bit (IBSR:RSC)
Transmis-sion	TDRE	SSR	Transmission register is empty	SMR:TIE	Write to the transmission data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)*2
			"1" is written to the transmission buffer empty flag set bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Write "0" to the FIFO transmission data request bit, or the transmission FIFO is full

*1: If normal data can be transmitted and received and TDRE bit is "0", no interrupt is generated. This is because the DMA transfer is supported.
If you want INT flag to be generated on the data transmission/reception, TDRE bit must be "1" before the timing when the INT flag is set.

*2: Set TIE bit to "1" after TDRE bit has become "0".

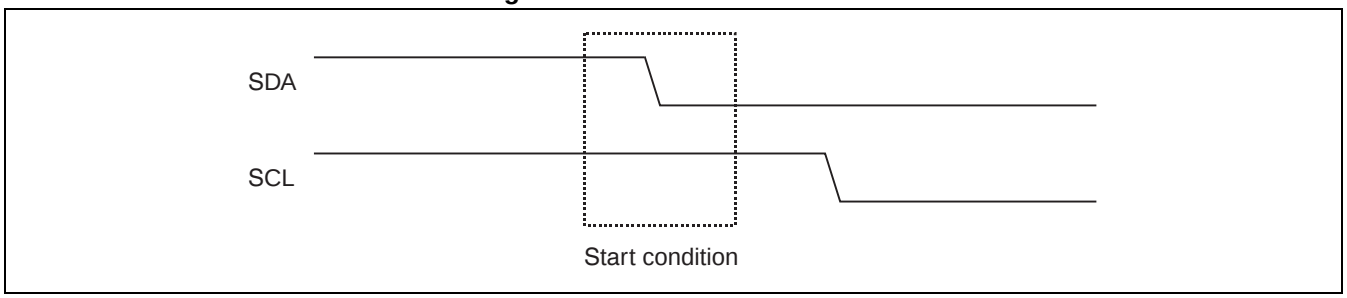
15.20 Operations of I2C Interface Communication

I2C interface uses 2 bidirectional bus lines, the serial data line (SDA) and the serial clock line (SCL), for communication.

■ I2C Bus Start Condition

The figure below shows the start condition for I2C bus.

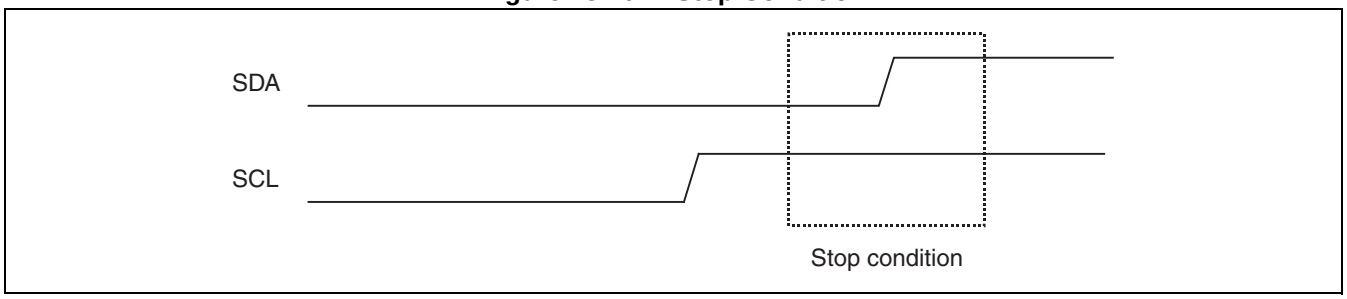
Figure 15.20-1 Start Condition



■ I2C Bus Stop Condition

The figure below shows the stop condition for I2C bus.

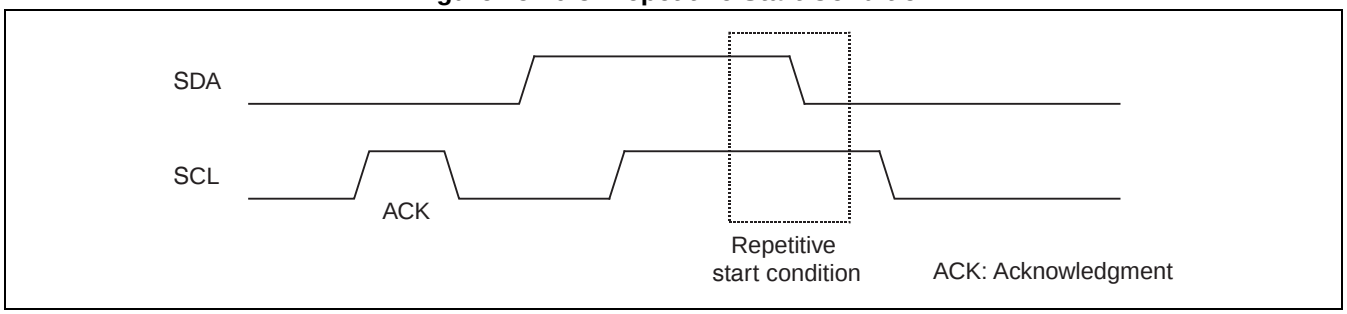
Figure 15.20-2 Stop Condition



■ I2C Bus Repetitive Start Condition

The figure below shows the repetitive start condition for I2C bus.

Figure 15.20-3 Repetitive Start Condition



15.20.1 Master Mode

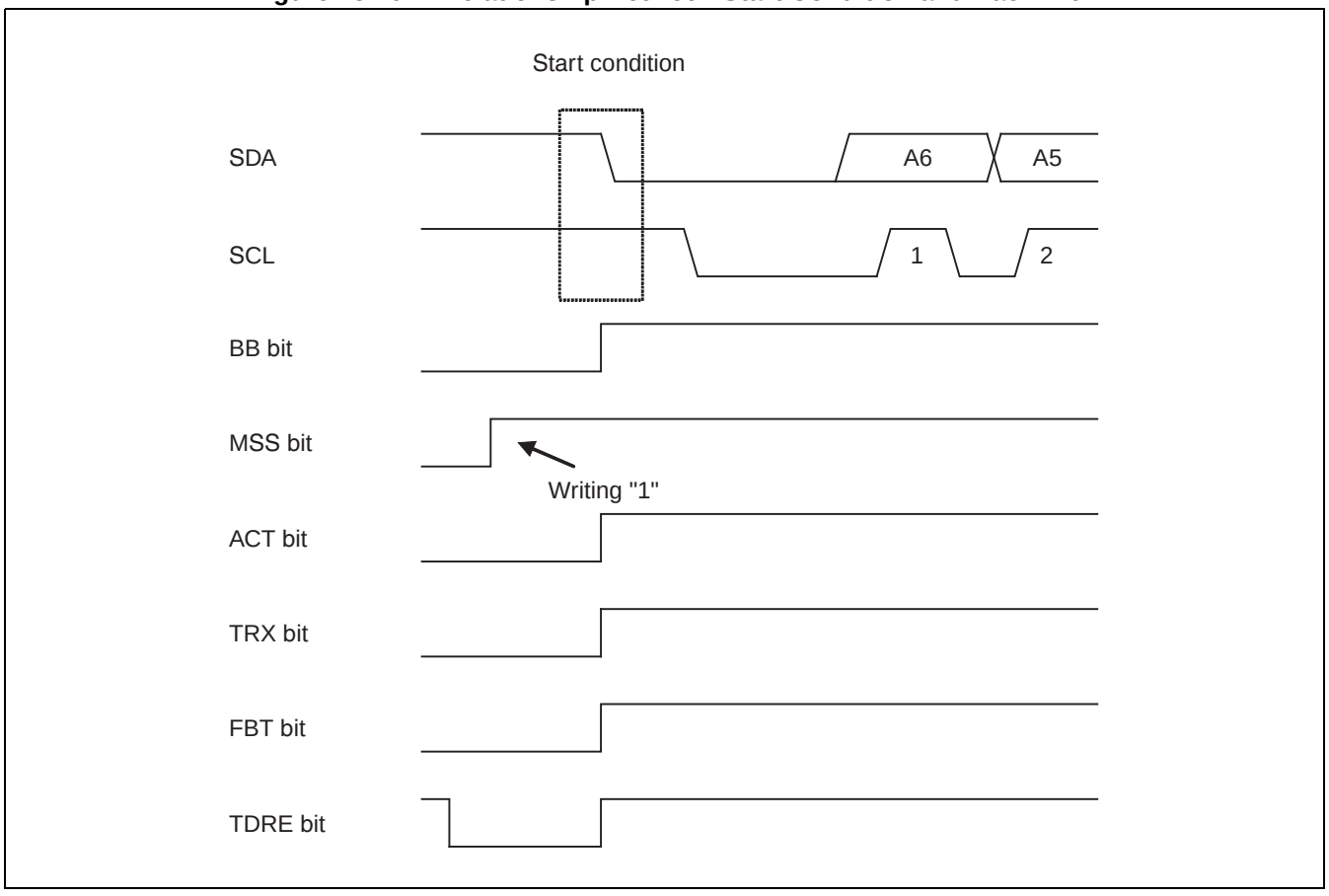
Master mode gets I2C bus to generate a start condition and outputs the clock to I2C bus. If I2C bus is in the idle state (SCL = "H", SDA =" H"), the master mode is selected when "1" is set to MSS bit in the IBCR register, and ACT bit in the IBCR register becomes "1".

■ Generation of the Start Condition

A start condition is output under the following condition:

When SDA = "H", SCL = "H", EN = 1, and BB = 0, ACT bit is set to "1" if "1" is written to MSS bit and a start condition is output to I2C bus. Then, once the start condition is received, BB bit is set to "1", indicating that I2C bus is on the communication (See, Figure 15.20-4).

Figure 15.20-4 Relationship Between Start Condition and Each Bit



Note:

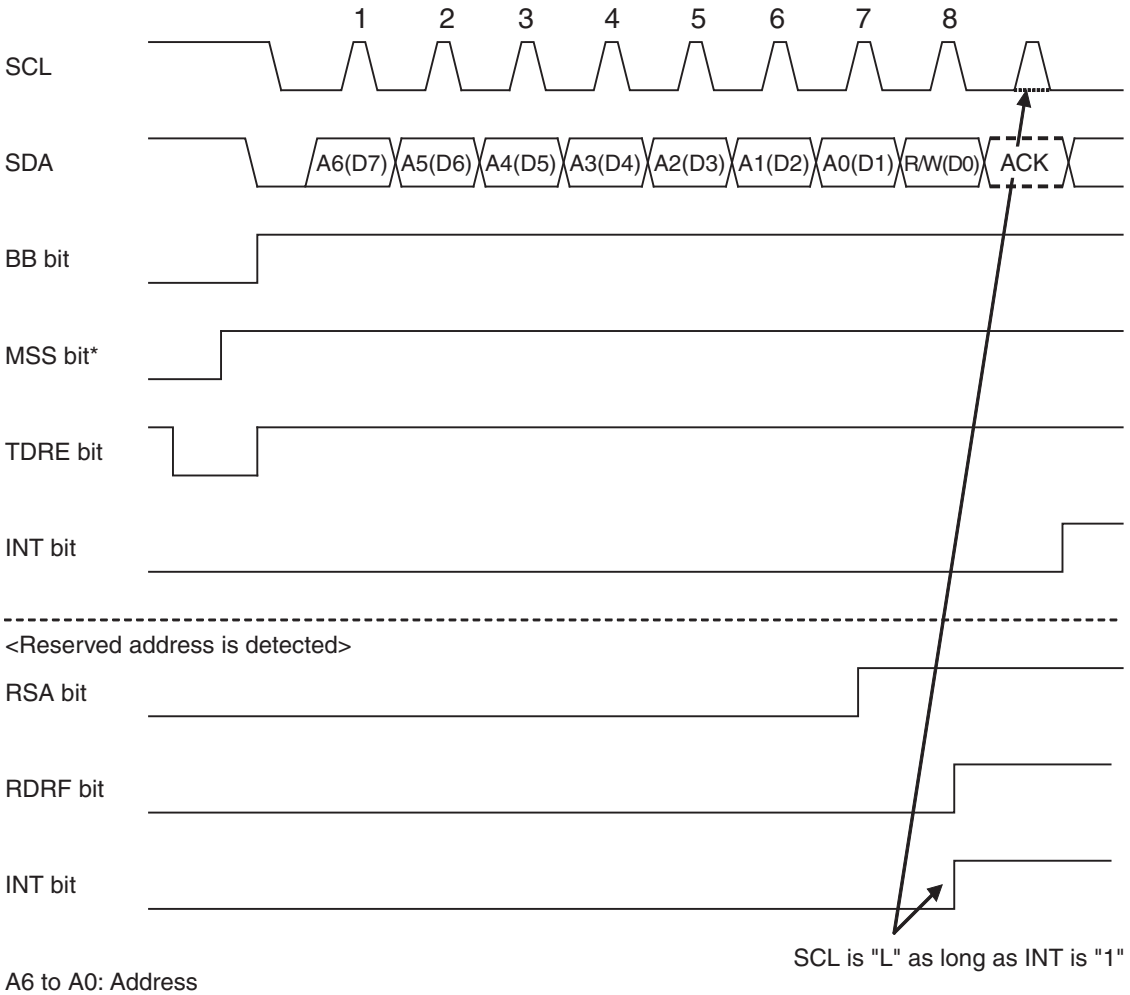
In operating mode 4 (I2C mode), the machine clock should be used for 8 MHz or more and setting over 400 kbps to the baud rate generator is disabled.

■ Slave Address Output

When the start condition is output, the data set in the TDR register is output from bit7 as the address. If FIFO is enabled, the first data written to TDR is output. bit0 is used as the data direction bit (R/W). When the data direction bit (R/W) is "0", the data indicates the write direction (master → slave). Set the address for the TDR register before "1" is written to MSS or SCC.

Figure 15.20-5 and Figure 15.20-6 show the output timing of the address and data direction.

Figure 15.20-5 Address and Data Direction (When FIFO is Disabled)



A6 to A0: Address

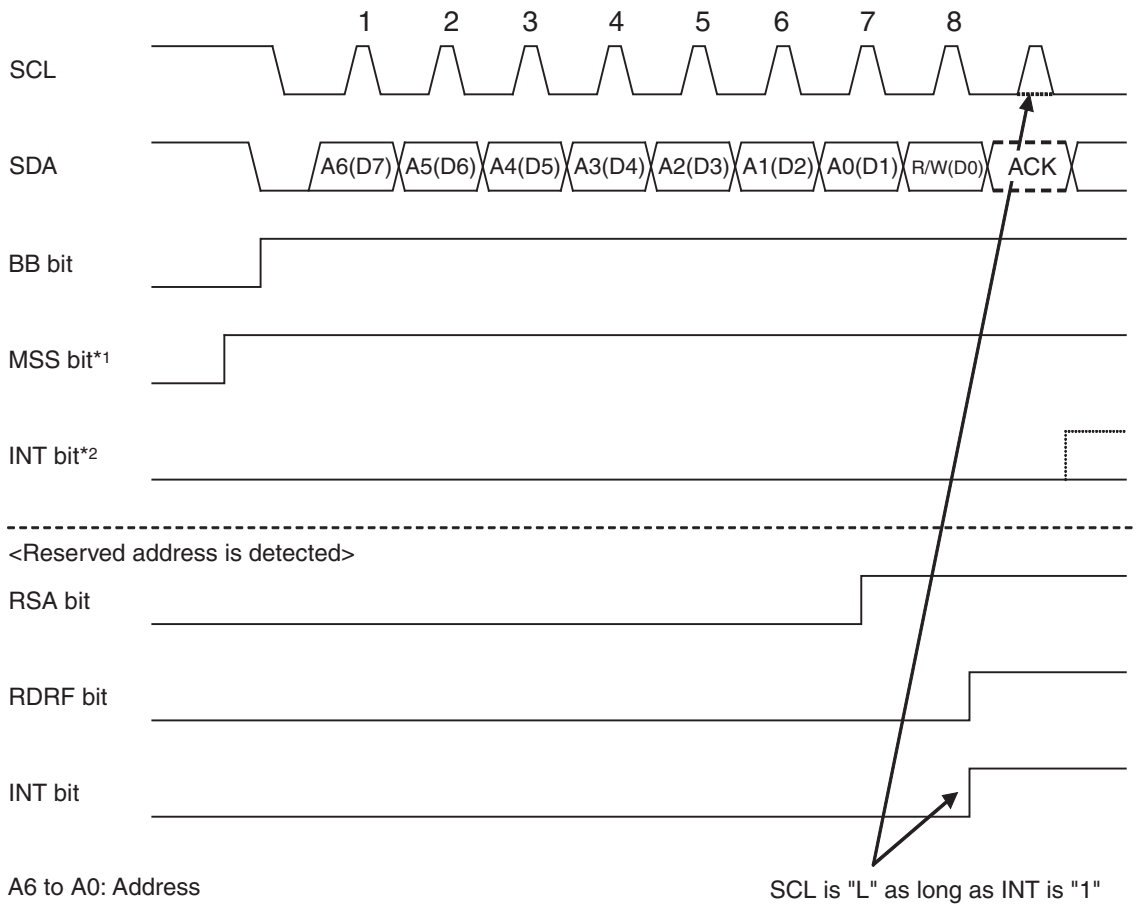
D7 to D0: TDR register bit

R/W: Data direction ("L": write direction)

ACK: Acknowledgment ("L": acknowledgment, output from slave)

*: Set the address for the TDR register before "1" is written to MSS bit.

Figure 15.20-6 Address and Data Direction (When Transmission/Reception FIFO are Enabled)



A6 to A0: Address

D7 to D0: TDR register bit

R/W: Data direction ("L": write direction)

ACK: Acknowledgment ("L": acknowledgment, output from slave)

*1: Set the address for the TDR register before "1" is written to MSS bit.

*2: When acknowledgment is "L", R/W = "L", and transmission FIFO has data, or when acknowledgment is "L", R/W = "H", and reception FIFO has no data, INT bit doesn't become "1".

■ Acknowledgment Reception by Transmitting First Byte

When the data direction bit (R/W) is output, I2C interface receives the acknowledgment from the slave. The table below shows the operation when FIFO is enabled and when FIFO is disabled.

Table 15.20-1 Operation after Acknowledgment is Received (RSA bit = 0)

Trans-mission FIFO	Recep-tion FIFO	State of the transmis-sion FIFO	State of the reception FIFO	Data direction bit (R/W)	Operation right after the acknowledgment is received	
					Acknowledgment is ACK	Acknowledgment is NACK
Disable	Disable	-	-	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit is set to "1" and wait.
				1		
Disable	Enable	-	No data	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit is set to "1" and wait.
			There is data		INT bit is set to "1" and wait.	
			-	1	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	
Enable	Disable	-	-	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit is set to "1" and wait.
				1		
Enable	Enable	-	No data	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit is set to "1" and wait.
			There is data		INT bit is set to "1" and wait.	
			-	1	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	

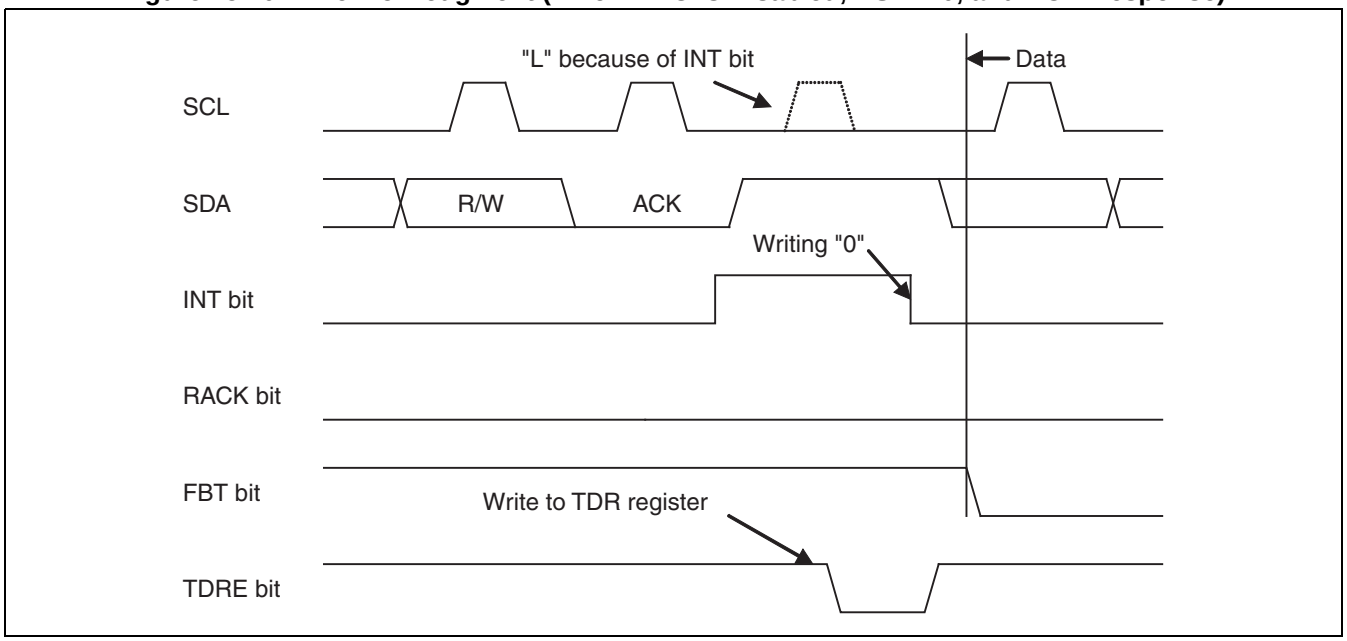
● FIFO disabled (both transmission and reception FIFOs are disabled)

- When RSA bit is "0", the interrupt flag bit (INT) is set to "1" and SCL is retained as "L" to wait if TDRE bit is "1" after the acknowledgment is received. To release the wait, write "0" to the interrupt flag. If TDRE bit is "0", the clock is generated to SCL instead of setting "1" to the interrupt flag when ACK is received.
- When RSA bit is "1", the interrupt flag bit (INT) is set to "1" and SCL is retained as "L" to wait after the reserved address is received (before the acknowledgment). After RDR register is read, the interrupt flag becomes "0" to release the wait when you set ACKE bit and the transmission data, and write "0" to the interrupt flag.
- The received acknowledgment is set to RACK bit. If you check the RACK bit during the wait and it is NACK, write "0" to MSS bit or "1" to SCC bit in order to generate a stop or repetitive start condition. At this point, INT bit is automatically cleared to "0".

● FIFO enabled

- FIFO should be set as follows before "1" is set to MSS bit.
  - When transmitting to slave (data direction bit = 0), data including slave address should be set to the transmission FIFO
  - When receiving data from slave (data direction bit = 1), specify the number of data to be received to the FIFO byte count register, and write to the transmission data register using slave address, data direction bit, and the number of dummy data that you want to receive.
- When RSA bit is "0", the data is transmitted and received according to the data direction bit, instead of setting the interrupt flag bit (INT) to "1" after the acknowledgment is received and it is ACK (no wait). If it is NACK, the interrupt flag bit (INT) is set to "1" and SCL is retained as "L" to wait.
- The received acknowledgment is stored to RACK bit. If you check the RACK bit during the wait and it is NACK, write "0" to MSS bit or "1" to SCC bit in order to generate a stop or repetitive start condition. At this point, INT bit is automatically cleared to "0".

Figure 15.20-7 Acknowledgment (When FIFO is Disabled, RSA = 0, and ACK Response)



The wait for the address is generated:

- After the reception of the acknowledgment when RSA bit is "0"
- Before the reception of the acknowledgment when RSA bit is "1"

These are independent of the WSEL setting.

Figure 15.20-8 Acknowledgment (When FIFO is Disabled, RSA = 0, and NACK Response)

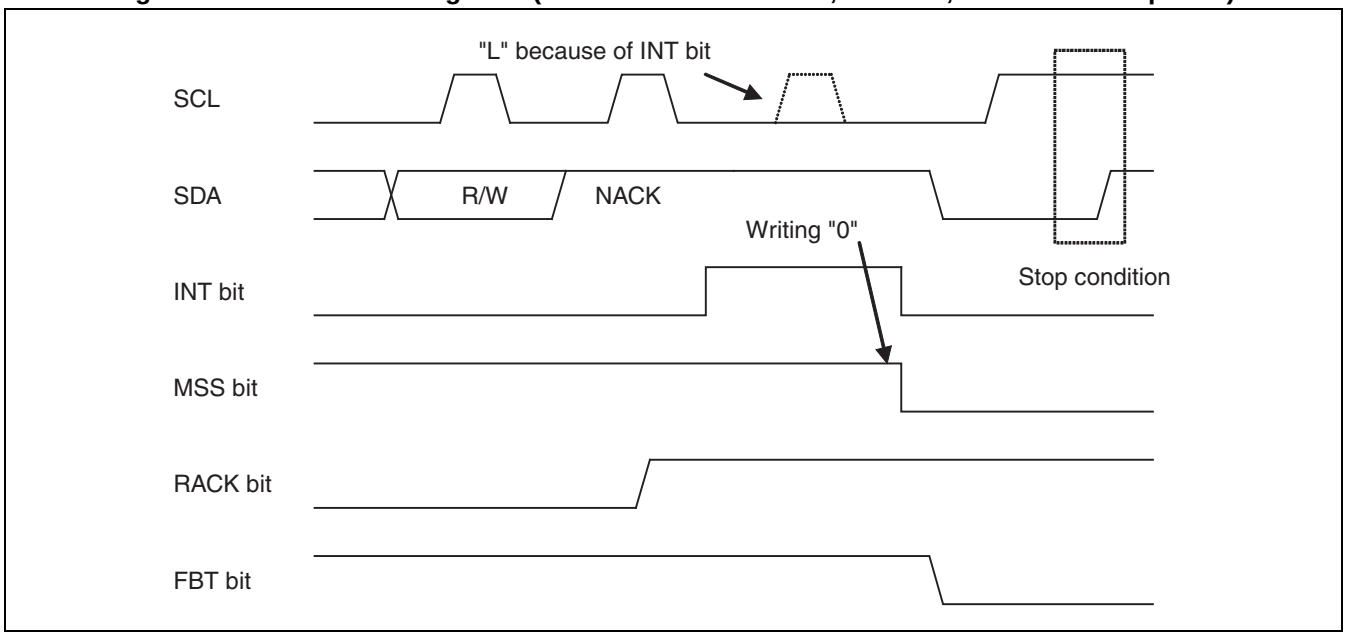


Figure 15.20-9 Acknowledgment (When FIFO is Disabled, RSA = 1, and ACK Response)

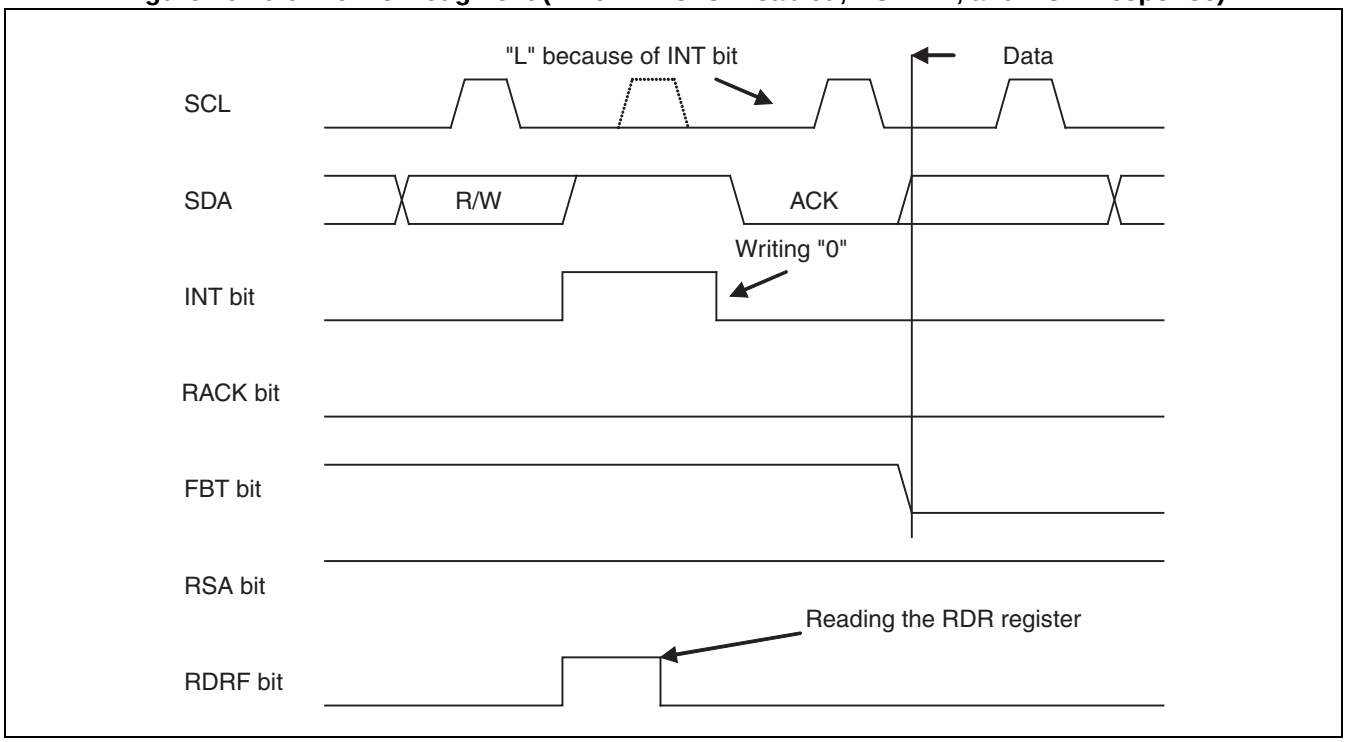


Figure 15.20-10 Acknowledgment (When FIFO is Disabled, RSA = 1, and NACK Response)

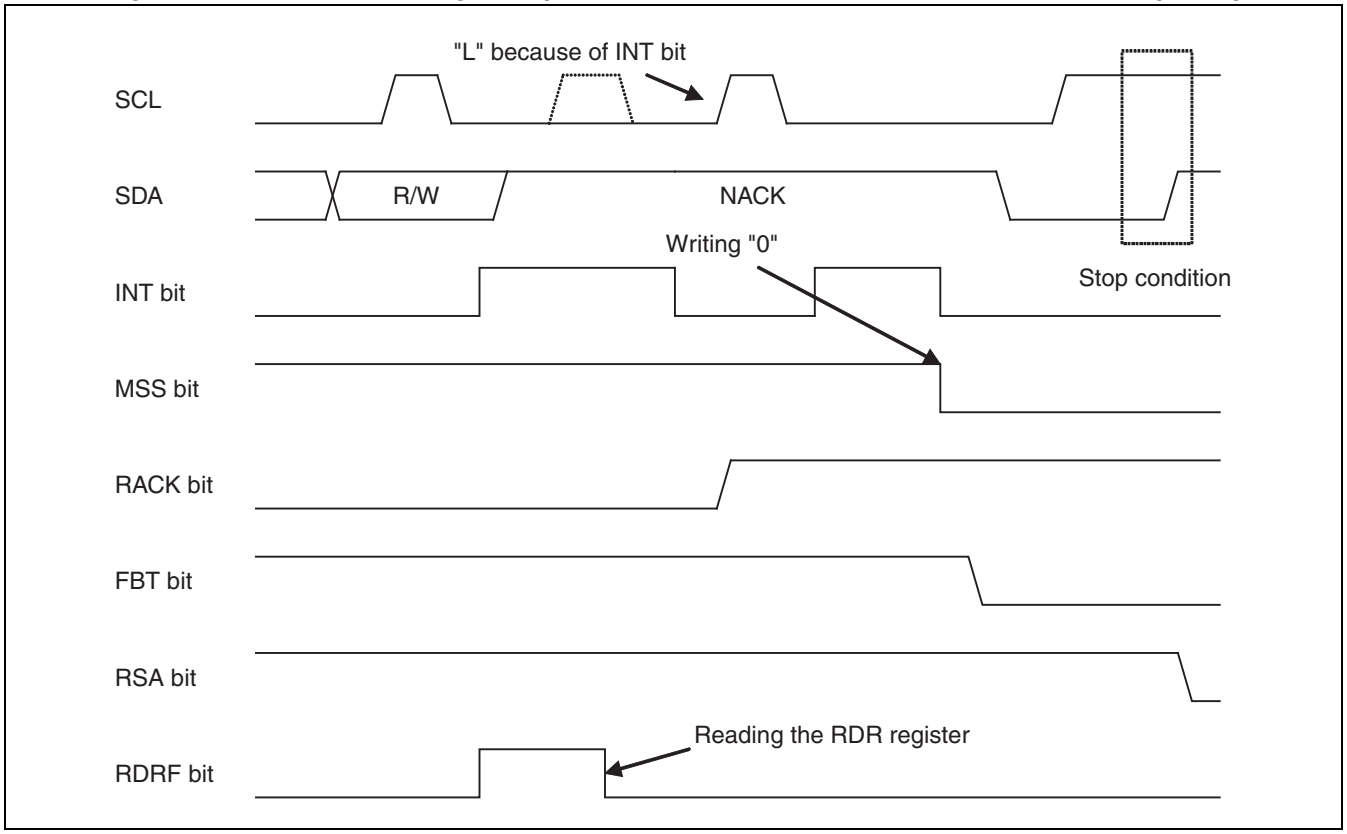
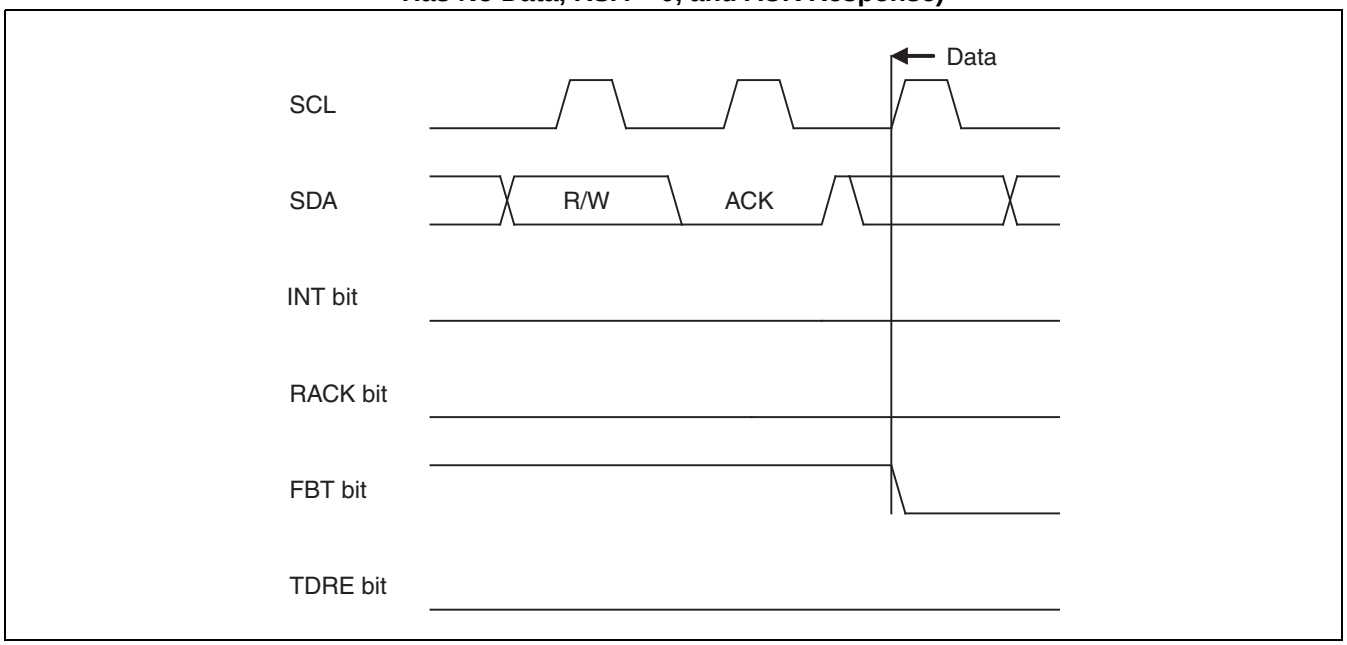


Figure 15.20-11 Acknowledgment (When FIFO is Enabled, Transmission FIFO Has Data, Reception FIFO Has No Data, RSA = 0, and ACK Response)



■ Data Transmission by Master

When the data direction bit (R/W) is "0", data is transmitted from the master. Slave responds ACK or NACK on every 1 byte transmission.

The wait is generated at the following point depending on the setting for WSEL bit:

Table 15.20-2 WSEL Bit When Transmitting Master Data

WSEL bit	Operation
0	In the 2nd or subsequent byte, the interrupt flag bit (INT) is set to "0" and SCL is set to "L" to go into the wait state when TDRE bit is "1" or after the acknowledgment on the arbitration lost detection. The interrupt flag bit (INT) is set to "1" to go into the wait state after the acknowledgment if FIFO is enabled, or after the acknowledgment when the arbitration lost is detected or the transmission data register has no longer valid data (TDRE = 1).
1	In the 2nd or subsequent byte, the interrupt flag bit (INT) is set to "1" and SCL is set to "L" to go into the wait state when TDRE bit is "1" or after the master transmitted 1 byte data on the arbitration lost detection. The interrupt flag bit (INT) is set to "1" to go into the wait state after the data is transmitted when the arbitration lost is detected or the transmission data register has no longer valid data (TDRE = 1), if FIFO is enabled.

However, the interrupt flag (INT) is set after the acknowledgment regardless of the WSEL setting in the following case:

NACK is received other than when the stop conditions are set (MSS = 0, ACT= 1)

An example of procedure to transmit the data to the slave is described below:

● When transmitting to other than the reserved address

- When the transmission FIFO is disabled
  (1) Set the slave address (including the data direction bit) to the TDR register, and write "1" to MSS bit.
  (2) After the slave address is transmitted, ACK is received and the interrupt flag (INT) becomes "1".
  (3) Write the data to transmit to the TDR register.
  (4) Write "0" to the interrupt flag (INT) together with the update of WSEL bit to release the wait of I^2C bus.
  (5) Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (2) to (4) until the predefined number of data is transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".
  (6) Set MSS bit to "0" or SCC bit to "1" to generate a stop or repetitive start condition.
- When the transmission FIFO is enabled
  (1) Write the slave address (including the data direction bit) and the transmission data to the TDR register.
  (2) Write "1" to MSS bit along with setting up WSEL bit.
  (3) Set the interrupt flag bit (INT) to "1" to wait the I^2C bus right after the NACK is received during transmission. Set the interrupt flag to "1" to wait the I^2C bus according to the setting for WSEL after the last byte is transmitted when all ACK responses are received.
  (4) Generate a stop condition by writing "0" to MSS bit.

● When transmitting to the reserved address

- When the transmission FIFO is disabled
  (1) Set the reserved address as a slave address to the TDR register, and write "1" to MSS bit.
  (2) Once the slave address is transmitted, the interrupt flag (INT) becomes "1".
  (3) Read the RDR register and verify the reserved address.*
  (4) Write the data to transmit to the TDR register.
  (5) Write "0" to the interrupt flag (INT) together with the update of WSEL bit to release the wait of I^2C bus.
  (6) Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (4) to (6) until the predefined number of data is transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".
  (7) Set MSS bit to "0" or SCC bit to "1" to generate a stop or repetitive start condition.

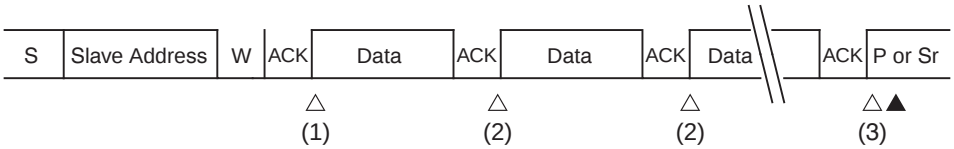
- When the transmission FIFO is enabled
  (1) Set the reserved address as a slave address to the TDR register, and write "1" to MSS bit.
  (2) Once the slave address is transmitted, the interrupt flag (INT) becomes "1".
  (3) Read the RDR register and verify the reserved address.*
  (4) Write all transmission data to the TDR register (until the transmission FIFO gets full, if possible).
  (5) Set the interrupt flag bit (INT) to "1" to wait the I2C bus right after the NACK is received during transmission. Set the interrupt flag to "1" to wait the I2C bus according to the setting for WSEL after the last byte is transmitted when all ACK responses are received.
  (6) Set MSS bit to "0" or SCC bit to "1" to generate a stop or repetitive start condition.

*: Set ACKE and WSEL bits to "1" to verify whether to operate as a master or a slave on next data if an arbitration lost is generated and there is a possibility to operate as a slave when the reserved address is a general call in the multi-master.

Notes:

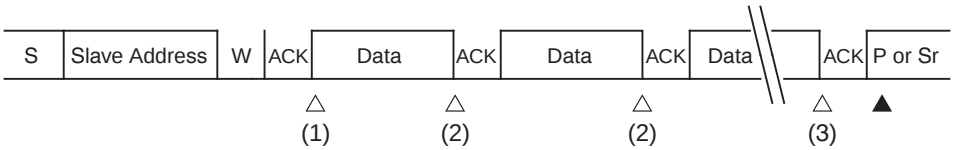
- Change IBCR register during transmission and reception, when the interrupt flag (INT) is "1".
- WSEL bit is changed, which is used as a generation condition of an interrupt flag (INT) for next data.
- If the transmission data is written to the TDR register when transmitting the data (TDRE = 1) and the ACK response is detected, the interrupt flag (INT) will not become "1" and the written data is transmitted instead.
- If the transmission data is written to the TDR register when transmitting the data (TDRE = 1) and the ACK response is detected, the interrupt flag (INT) will not become "1" and only RDRF becomes "1" instead (if the number of data set to the FBYTE register are received when the reception FIFO is enabled).

Figure 15.20-12 Interrupt of Master by Disabling FIFO (1) (WSEL = 0, RSA = 0)



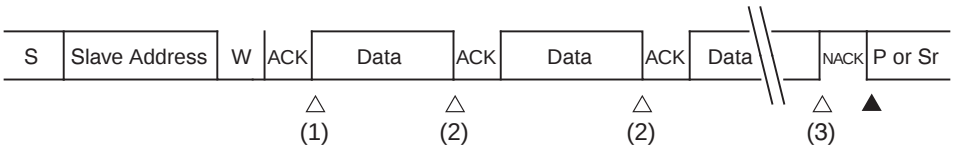
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(3) An interrupt generated by 1 byte transmission + acknowledgment reception
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-13 Interrupt of Master Transmission by Disabling FIFO (2) (WSEL = 1, RSA = 0, ACK Response)



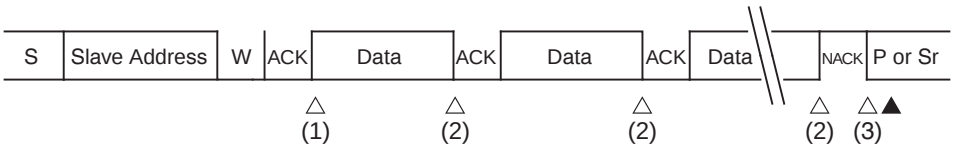
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = 0 after the transmission data is written to TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

**Figure 15.20-14 Interrupt of Master Transmission by Disabling FIFO (3)
(WSEL = 1, RSA = 0, NACK Response)**



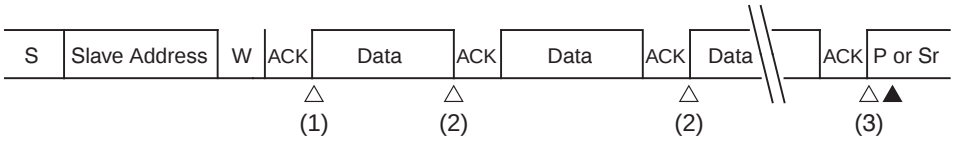
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = 0 after the transmission data is written to TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

**Figure 15.20-15 Interrupt of Master Transmission by Disabling FIFO (4)
(WSEL = 1, RSA = 0, Halfway NACK Response)**



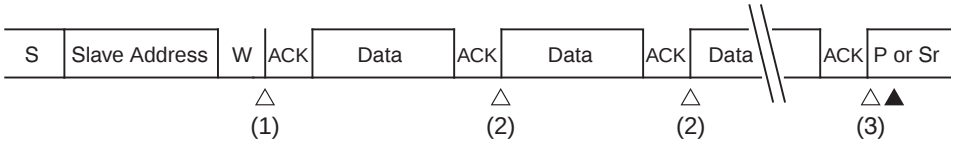
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = 0 after the transmission data is written to TDR register
(3) An interrupt generated by NACK response
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-16 Interrupt of Master Transmission by Disabling FIFO (5) (WSEL = 1 → 0, RSA = 0, ACK Response)



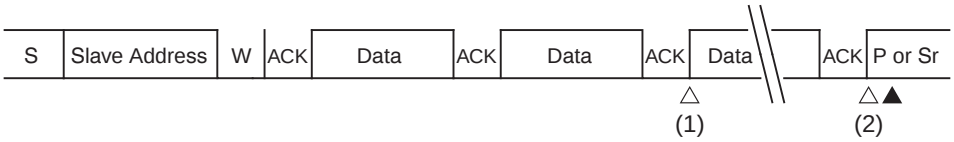
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to the transmission buffer
(2) An interrupt generated by 1 byte transmission
- Write INT = 0 and WSEL = 0 after the transmission data is written to the transmission buffer
(3) An interrupt generated by 1 byte transmission
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-17 Interrupt of Master by Disabling FIFO (6) (WSEL = 0, RSA = 1)



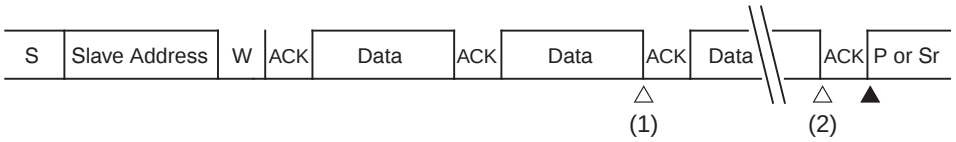
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- Write INT = 0 after the transmission data is written to TDR register
(3) An interrupt generated by 1 byte transmission + acknowledgment reception
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-18 Interrupt of Master Transmission by Enabling FIFO (7) (WSEL = 0, RSA = 0, ACK Response)



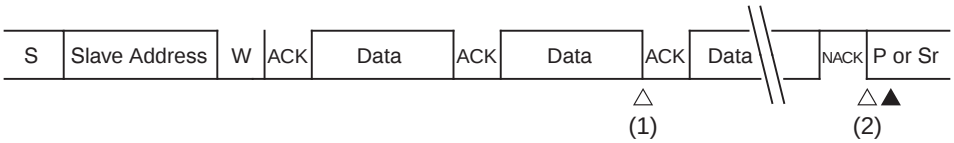
S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = 0 after the transmission data is written to the transmission FIFO
(2) An interrupt generated by the last byte transmission (transmission FIFO is empty)
+ acknowledgment reception
- Set MSS = 0 or MSS = 1 and SCC = 1

Figure 15.20-19 Interrupt of Master Transmission by Enabling FIFO (8) (WSEL = 1, RSA = 0)



S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = 0 after the transmission data is written to the transmission FIFO
(2) An interrupt generated by the last byte transmission (transmission FIFO is empty)
- Set MSS = 0 or MSS = 1 and SCC = 1

Figure 15.20-20 Interrupt of Master Transmission by Enabling FIFO (9) (WSEL = 1, RSA = 0, NACK Response)



S : Start condition
W : Data direction bit (Write direction)
P : Stop condition
Sr : Repetitive start condition
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = 0 after the transmission data is written to the transmission FIFO
(2) An interrupt generated by NACK response
- Set MSS = 0 or MSS = 1 and SCC = 1

■ Data Reception by Master

When the data direction bit (R/W) is "1", the data transmitted from the slave is received.

When FIFO is disabled, master generates a wait every 1 byte received (INT = 1, RDRF = 1) if TDRE bit is "1", and performs ACK or NACK response with the setting for ACKE bit in the IBCR register according to WSEL bit. If TDRE bit is "0", next data is received instead of generating a wait (INT = 0) when ACK response is performed with the setting for ACKE bit in the IBCR register, or a wait is generated when NACK response is performed (INT = 1).

When FIFO is enabled, RDRF bit is set once the same number of bytes as the defined number of reception byte is received. Set the interrupt flag when TDRE bit is "1" to wait the I^2C bus. NACK response is performed to set the interrupt flag to "1" when TDRE bit becomes "1" if WSEL = 0. When WSEL = 1, a wait is generated after the last byte is received. Set ACKE bit during the wait, and then ACK or NACK response is performed according to the setting for ACKE bit once the interrupt flag is cleared to "0". The data is stored to the reception FIFO as the reception data even if NACK is output.

See the table below about the wait generated by the interrupt:

Table 15.20-3 WSEL Bit When Receiving Master Data

WSEL bit	Operation
0	In the 2nd or subsequent byte, the interrupt flag bit (INT) is set to "1" and SCL is set to "L" to go into the wait state after the acknowledgment when TDRE bit is "1".
1	In the 2nd or subsequent byte, the interrupt flag bit (INT) is set to "1" and SCL is set to "L" to go into the wait state after the master received 1 byte data when TDRE bit is "1

An example of procedure to receive the data from the slave is described below:

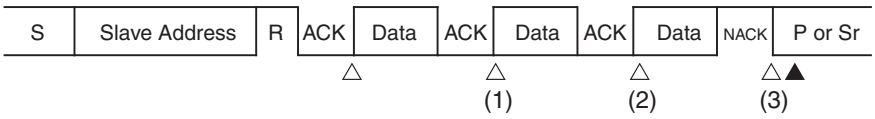
- When the reception FIFO is disabled
  (1) Set the slave address (including the data direction bit) to the TDR register, and write "1" to MSS bit.
  (2) After the slave address is transmitted, ACK is received and the interrupt flag (INT) becomes "1".
  (3) Write "0" to the interrupt flag bit (INT) together with the update of WSEL bit to release the wait of I^2C bus.
  (4) Put the I^2C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when WSEL is set to "0", or immediately after one byte has been received when WSEL is set to "1". Repeat (2) to (4) until the predefined number of data is received.
  (5) Output NACK and set MSS bit to "0" or SCC bit to "1" to generate a stop or repetitive start condition after the last data is received.

- When the transmission and reception FIFO is enabled
  (1) Write the slave address (including the data direction bit) to the TDR register.
  (2) Set the number of data to be received to the FBYTE register.
  (3) Write "1" to MSS bit.
  (4) ACK response is performed and receiving is continued as long as TDRE bit is "0". RDRF becomes "1" once the amount of data predefined to the FBYTE is received during this receiving period. RDR register is read when RDRF bit becomes "1".
  (5) Set the interrupt flag to "1" to wait the I^2C bus after NACK is output when WSEL = 0, or right after 1st byte is received when WSEL = 0 once TDRE bit becomes "1".
  (6) Set MSS bit to "0" or SCC bit to "1" to generate a stop or repetitive start condition after ACKE bit is set to "0" when WSEL = 1, or regardless of the setting for ACKE bit when WSEL = 0.

Notes:

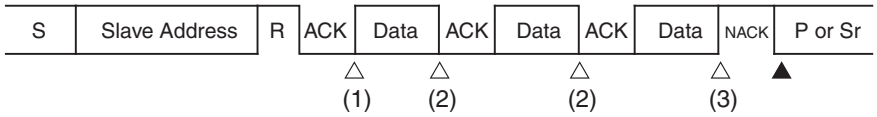
- When TDRE bit is "0", even if an overrun error is generated, the acknowledgment is output according to the setting for ACKE bit and the following processes are executed:
- Change IBCR register during transmission and reception, when the interrupt flag (INT) is "1".
- The interrupt flag (INT) remains "0" and next data is received when a dummy data is written to the TDR register and TDRE bit is "0" at the timing when the interrupt flag (INT) becomes "1" in master reception.
- If data is received when the reception FIFO is enabled and WSEL = 0, RDRF bit becomes "1" after the last bit is received and the interrupt flag (INT) becomes "1" after ACK is transmitted.

Figure 15.20-21 Interrupt of Master Reception by Disabling FIFO (1) (WSEL = 0, RSA = 0)



△ : Interrupt because of INTE = 1
▲: Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Interrupt is cleared to "0" by writing INT = 0
(2) An interrupt generated by 1 byte reception + acknowledgment transmission
- Set ACKE = 0 and then write INT = 0 after the reception data is read
(3) An interrupt generated by 1 byte reception + acknowledgment transmission
- Set MSS = 0 or MSS = 1 and SCC = 1
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-22 Interrupt of Master Reception by Disabling FIFO (2) (WSEL = 1, RSA = 0)



△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Interrupt is cleared to "0" by writing INT = 0
(2) An interrupt generated by 1 byte reception
- Write INT = 0 after the reception data is read
(3) An interrupt generated by 1 byte reception
- Set ACKE = 0 and MSS = 0, or MSS = 1, SCC = 1 after reception data is read
(Note) TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 15.20-23 Interrupt of Master Reception by Enabling FIFO (3) (WSEL = 0, ACKE=0, RSA = 0)

△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by TDRE = 1
- Set MSS = 0, or MSS = 1 and SCC = 1 after all data is read from the reception FIFO

Figure 15.20-24 Interrupt of Master Reception by Enabling FIFO (4) (WSEL = 1, RSA = 0)

△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by TDRE = 1
- Set ACKE = 0 and MSS = 0, or MSS = 1 and SCC = 1 after all data is read from the reception FIFO

■ Arbitration Lost

When a master receives the different data than the transmitted data due to a collision of data from another master, which is recognized as arbitration lost, and the master becomes available to operate as slave mode, setting MSS bit to "0" and AL bit to "1".

AL bit is cleared to "0" when:

- "1" is written to MSS bit
- "0" is written to INT bit
- "0" is written to SPC bit when AL bit = 1 and SPC bit = 1
- I^2C interface is disabled (EN bit = 0)

When the arbitration lost is generated, the interrupt flag (INT) is set to "1" and the SCL of I^2C bus to "L" according to the setting for WSEL.

■ Wait of the Master Mode

When BB bit is "1" and MSS bit is set to "1", if the slave mode is not operated, wait the master mode as long as the BB bit is "1". The start condition is transmitted when BB bit becomes "0". You can judge whether the master mode is in wait state or not using MSS and ACT bits (in wait state if MSS = 1 and ACT = 0). To operate as the slave mode after MSS bit is set to "1", set AL and ACT bits to "1" and MSS bit to "0".

15.20.2 Slave Mode

Slave mode is used to detect the (repetitive) start condition. When the combination of ISBA and ISMK registers matches with received address, ACK response is performed and the slave mode is enabled.

■ Slave Address Match Detection

When the (repetitive) start condition is detected, 7-bit of next data is received as the address. ISBA register and each bit of the received address are compared about the bit where "1" is set in the ISMK register, and if they are matched, ACK is output.

Table 15.20-4 Operation Right After Acknowledgment is Output for Slave Address

Trans-mission FIFO	Recep-tion FIFO	State of the transmis-sion FIFO	State of the reception FIFO	Data direction bit (R/W)	Operation right after the acknowledgment	
					Acknowledgment is ACK	Acknowledgment is NACK
Disable	Disable	-	-	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit remains "0" and no wait.
				1		
Disable	Enable	-	No data	0	INT bit remains "0" and no wait.	INT bit remains "0" and no wait.
			There is data		INT bit is set to "1" and wait.	
			-	1	TINT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	
Enable	Disable	-	-	0	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	INT bit remains "0" and no wait.
				1		
Enable	Enable	-	No data	0	INT bit remains "0" and no wait.	INT bit remains "0" and no wait.
			There is data		INT bit is set to "1" and wait.	
			-	1	INT bit is set to "1" and wait when TDRE bit is "1". INT bit remains "0" and no wait when TDRE bit is "0".	

- Reserved address detection

  If 1st byte matches with the reserved address ("$0000XXXX_B$" or "$1111XXXX_B$"), INT bit is set to "1" to wait I^2C bus after the data in 8th bit is received whichever the transmission and reception FIFO is enabled or not. If you want to read the reception data to operate as a slave at this point, set ACKE to "1" to clear INT bit to "0". Then, it operates as a slave. If ACKE is "0", it does not operate as slave after the acknowledgment is output.

■ Data Direction Bit

After the address is received, the data direction bit is received that decides the transmission and reception of the data. When this bit is set to "0", it indicates the transmission from the master, and the data is received as a slave.

■ Reception by Slave

When the slave address matches and the data direction bit is "0", which indicates the reception by the slave mode. An example of procedure to receive data by the slave mode is described below:

- When the reception FIFO is disabled

  (1) After ACK is transmitted, set the interrupt flag bit (INT) to "1" to wait the I^2C bus. When recognized as an interrupt generated by the match of the salve address using MSS, ACT, and FBT bits, 1 is written to the ACKE bit and "0" is written to the interrupt flag (INT) to release the wait of I^2C bus (See, Table 15.20-4).

  (2) After 1 byte data is received, the interrupt flag (INT) is set to "1" to wait the I^2C bus according to the setting for WSEL.

  (3) After the received data is read from the RDR register and ACKE bit is set, "0" is written to the interrupt flag (INT) to release the wait of I^2C bus.

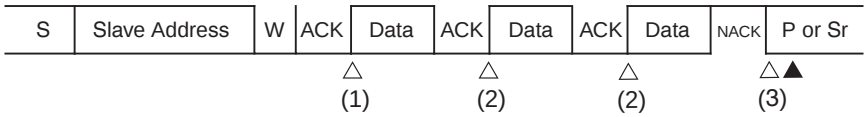
  (4) (2) to (3) are repeated until a stop or repetitive start condition is detected.

- When the reception FIFO is enabled

  (1) The interrupt flag bit (INT) becomes "1" to wait the I^2C bus when the NACK is detected or the reception FIFO gets full. When a stop or repetitive start condition is detected, SPC and RSC bits are set to "1" instead of the interrupt flag bit (INT) (no wait of I^2C bus). The reception FIFO sets RDRF bit to "1" when the number of received data reaches the value set to FIFO byte register (FBYTE). If RIE bit is "1" at this point, a reception interrupt occurs.

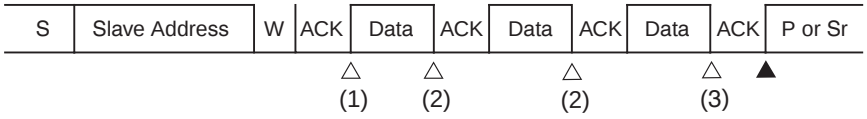
  (2) When the interrupt flag (INT) becomes "1", the received data is read from the RDR register. Once all the data is read, "0" is written to the interrupt flag to release the wait of I^2C bus. When a stop or repetitive start condition is detected, all the received data is read from the RDR register to clear SPC or RSC bit to "0".

Figure 15.20-25 Interrupt of Slave Reception by Disabling FIFO (1) (WSEL = 0, RSA = 0)



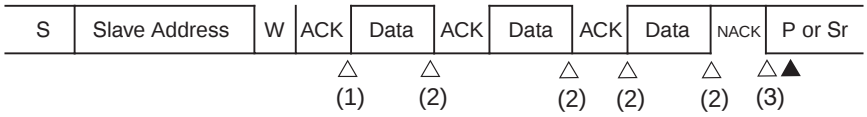
△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE = 1 and INT = 0
(2) An interrupt generated by 1 byte reception + ACK response
- Write INT = 0 after the reception data is read from the reception buffer
(3) An interrupt generated by 1 byte reception + NACK response
- Write INT = 0 after the reception data is read from the reception buffer

Figure 15.20-26 Interrupt of Slave Reception by Disabling FIFO (2) (WSEL = 1, RSA = 0)



△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE = 1 and INT = 0
(2) An interrupt generated by 1 byte reception
- Write INT = 0 after the reception data is read from the reception buffer
(3) An interrupt generated by 1 byte reception
- Write INT = 0 after the reception data is read from the reception buffer

Figure 15.20-27 Interrupt of Slave Reception by Disabling FIFO (3) (WSEL = 1, RSA = 0)



△ : Interrupt because of INTE = 1
▲ : Interrupt because of CNDE = 1
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE = 1 and INT = 0
(2) An interrupt generated by 1 byte reception
- Write INT = 0 after the reception data is read from the reception buffer
(3) An interrupt generated by NACK response
- Write INT = 0

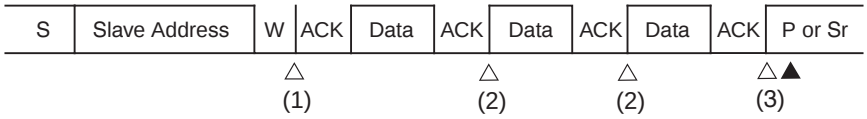
Figure 15.20-28 Interrupt of Slave Reception by Enabling Reception FIFO (4) (RSA = 0)

▲: Interrupt because of CNDE = 1
(1) An interrupt generated by a stop or repetitive start condition
- Read all data from the reception FIFO

Figure 15.20-29 Interrupt of Slave Reception by Enabling Reception FIFO (5) (RSA = 0)

△ : Interrupt because of INTE = 1
▲: Interrupt because of CNDE = 1
(1) An interrupt generated because the reception FIFO gets full
- Read all data from the reception FIFO, and write INT = 0

Figure 15.20-30 Interrupt of Slave Reception by Disabling FIFO (6) (WSEL = 0, RSA = 1)



△ : Interrupt because of INTE = 1
▲: Interrupt because of CNDE = 1
(1) An interrupt generated by matching the reserved address ("0000XXXX" of "1111XXXX")
- Read the reception data, and write ACKE = 1 and INT = 0
(2) An interrupt generated by 1 byte reception + acknowledgment output
- Write INT = 0
(3) An interrupt generated by 1 byte reception + acknowledgment output
- Interrupt by writing INT = 0

■ Transmission by Slave

When the slave address matches and the data direction bit is "1", which indicates the transmission by the slave mode. Set the interrupt flag (INT) to "1" to generate a wait after 1 byte is transmitted or ACK response depending on the setting for WSEL bit when FIFO is disabled (See, Table 15.20-4).

You can verify the acknowledgment output from the master using RACK bit. If NACK response is output, which indicates that the master cannot receive properly or data reception is completed. NACK is detected when WSEL = 1, an interrupt occurs and a wait is generated.

15.20.3 Bus Error

It is recognized as a bus error when a stop or (repetitive) start condition is detected while transmitting/receiving data on I2C bus.

■ Bus Error Generation Condition

A bus error sets BER bit to "1" when:

- A (repetitive) start or stop condition is detected while transferring 1st byte
- A (repetitive) start or stop condition is detected at 2nd to 9th bit (acknowledgment) of the data

■ Bus Error Operation

Implement appropriate error process if you verify BER bit when the interrupt flag (INT) generated by a transmission/reception becomes "1", and you found that BER bit is "1". You can clear BER bit by writing "0" to INT bit.

A bus error sets INT bit to "1" however, it does not start a wait state by setting SCL of the I2C bus to "L".

MB91313A Series

15.21 Dedicated Baud Rate Generator

Dedicated baud rate generator is used to set the frequency of the serial clock.

■ Baud Rate Selection

- Baud rate that can be obtained by dividing the internal clock with the dedicated baud rate generator (reload counter)

  There are 2 internal reload counters that correspond to transmission and reception serial clocks. Baud rate can be specified by setting the reload value for 15 bits with the baud rate generator registers 1, 0 (BGR1/BGR0).

  The reload counter divides the internal clock by the set value.

■ Calculating Baud Rate

2 of 15-bit reload counters are set using the baud rate generator registers 1, 0 (BGR1/BGR0).

The equation to calculate the baud rate is shown below:

(1) Reload value:

$V = \phi / b - 1$

V: Reload value, b: Baud rate, ϕ: Machine clock frequency

However, you may need to adjust the reload value because the defined baud rate may not be generated depending on the rising time of the SCL in the I^2C bus.

(2) Example of calculation

Where the machine clock is 16 MHz and the baud rate is set to 400 kbps, the reload value is calculated as follows:
Reload value:

$V = (16 \times 1000000) / 400000 - 1 = 39$

Therefore, the baud rate is:

$b = (16 \times 1000000) / (38 + 2) = 400$ kbps

Notes:

- Write to the baud rate generator registers 1, 0 (BGR1/BGR0) by 16-bit access.
- Set the baud rate generator registers when EN bit in the ISMK register is "0".
- In operating mode 4 (I^2C mode), the machine clock should be used for 8 MHz or more and setting over 400 kbps to the baud rate generator is disabled.
- If the reload value set is to "0", the reload counter will stop.

■ Reload Value and Baud Rate for Each Machine Clock Frequency

Table 15.21-1 Reload Value And Baud Rate

Baud rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32 MHz
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

Note: The table indicates the Value (set value of the BGR1/BGR0 register).

These values are for when the SCL rising in the I^2C bus is "0". If the SCL rising in the I^2C bus is delayed, the baud rate gets slower than the value above.

■ Functions of the Reload Counter

The reload counter consists of 15-bit register and generates a transmission and reception clock from the internal clock. You can read the count value of the transmission reload counter from the baud rate generator registers 1, 0 (BGR1/BGR0).

■ Start of a Count

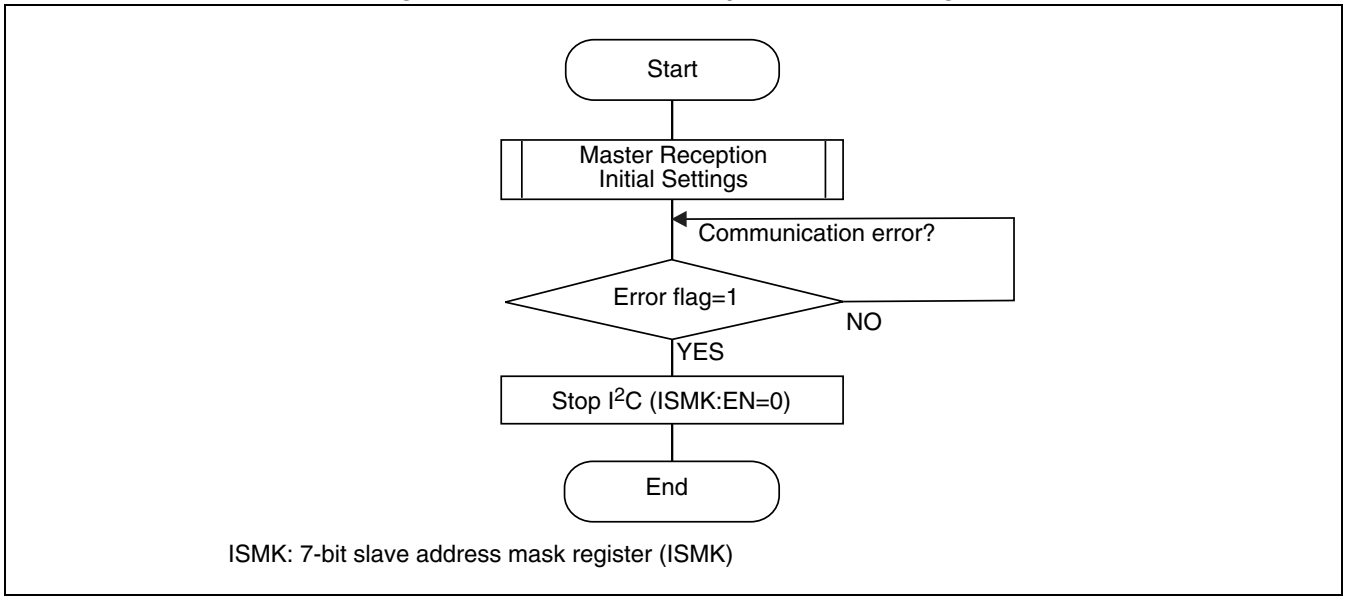
The reload counter starts counting when a reload value is written in the baud rate generator registers 1, 0 (BGR1/BGR0).

15.22 Examples of I2C Flowchart

This section shows examples of I2C communication flowchart.

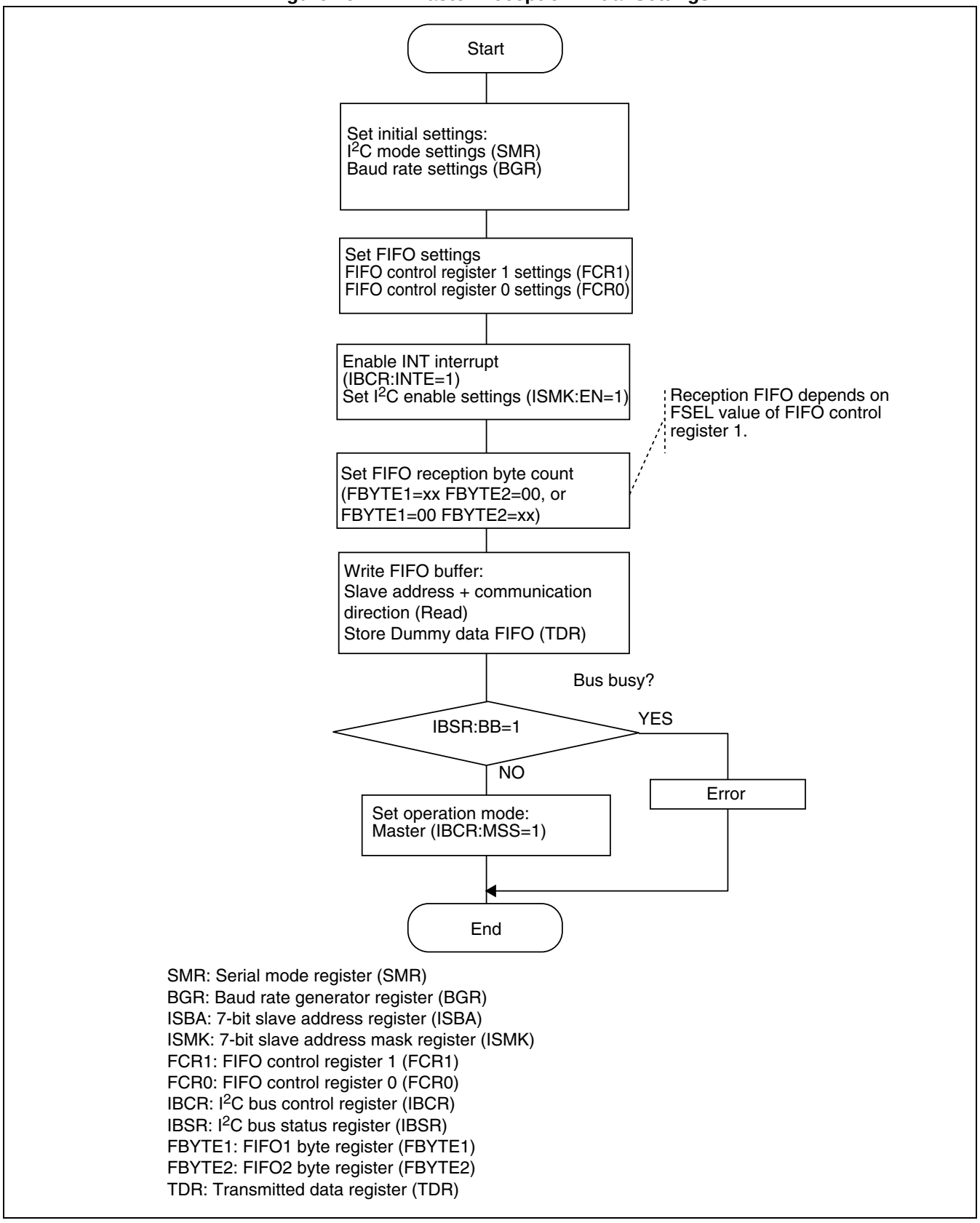
■ I2C Master Reception/ Slave Transmission FIFO Communication Flow

Figure 15.22-1 Master Reception Main Settings



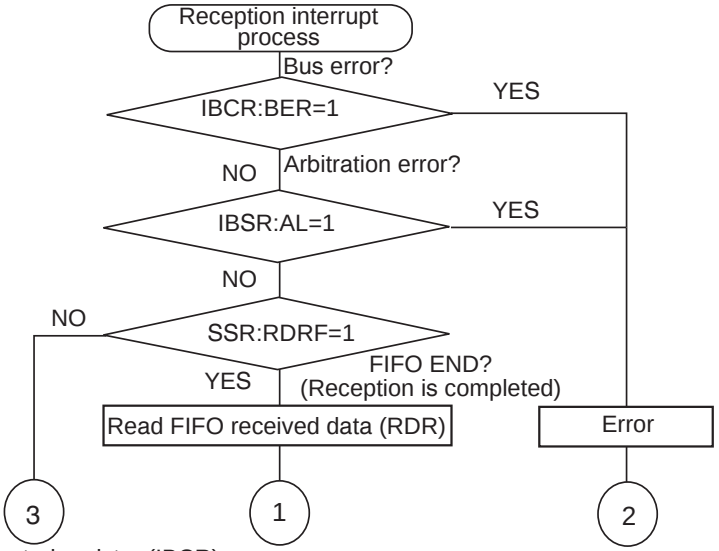
ISMK: 7-bit slave address mask register (ISMK)

Figure 15.22-2 Master Reception Initial Settings



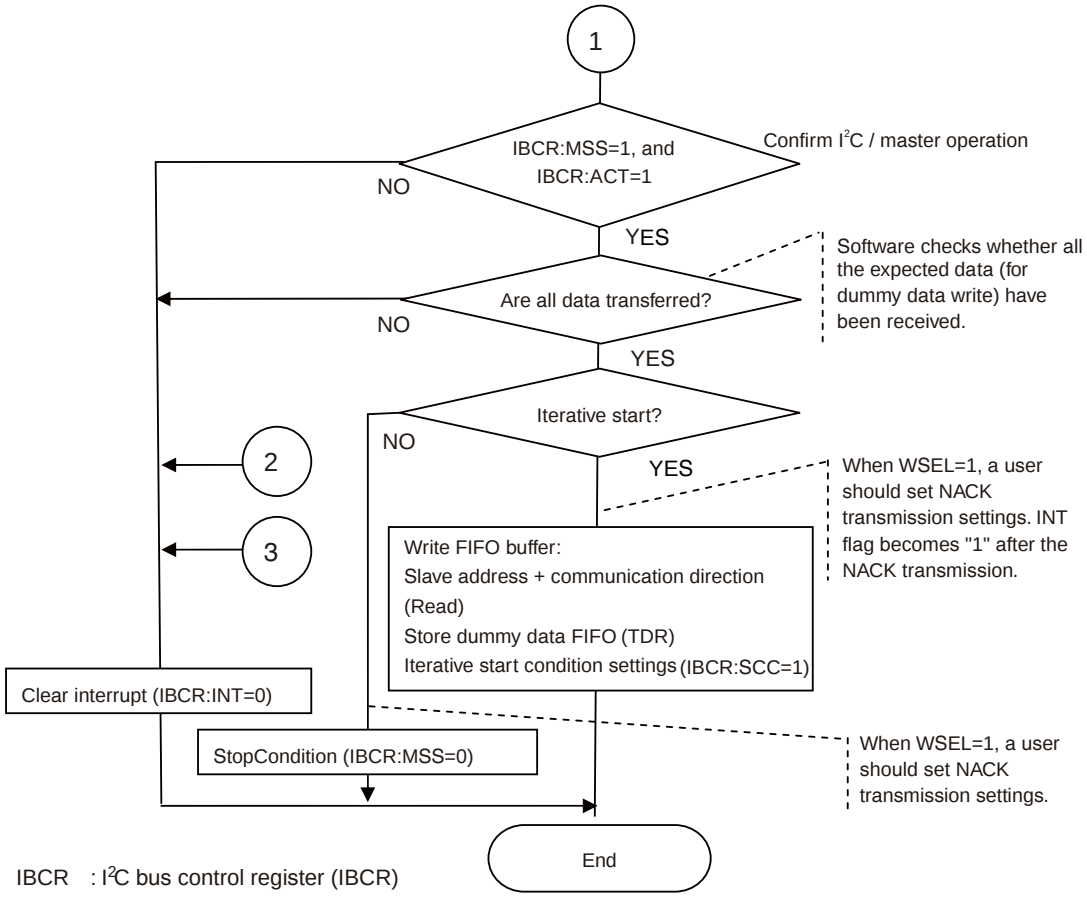
SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)
FBYTE1: FIFO1 byte register (FBYTE1)
FBYTE2: FIFO2 byte register (FBYTE2)
TDR: Transmitted data register (TDR)

Figure 15.22-3 Master Reception Interrupt Process



IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)
SSR: serial status register (SSR)
RDR: received data register (RDR)

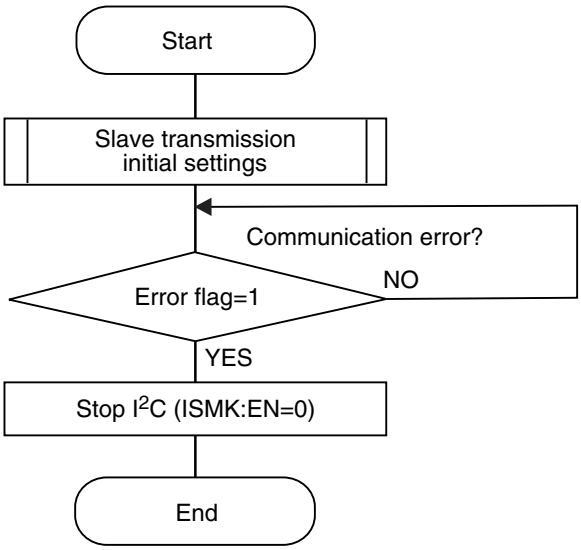
* For actual error handling, please judge each status error flag and handle each error by considering your systems.
* For final ACK process when receiving FIFO continuously, a wait will be generated after receiving the final data, if setting WSEL=1. Please respond with NACK by setting IBSR.RACK=1 during this wait.



IBCR : I2C bus control register (IBCR)

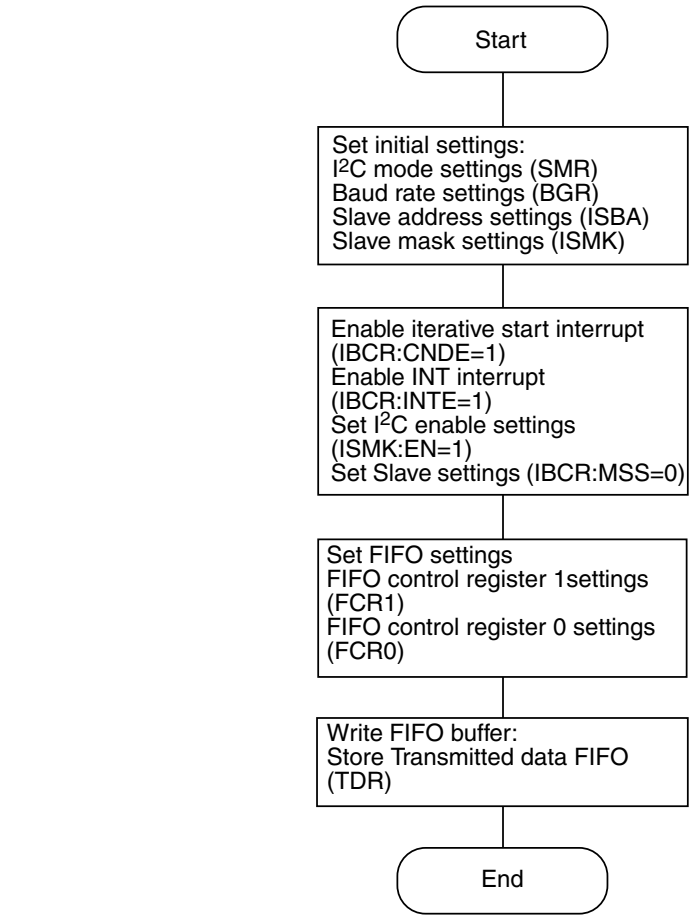
TDR : Transmitted data register (TDR)

Figure 15.22-4 Slave Transmission Main Settings



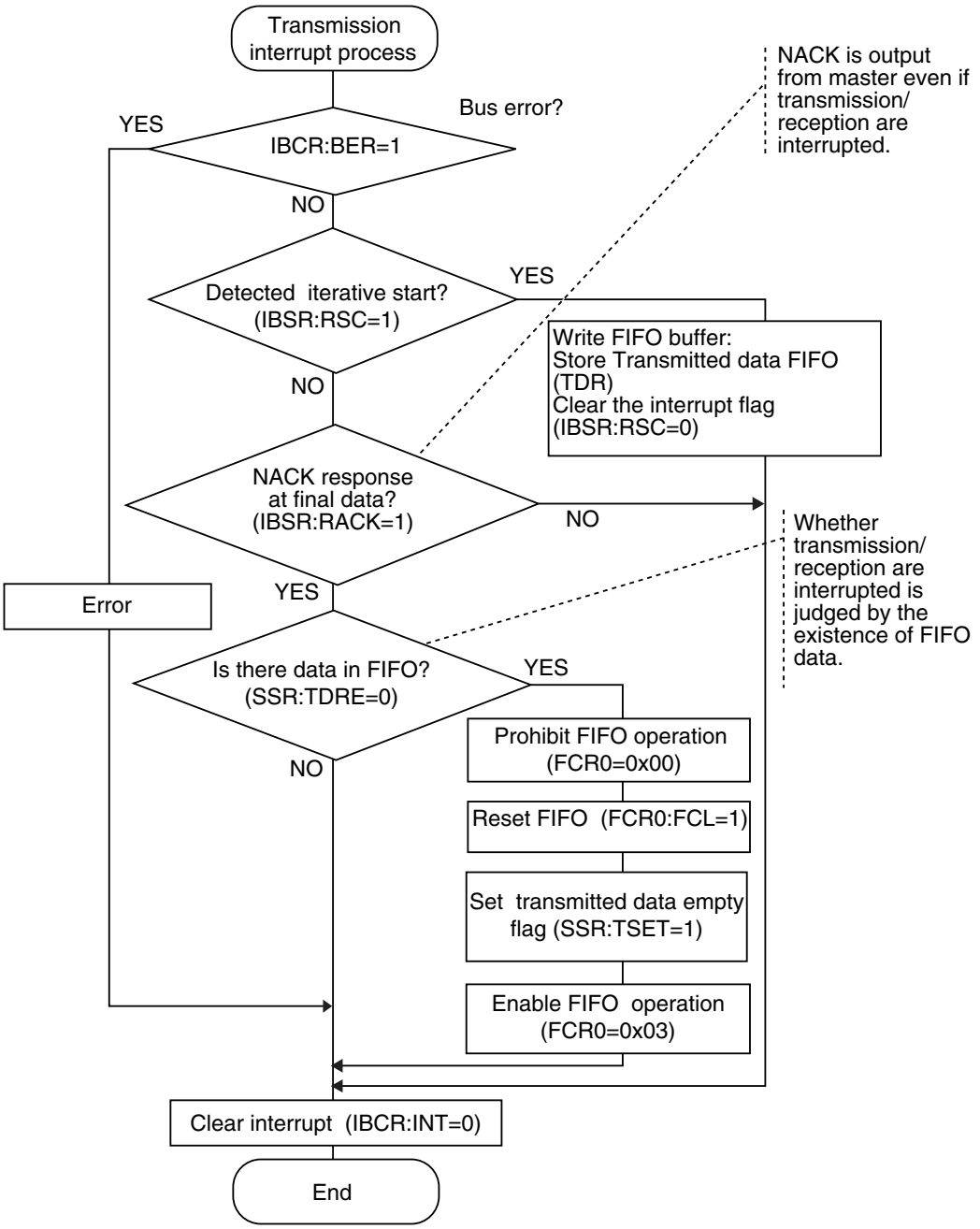
ISMK: 7-bit slave address mask register (ISMK)

Figure 15.22-5 Slave Transmission Initial Settings



SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
IBCR: I2C bus control register (IBCR)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
TDR: Transmitted data register (TDR)

Figure 15.22-6 Slave Transmission Interrupt Process



IBSR: I2C bus status register (IBSR)
TDR: Transmitted data register (TDR)
SSR: Serial status register (SSR)
FCR0: FIFO control register 0 (FCR0)
IBCR: I2C bus control register (IBCR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

■ I2C Master Transmission/ Slave Reception FIFO Communication Flow

Figure 15.22-7 Master Transmission Main Settings

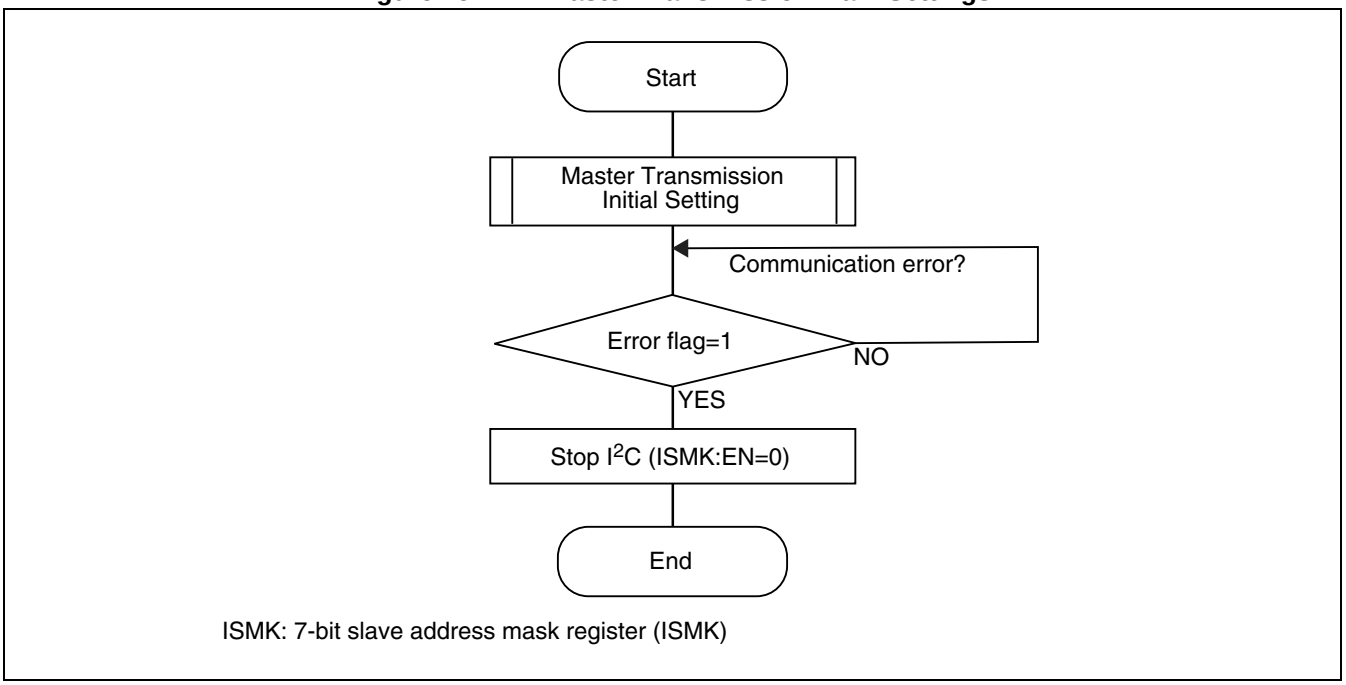
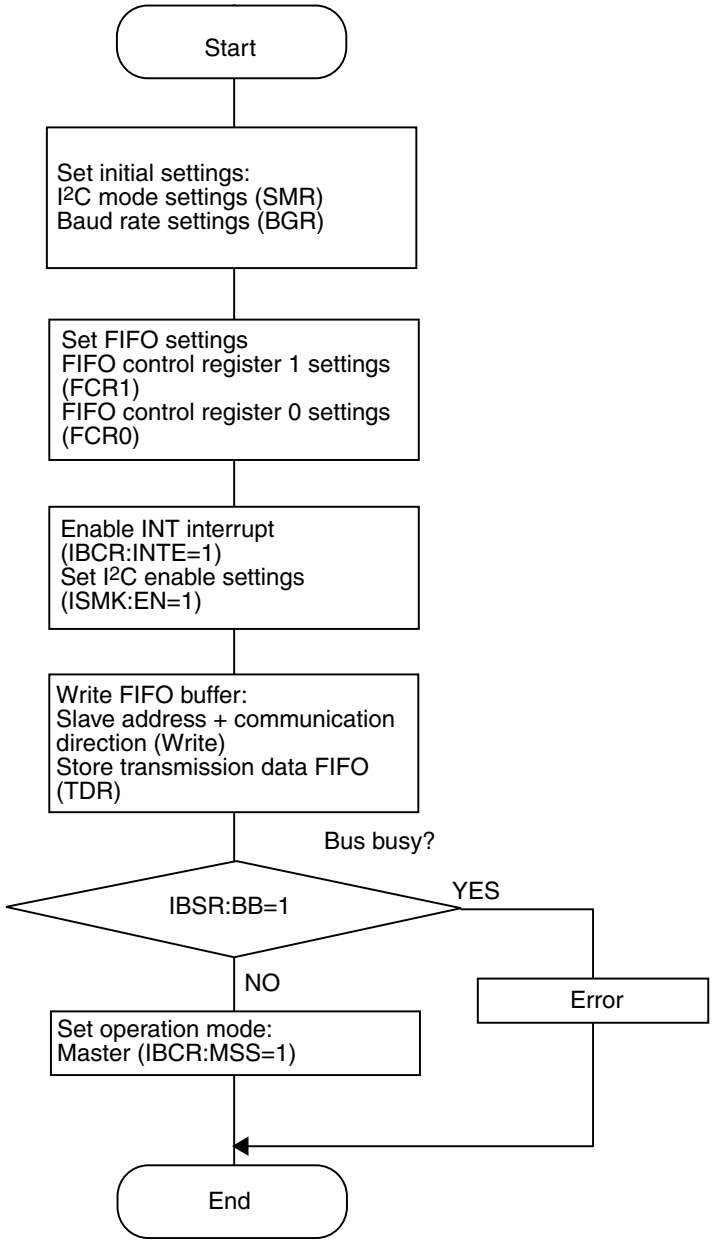
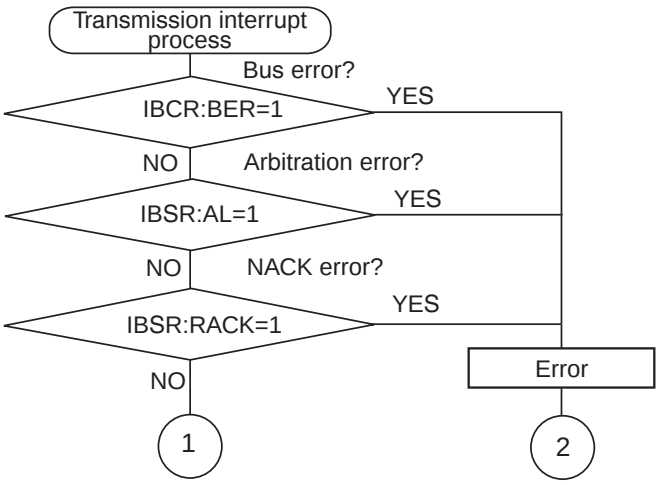


Figure 15.22-8 Master Transmission Initial Settings



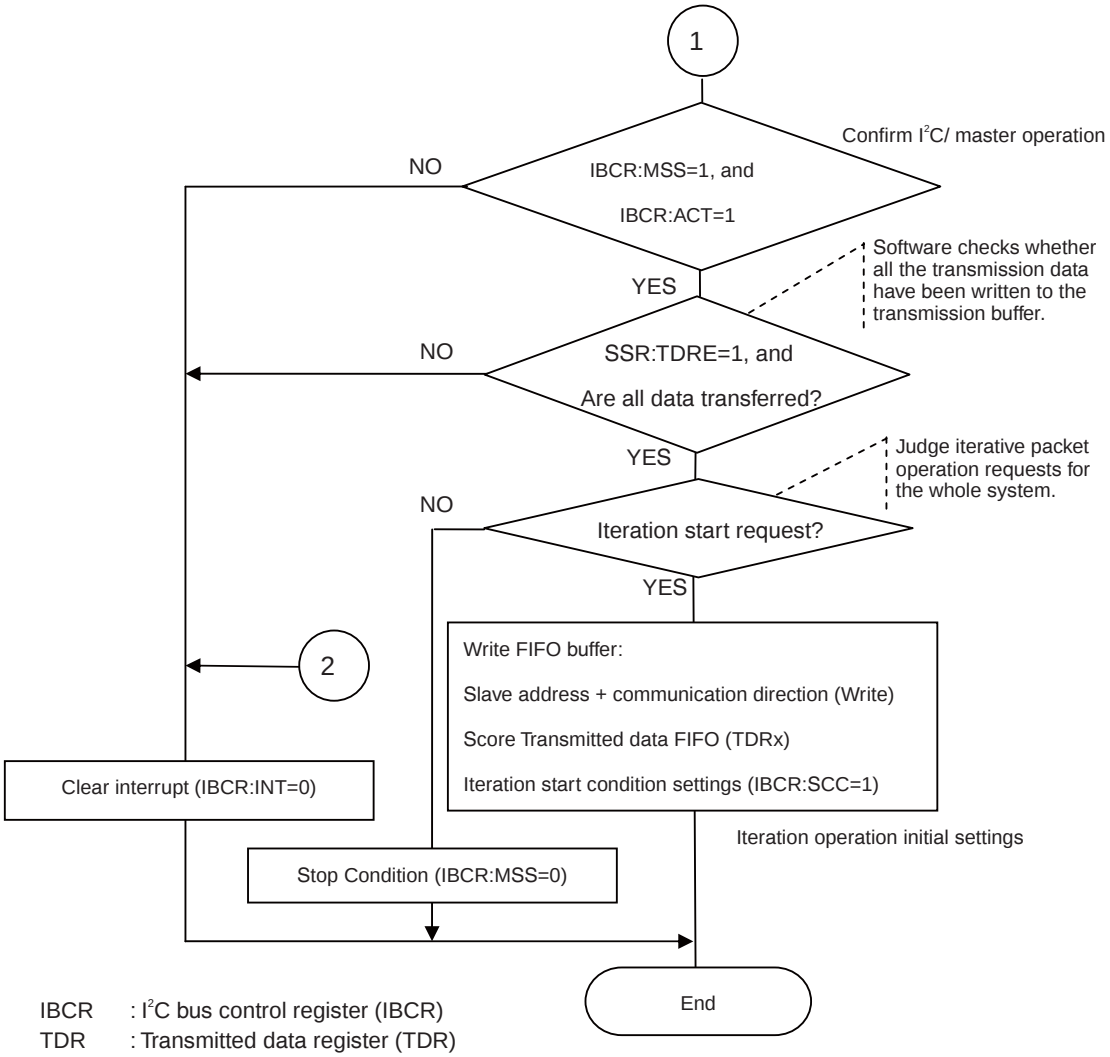
SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)
TDR: Transmitted data register (TDR)

Figure 15.22-9 Master Transmission Interrupt Process



IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)

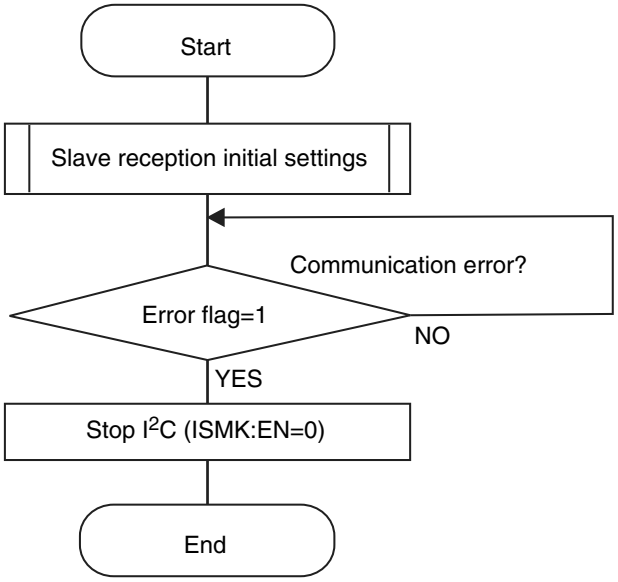
* For actual error handling, please judge each status error flag and handle each error by considering your systems.



IBCR : I2C bus control register (IBCR)
TDR : Transmitted data register (TDR)
SSR : Serial status register (SSR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

Figure 15.22-10 Slave Reception Main Settings



ISMK: 7-bit slave address mask register (ISMK)

Figure 15.22-11 Slave Reception Initial Settings

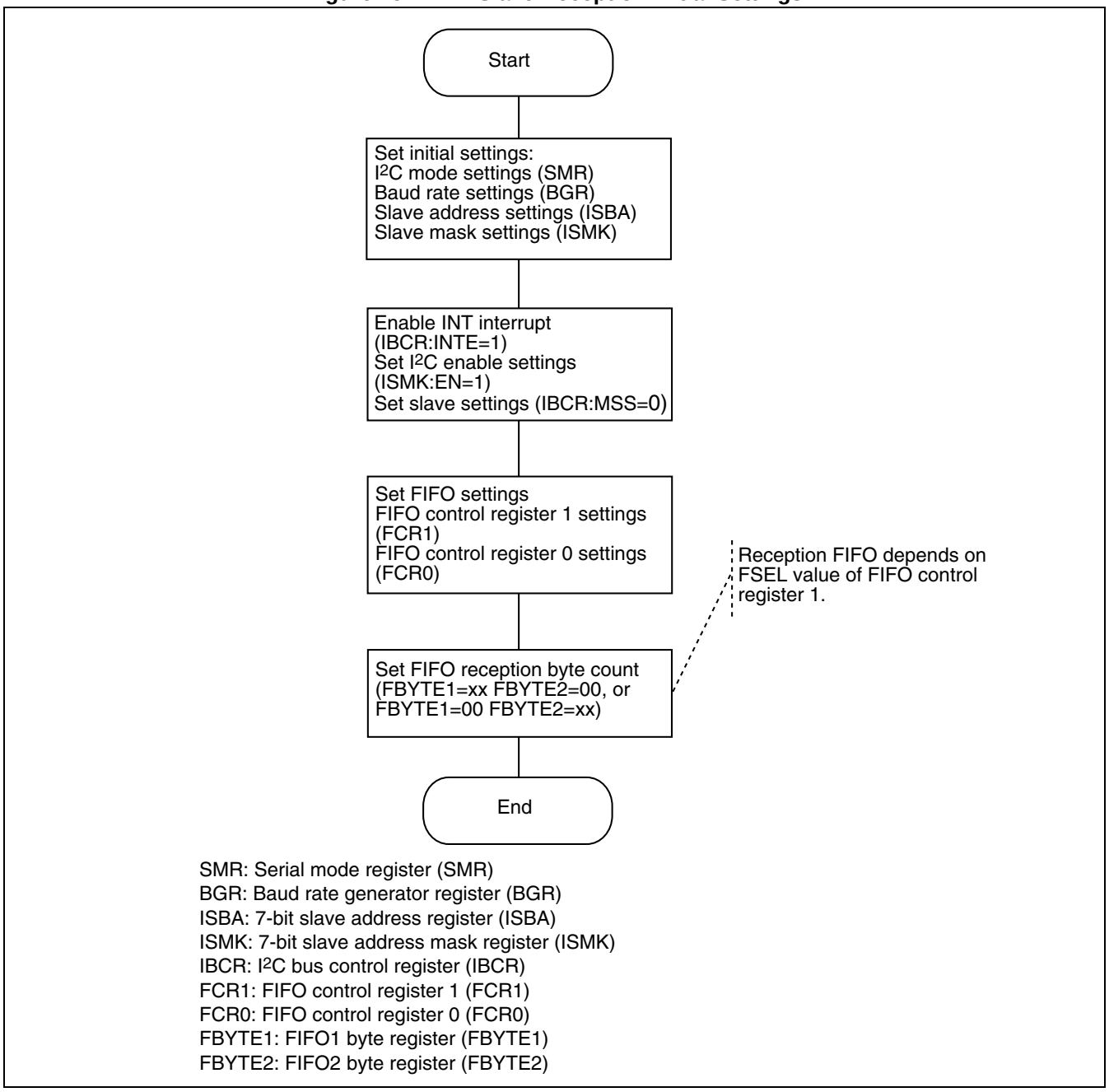
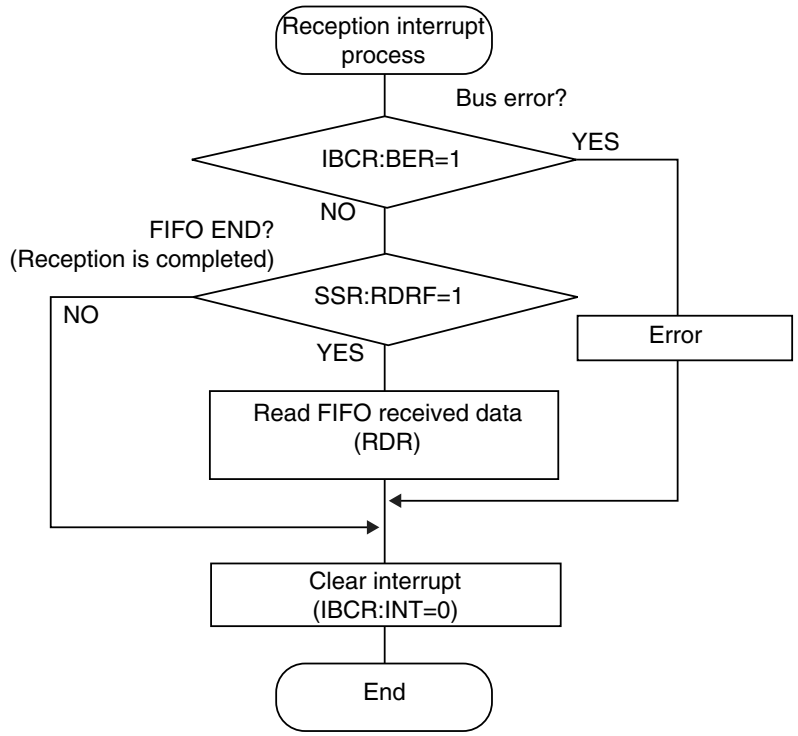


Figure 15.22-12 Slave Reception Interrupt Process



IBSR: I2C bus status register (IBSR)
SSR: Serial status register (SSR)
RDR: Received data register (RDR)
IBCR: I2C bus control register (IBCR)

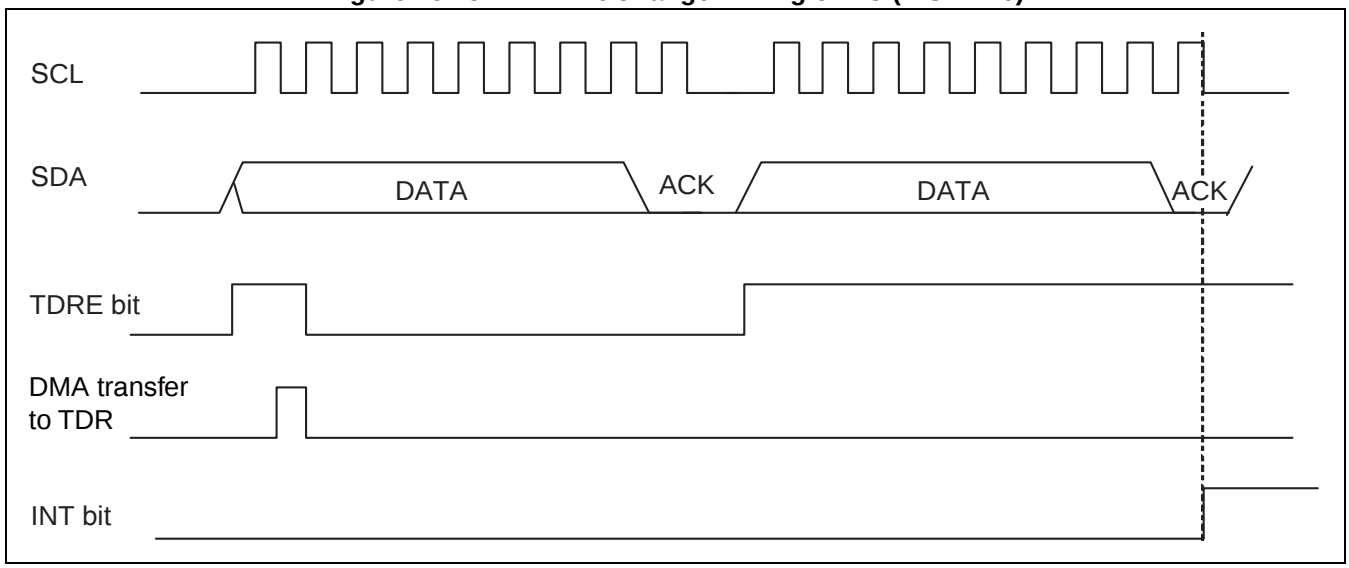
* For actual error handling, please judge each status error flag and handle each error by considering your systems.

15.23 Notes on I2C Mode

The notes for when you use the I2C mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.
- In I2C mode, if there is no valid data in transmission register (TDR), and transmission data empty flag bit (TDRE) is "1", the interrupt flag (INT) becomes "1" as shown in Figure 15.23-1 when the data on I2C bus for 9 bits (WSEL=0) or for 8 bits (WSEL=1) is transmitted. When the interrupt flag (INT) becomes "1" during DMA transfer, DMA transfer cannot be continued unless clearing the bit to "0" by software. (Common to master transmission, slave transmission, master reception, and slave reception.)

Figure 15.23-1 INT Bit Change Timing of I2C (WSEL= 0)



To perform DMA transfer in I2C mode, since the specification is as shown above, such operations listed below are required for performing DMA transfer to TDR before the interrupt flag (INT) becomes "1". Below operations are possible to perform to prioritize DMA transfer of I2C.

- Use DMA which has a higher priority (channel number is small). It is enabled to use by fixing the priority setting bit (AT=0).
- Set the value of DMA-halt by interrupt level bit as small as possible (LVL4-LVL0 bit in DILVR register).

- In case of writing the transmission data to transmission data register (TDR) by DMA transfer after transmission data empty flag (SSR: TDRE) becomes "1", or writing the data by software confirming the transmission data empty flag (SSR:TDRE), transmission data empty flag (SSR:TDRE) may not become "0". Therefore, the transmission data should be written before SCL in ACK field falls. There are no restrictions on writing the transmission data by software after the interrupt flag (IBCR:INT) becomes "1".

  When performing DMA transfer or sending the data by software confirming the transmission data empty flag (SSR:TDRE), please follow below procedures if the data cannot be written before SCL in ACK field falls.

  - Setting

    Set the timing of interrupt flag (IBCR:INT) becoming "1" to the 8th bit (WSEL=1).

  - Procedures

    To transmit or receive data by master, the following procedures are required. To transmit or receive data by slave, it is not required to perform the following.

    1. Write the first byte (slave address) to the transmission data register by software.
    2. Set to 8-bit for wait selection (IBCR:WSEL="1" write) at the same time that master is started (IBCR:MSS="1" write).
    3. After sending the first byte, the interrupt flag (IBCR:INT) becomes "1". Write the second byte to transmission data register (TDR) by software after confirming ACK response (IBSR:RACK="0"). Set the DMAC, and activate DMA transfer, then write "0" to interrupt flag (IBCR:INT).
    4. After transmission and reception are completed, terminate the master (IBCR:MSS="0" write) or reboot (IBCR:SCC="1" write).

CHAPTER 16

DMAC (DMA CONTROLLER)

This chapter gives an overview of the DMA controller (DMAC) and describes its register configuration and its operations.

16.1 Overview of DMA Controller

This module is used to carry out the DMA (Direct Memory Access) transfer in the FR family devices.
The DMA transfer controlled by this module enables various data to be transferred quickly without using CPU, resulting into an improvement of the system performance.

■ Hardware Configuration

This module is composed of the following items:

- Five independent DMA channels
- Five independent channels access control circuit
- 20-bit address register (reload selectable: ch.0 to ch.3)
- 24-bit address register (reload selectable: ch.4)
- 16-bit transfer count register (reload selectable: one for each channel)
- 4-bit block count register (one for each channel)
- Two-cycle transfer

■ Main Functions

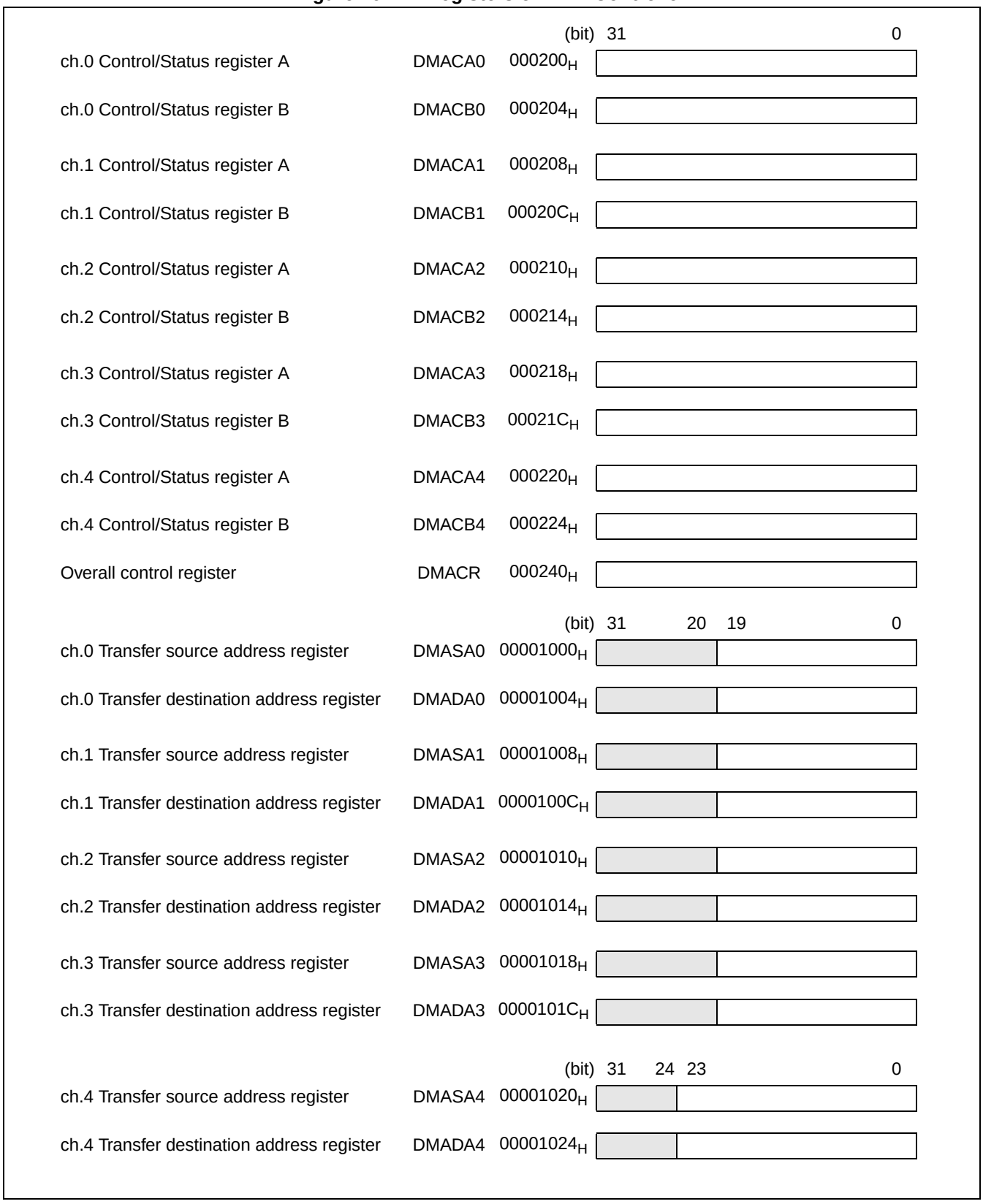
The data transfer in this module has the following functions:

- Data can be transferred dependently from multiple channels (5 channels)
  - Priority (ch.0 > ch.1 > ch.2 > ch.3 > ch.4)
  - Priority can be rotated between ch.0 and ch.1
  - DMAC activation source
  - Internal peripheral request
  - Software request (register programming)
  - Transfer mode
  - Burst transfer/step transfer/block transfer
  - Addressing mode: 20-bit (24-bit) address specification (increased/reduced/fixed) (range of change in address is ±1, 2, 4 fixed)
  - Data type: byte/half word/word length
  - Single shot/reload selectable

MB91313A Series

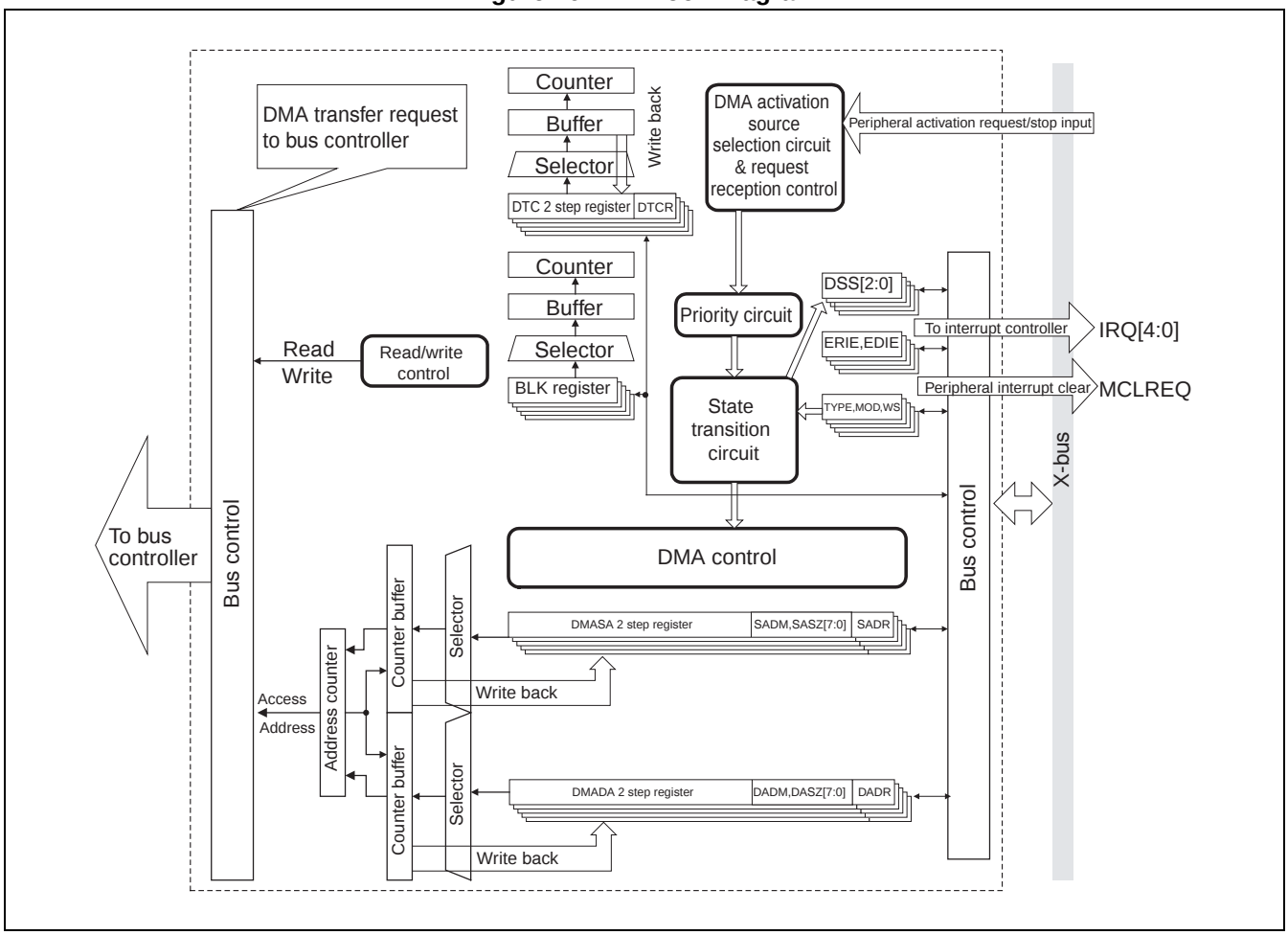
■ Registers of DMA Controller

Figure 16.1-1 Registers of DMA Controller



■ Block Diagram

Figure 16.1-2 Block Diagram



■ Notes on Setting Register

If setting this DMAC, there is a bit that needs to be executed when DMA is stopped. If bit is set during the operation (transfer), this DMAC may not normally operate.

The * mark indicates that the bit effects the operation if set during the DMAC transfer. This bit should be rewritten when the DMCA transfer is stopped (is not permitted to activate or halted).

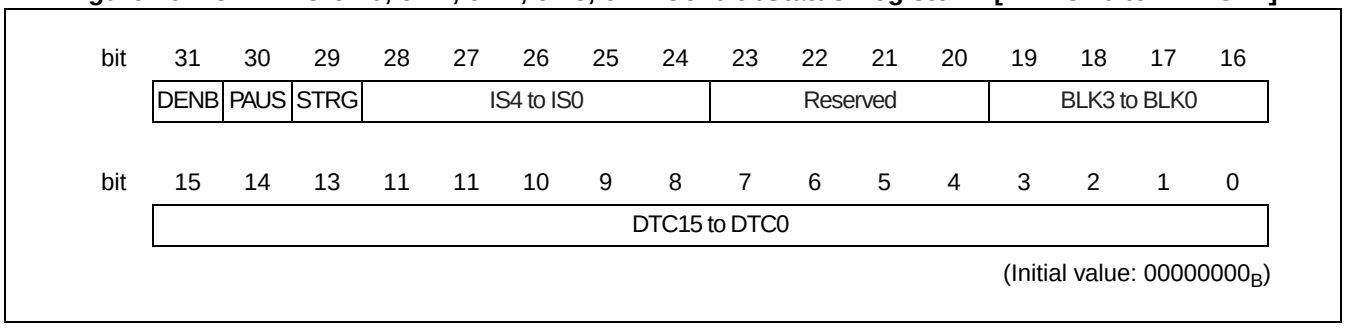
If it is set when the DMA transfer is not permitted to activate (DMACR:DMAE=0 or DMACA:DENB=0), the setting is enabled after the activation is permitted.

If it is set when the DMA transfer is halted (DMACR:DMAH[3:0] ≠ 0000_B or DMACA:PAUS=1), the setting is enabled after the halt is cancelled.

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 Control/Status Register A [DMACA0 to DMACA4]

DMACA0 to DMACA4 are the registers that control the operations of each of the DMAC channels, and they exist independently for each of the channels. The bit functions are as follows:

Figure 16.1-3 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 Control/Status Register A [DMACA0 to DMACA4]



[bit31] DENB (Dma ENaBle): DMA operation enable bit

Corresponds to each of the transfer channels and enables/disables the DMA transfer activation.

The activated channel starts the DMA transfer when the transfer request occurs and is accepted.

Any transfer requests, if occurred on the channel that is disabled to activate, are invalid.

If the activated channel's transfer is finished after repeated by specified counts, this bit turns to "0" and the transfer stops.

Writing "0" to this bit forcibly stops the transfer. However, be sure to halt DMA in the PAUS bit [DMACA bit30] before forcibly stopping the operation (write "0"). If it is stopped forcibly without halting it, DMA is stopped but the transfer data is not guaranteed. To check if it is stopped, see the DSS[2:0] bit (DMACB:bit18 to bit16).

DENB	Function
0	Corresponding channel DMA operation disabled (Initial value)
1	Corresponding channel DMA operation enabled

- If the stop request is accepted when reset: Initialized to "0".
- This bit is readable and writable.
- If operations of all of the channels are disabled with the bit15:DMAE, a bit in the DMAC overall control register DMACR, writing "1" to this bit is invalid and the stop status remains. If the operation is disabled with the bit mentioned above when the operation is enabled by this bit, this bit turns to "0" and the transfer is stopped (forcibly stopped).

[bit30] PAUS (PAUSe): Instruction for pause

Pauses the DMA transfer for the corresponding channel. If this bit is set, the DMA transfer is not executed before this bit is cleared again (When DMA is paused, the DSS bit is "$1XX_B$").

If this bit is set before activation, it keeps paused.

A transfer request that occurs when this bit is set can be accepted, but the transfer is not started unless this bit is cleared (see "■ Acceptance and Transfer of Transfer Request").

PAUS	Function
0	Corresponding channel DMA operation enabled (Initial value)
1	Corresponding channel DMA paused

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit29] STRG (Software TRiGger): Transfer request

Generates the DMA transfer request for the corresponding channel. If "1" is written to this bit, a transfer request occurs when writing to the register is completed, starting the transfer to the corresponding channel. However, if the corresponding channel is not activated, any operation to this bit is disabled.

Note: If the channel is activated by writing to the DMAE bit at the same time when a transfer request occurs from this bit, the transfer request is enabled and the transfer is started. The transfer request is enabled just when "1" is written to the PAUS bit, but the DMA transfer is not started until the PAUS bit is returned to "0".

STRG	Function
0	Ignored
1	DMA activation request

- When reset: Initialized to "0".
- Read value is always "0".
- The write value only works on "1". "0" does not effect the operations.

[bit28 to bit24] IS4 to IS0 (Input Select) *: Transfer source selection

Select the transfer request source as follows. However, the software transfer request by the STRG bit function is enabled regardless of this setting.

IS	Function	Transfer stop request
00000_B	Software transfer request only	No
00001_B to 01111_B	Setting disabled	
10000_B	UART0 (Reception complete)	Yes
10001_B	UART1 (Reception complete)	
10010_B	UART2 (Reception complete)	
10011_B	UART0 (Transmission complete)	No
10100_B	UART1 (Transmission complete)	
10101_B	UART2 (Transmission complete)	
10110_B	Setting disabled	
10111_B	Setting disabled	
11000_B	Setting disabled	
11001_B	Setting disabled	
11010_B	Setting disabled	
11011_B	Setting disabled	
11100_B	PPG0	
11101_B	PPG1	
11110_B	PPG2	
11111_B	PPG3	

- When reset: initialized to "00000_B".
- These bits are readable and writable.

Notes:
- If setting DMA activation with the interrupt of peripheral functions (IS=$1XXXX_B$), the selected function should disable the interrupt in the ICR register.
- When the software transfer request causes the DMA transfer to be activated if the DMA activation with the interrupt of peripheral functions is set, the source is cleared for the appropriate peripherals after the transfer is completed. Therefore, the original transfer request can be cleared, so do not activate the transfer from the software transfer request if the DMA activation with the interrupt of peripheral functions is set.

[bit23 to bit20] Reserved: Reserved bits

- The read value is fixed as "0000_B". The write is disabled.

[bit19 to bit16] BLK3 to BLK0 (BLocK size): Block size specification

Specifies the block size for the corresponding channel at the time of the block transfer. The value specified in these bits is the number of words in a transfer unit at a time (the number of the repeats of the data width setting). Set 0001_B (size 1) if the block transfer is not to be performed.

BLK	Function
$XXXX_B$	The block size specification for the corresponding channel

- When reset: Initialized to "0000_B".
- These bits are readable and writable.
- If all the bits are set to "0", the block size is 16 words.
- When reading, the block size (reload value) is always read.

[bit15 to bit0] DTC15 to DTC0 (Dma Terminal Count register) *: Transfer count register

This is a register that stores the transfer counts. Each register has 16-bit length.

Every register has its reload register. If reloading of the transfer count register is used for the enabled channel, the initial value is automatically returned to the register when the transfer is completed.

DTC	Function
$XXXX_H$	The transfer counts specification for the corresponding channel

When the DMA transfer is activated, this register's data is stored in the counter buffer for the transfer counter for DMA and is counted by -1 (subtracted by 1) per transfer unit. DMA is finished by writing back the content of the counter buffer to this register when the DMA transfer is finished. Therefore, you cannot read the specified value of the transfer counts during the DMA operation.

- When reset: Initialized to "0000_H".
- These bits are readable and writable. Be sure to use the half word length or word length to access to DTC.
- The read value is the count value. The reload value cannot be read.

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 Control/Status Register B [DMACB0 to DMACB4]

DMACB0 to DMACB4 are the registers that control the operation of each of the DMAC channels, and they exist independently for each of the channels. The bit functions are as follows

Figure 16.1-4 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 Control/Status Register B [DMACB0 to DMACB4]

Address	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ch.0 000204_H ch.1 $00020C_H$ ch.2 000214_H ch.3 $00021C_H$ ch.4 000224_H		TYPE1,TYPE0		MOD1,MOD:0		WS1,WS0		SADM	DADM	DTCR	SADR	DADR	ERIE	EDIE	DSS2 to DSS0		
	bit	15	14	13	11	11	10	9	8	7	6	5	4	3	2	1	0
		SASZ7 to SASZ0								DASZ7 to DASZ0							

(Initial value: 00000000_B)

[bit31, bit30] TYPE1, TYPE0 (TYPE) *: Transfer type setting

Select the operation type for the corresponding channel as follows:

Two-cycle transfer mode: This is a mode in which the read and write operations are transferred repeatedly by the transfer counts after setting the transfer source address (DMASA) and the transfer destination address (DMADA).

TYPE1, TYPE0	Function
00_B	Two-cycle transfer (Initial value)
01_B	Setting disabled
10_B	Setting disabled
11_B	Setting disabled

- When reset: Initialized to "00_B".
- These bits are readable and writable.
- Be sure set the value to "00_B".

[bit29, bit28] MOD1, MOD0 (MODe) *: Transfer mode setting

Select the operation mode for the corresponding channel as follows:

MOD1, MOD0	Function
00_B	Block/step transfer mode (Initial value)
01_B	Burst transfer mode
10_B	Setting disabled
11_B	Setting disabled

- When reset: Initialized to "00_B".
- These bits are readable and writable.

[bit27, bit26] WS1, WS0 (Word Size): Transfer data width selection

Select the transfer data range for the corresponding channel. Transfer the data by the specified number of times on a data-range basis specified in this register.

WS1, WS0	Function
00_B	Transfer on a byte basis (Initial value)
01_B	Transfer on a half word basis
10_B	Transfer on a word range basis
11_B	Setting disabled

- When reset: Initialized to "00_B".
- These bits are readable and writable.

[bit25] SADM (Source-ADdr. count-Mode select) *: Transfer source address count mode specification

Specifies the processing of the source address per transfer for the corresponding channel.

Address is incremented/decremented after each transfer is completed based on the specified source address count range (SASZ) and the next accessing address is written to the corresponding address register (DMASA) when the transfer is completed.

Therefore, the transfer source address register is not updated until the DMA transfer is finished.

To fix the address, specify this bit to "0" or "1" and set the address count range (SASZ, DASZ) to "0".

SADM	Function
0	The transfer source address is incremented. (Initial value)
1	The transfer source address is decremented.

- When reset: Initialized to "0".
- These bits are readable and writable.

[bit24] DADM (Destination-ADdr. Count-Mode select) *: Transfer destination address count mode specification

Specifies the processing of the destination address per transfer for the corresponding channel.

Address is incremented/decremented after each transfer is completed based on the specified destination address count width (DASZ) and the next accessing address is written to the corresponding address register (DMADA) when the transfer is completed.

Therefore, the transfer destination address register is not updated until the DMA transfer is finished.

To fix the address, specify this bit to "0" or "1" and set the address count width (SASZ, DASZ) to "0".

DADM	Function
0	The transfer destination address is incremented. (Initial value)
1	The transfer destination address is decremented.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit23] DTCR (DTC-reg. Reload) *: Transfer count register reload specification

Controls the reload function of the transfer count register for the corresponding channel.

If the reload operation is enabled by this bit, the count register value is returned to the initial value and stopped when the transfer is finished, resulting into a status of waiting for a transfer request (activation request by the STRG or IS setting). (If this bit is 1, the DENB bit is not cleared.)

Setting DENB=0 or DMAE=0 forcibly stops the operation.

If the reload operation of the counter is disabled, specifying the reload in the address register makes the operation single shot which would be stopped when the transfer is finished. In this case, the DENB bit is cleared.

DTCR	Function
0	Disable the reload of the transfer count register (Initial value)
1	Enable the reload of the transfer count register

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit22] SADR (Source-ADdr.-reg. Reload) *: Reload specification of the transfer source address register

Controls the reload function of the transfer source address register for the corresponding channel.

If the reload operation is enabled by this bit, the transfer source address register value is returned to the initial value when the transfer is finished.

If the reload operation of the counter is disabled, specifying the reload in the address register makes the operation single shot which would be stopped when the transfer is finished. In this case, the address register value is stopped with the initial setting value having been reloaded.

If the reload operation is disabled by this bit when the transfer is finished, the address register value is the next access address to the last address (or incremented address if the address increment is specified).

SADR	Function
0	Disable the reload of the transfer source address register (Initial value)
1	Enable the reload of the transfer source address register

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit21] DADR (Dest.-ADdr.-reg. Reload) *: Reload specification of the transfer destination address register

Controls the reload function of the transfer destination address register for the corresponding channel.

If the reload operation is enabled by this bit, the transfer destination address register value is returned to the initial value when the transfer is finished.

Other details of the function is equivalent to the content of bit22:SADR.

DADR	Function
0	Disable the reload of the transfer destination address register (Initial value)
1	Enable the reload of the transfer destination address register

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit20] ERIE (ERror Interrupt Enable)*: Error interrupt output enabled

Controls the interrupt occurrence when finished due to an occurrence of error. The content of occurred error is indicated in DSS2 to DSS0. Note that this interrupt occurs only for particular finish sources, not for every finish source (see the explanation of the DSS2 to DSS0 bits).

ERIE	Function
0	Error interrupt request output disabled (Initial value)
1	Error interrupt request output enabled

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit19] EDIE (EnD Interrupt Enable) *: End interrupt output enable

Controls the interrupt occurrence when normally finished.

EDIE	Function
0	End interrupt request output disabled (Initial value)
1	End interrupt request output enabled

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit18 to bit16] DSS2 to DSS0 (Dma Stop Status)*: Transfer stop source display

Displays a 3-bit code (end code) that indicates the source of the DMA transfer stop/finish for the corresponding channel. The contents of the end code are as follows:

DSS2	Function	Interrupt generation
0	Initial value	No
1	Pausing DMA (DMAH, PAUS bit, interrupt, and so on)	No

DSS1, DSS0	Function	Interrupt generation
00_B	Initial value	No
01_B	-	No
10_B	Transfer stop request	Error
11_B	Normal termination	End

The transfer stop request is set only if the request from peripheral circuit is used.

Note: The "interrupt generation" column indicates the type of possible interrupt requests.

- When reset: Initialized to "000_B".
- Cleared by writing "000_B".
- These bits are readable/writable, but only "000_B" can be written to this bit.

[bit15 to bit8] SASZ7 to SASZ0 (Source Addr count SiZe) *: Transfer source address count size specification

Specifies the increment/decrement width of the transfer source address (DMASA) per transfer for the corresponding channel. The value set to this bit is the address increment/decrement width per transfer. The address increment/decrement depends on the instruction of the transfer source address count mode (SADM).

SASZ7 to SASZ0	Function
00_H	Address fixed
01_H	Transfer on a byte basis
02_H	Transfer on a half word basis
04_H	Transfer on a word basis
Others	Setting disabled

- When reset: Initialized to "00_H".
- These bits are readable and writable.
- If set to other than address fixed, only the same transfer unit as the transfer data width (WS) can be set.

[bit7 to bit0] DASZ7 to DASZ0 (Des Addr count SiZe) *: Transfer destination address count size specification

Specifies the increment/decrement range of the transfer destination address (DMADA) per transfer for the corresponding channel. The value set to this bit is the address increment/decrement width per transfer. The address increment/decrement depends on the instruction of the transfer destination address count mode (DADM).

DASZ7 to DASZ0	Function
00_H	Address fixed
01_H	Transfer on a byte basis
02_H	Transfer on a half word basis
04_H	Transfer on a word basis
Others	Setting disabled

- When reset: Initialized to "00_H".
- These bits are readable and writable.
- If set to other than address fixed, only the same transfer unit as the transfer data width (WS) can be set.

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 Transfer Source/Destination Address Setting Registers [DMASA0 to DMASA4/DMADA0 to DMADA4]

DMASA0 to DMASA4/DMADA0 to DMADA4 are the registers that control the operations of each of the DMAC channels, and they exist independently for each of the channels. The bit functions are as follows:

Figure 16.1-5 DMAC Transfer Source/Destination Address Setting Registers

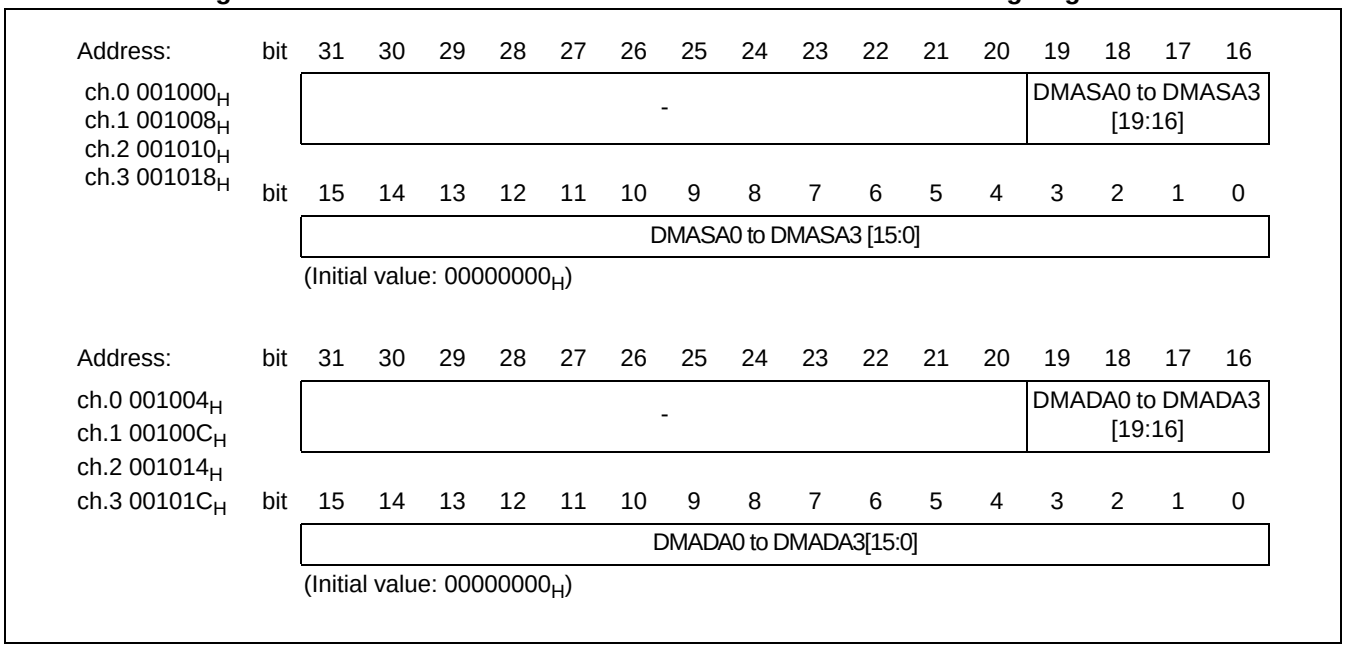
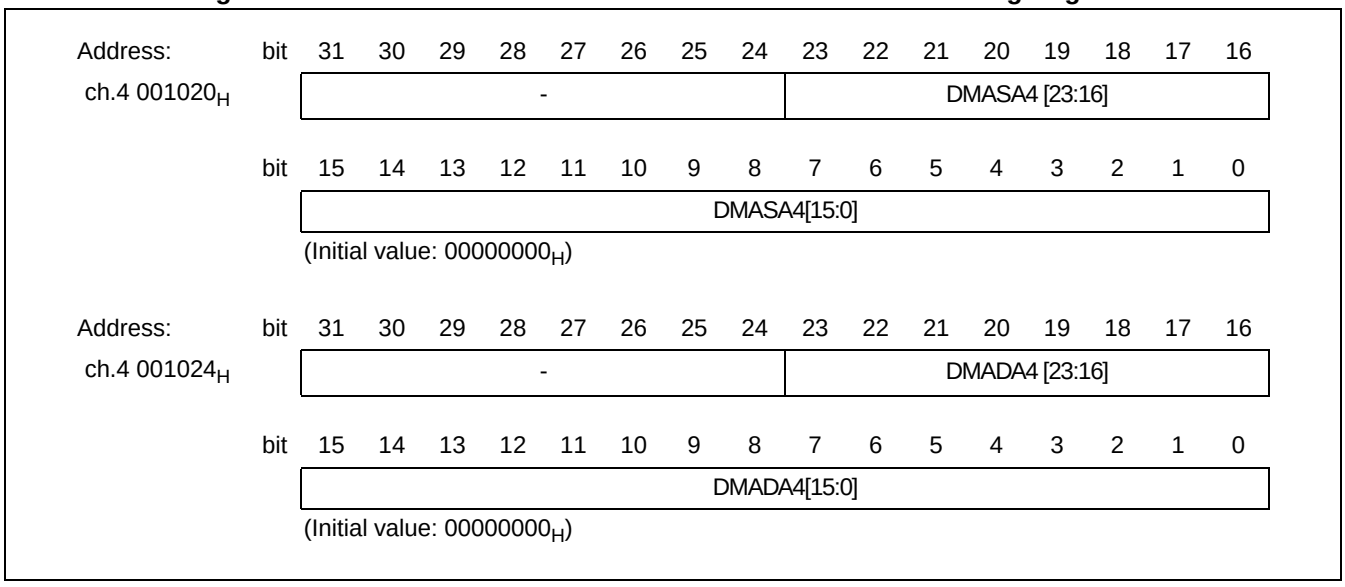


Figure 16.1-6 DMAC Transfer Source/Destination Address Setting Registers



These are a set of registers that store the transfer source/destination addresses. ch.0 to ch.3 have 20-bit length, ch.4 has 24-bit length.

[bit31 to bit0] DMASA (DMA Source Addr) *: Transfer source address setting

Sets the transfer source address.

[bit31 to bit0] DMADA (DMA Destination Addr) *: Transfer destination address setting

Sets the transfer destination address.

When the DMA transfer is activated, this register's data is stored in the counter buffer for the address counter for DMA and the address is counted per transfer based on the setting. DMA is finished by writing back the content of the counter buffer to this register when the DMA transfer is finished. Therefore, you cannot read the address counter value during the DMA operation.

Every register has its reload register. If reloading of the transfer source/destination address register is used for the enabled channel, the initial value is automatically returned to the register when the transfer is completed. In this case, other address registers are not effected.

- When reset: Initialized to "00000000_H".
- These bits are readable and writable. With this register, be sure to use 32 bit data for access.
- The read value is an address value before transfer when the transfer is being executed, and the next access address value when the transfer is finished. The reload value cannot be read. Therefore, the transfer address cannot be read at real time.
- Set "0" to a nonexistent upper bit.

Note:

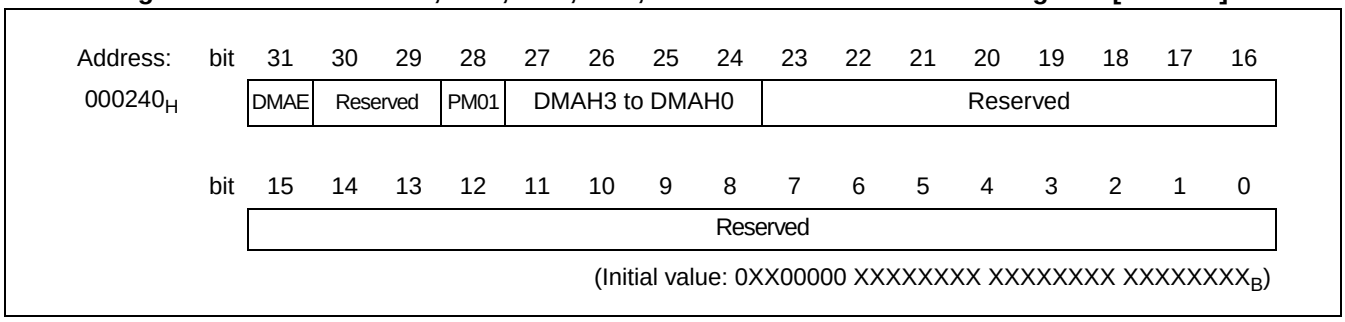
Do not use this register to set the register of DMAC itself. The DMA transfer cannot be executed to the register of DMAC itself.

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 DMAC Overall Control Register [DMACR]

DMACR is a register that controls the operation for the DMAC 5 channels. With this register, be sure to use byte length for access.

The bit functions are as follows:

Figure 16.1-7 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 DMAC Overall Control Register [DMACR]



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

Controls the operations of all the DMA channels.

If the DMA operation is disabled by this bit, the transfer operations of all the channels are disabled regardless of the settings of activation/stop and operation statuses of the channels. The channel in which a transfer is in progress turns down the request and stops the transfer at the block boundary. Any activation operations for each channel, if disabled, are invalid.

If the DMA operation is enabled by this bit, the activation/stop operation is available for each channel. The channels will not be activated by enabling the DMA operation with this bit.

Writing "0" to this bit forcibly stops the transfer. However, be sure to halt DMA in the DMAH[3:0] bits (DMACR bit27 to bit24) before forcibly stopping the operation (write "0"). If it is stopped forcibly without halting it, DMA is stopped but the transfer data is not guaranteed. To check if it is stopped, see the DSS[2:0] bit (DMACB:bit18 to bit16).

DMAE	Function
0	All the channels' DMA operation disabled (Initial value)
1	All the channels' DMA operation enabled

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit28] PM01 (Priority Mode ch.0, ch.1 robin): Channel priority rotation

Set this bit if the priority of ch.0. ch.1 should be rotated for each transfer.

PM01	Function
0	Priority fixed (ch.0 > ch.1) (Initial value)
1	Priority rotated (ch.1 > ch.0)

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit27 to bit24] DMAH3 to DMAH0 (DMA Halt): DMA pause

Controls pausing of all the DMA channels. If these bits are set, all the channels' DMA transfers are not executed before these bits are cleared again.

If these bits are set before activation, the channels keep paused.

When these bits are set, all the transfer requests that occur on the channel with the DMA transfer enabled (DENB=1) are valid and the transfer is started by clearing these bits.

DMAH3 to DMAH0	Function
0000_B	All the channels, DMA operation enabled (Initial value)
Other than 0000_B	All the channels' DMA halted

- When reset: Initialized to "0".
- These bits are readable and writable.

[bit30, bit29, bit23 to bit0] Reserved: Reserved bits

- The read values are undefined.

16.2 Operations of DMA Controller

This section explains operation of a multifunction DMA controller that controls the high performance of data transfer without receiving any instructions from CPU.

■ Main Operations

- Each of the transfer channels independently sets each of the functions.
- If the activation is enabled, the channels do not execute the transfer operation until the specified transfer request is detected.
- By detecting a transfer request, the DMA transfer request is output to the bus controller, and the bus controller controls the operation to obtain the bus right and start the transfer.
- The transfer is performed according to the sequence based on the mode setting specified independently for each channel.

■ Transfer Mode

Each of the DMA channels executes the transfer operation according to the transfer mode specified in the MOD[1:0] bit in each of the DMACB registers.

● Block/step transfer

Transfers one block transfer unit for each transfer request. DMA stops the transfer request to the bus controller until the next transfer request is accepted.

1 block transfer unit: Specified block size (DMACA:BLK[3:0])

● Burst transfer

Repeats the transfer for a transfer request until the specified number of transfers is completed.

Specified transfer count: Block size × Transfer count
(DMACA:BLK[3:0] × DMACA:DTC[15:0])

■ Transfer Type

● Two-cycle transfer (normal transfer)

The DMA controller operates as a unit of the read and write operations.

The data is read from the address of the transfer source register, and written to the address of the transfer destination register.

■ Transfer Address

The following addressing is available. The address is set independently for each channel's transfer source/ destination.

● Specifying address in two-cycle transfer

Access to the value as the address that is read from the register (DMASA, DMADA) in which the address is set in advance.

After the transfer request is accepted, DMA starts the transfer after storing the address from the register to the temporary storage buffer.

The address to be accessed next is generated (addition, subtraction, or fix can be selected) at the address counter for each transfer (access) and returned to the temporary storage buffer. The content of this temporary storage buffer is written back to the register (DMASA, DMADA) for each completion of a block transfer unit.

Therefore, the address register (DMASA, DMADA) value is updated only on a block-transfer-unit basis, and the address cannot be known at real time.

■ Transfer Count and Transfer Termination

● Transfer count

The transfer count register is decremented (by 1) for each completion of a block transfer unit. If the transfer count register is "0", then the specified number of transfers is finished, resulting into a stop or re-activation after the end code is displayed.

The transfer count register value is updated only on a block-transfer unit basis, similarly to the address register.

If the reload of the transfer count register is disabled, the transfer is terminated. If it is enabled, the register value is initialized, resulting into a status where the transfer is waited (DMACB:DTCR).

● Transfer termination

The sources of the transfer termination are as follows. When terminated, one of the source is displayed as a end code (DMACB:DSS[2:0]).

- End of specified number of transfers (DMACA:BLK[3:0] × DMACA:DTC[15:0]) → Normal Termination
- Occurrence of transfer stop request from peripheral circuit → Error
- Occurrence of reset → Reset

The transfer stop source is displayed (DSS) for each of the termination sources, and the transfer termination interrupt/error interrupt can be generated,

MB91313A Series

16.3 Setting of Transfer Request

There are two types of the transfer request that activates the DMA transfer. The software request can always be used regardless of other request settings.

■ Built-in Peripheral Request

Generates a transfer request due to an occurrence of the interrupt from the built-in peripheral circuit.

Set which of the peripheral interrupts generates a transfer request for each channel (DMACA: IS[4:0]=1XXXX$_B$).

Note: The interrupt request used for a transfer request can be seen as an interrupt request to CPU, so disable the interrupt for the interrupt controller setting (ICR register).

■ Software Request

Generates a transfer request by writing to the trigger bit in the register (DMACA:STRG).

It is independent of the transfer request shown above, and it can always be used.

If a software request is issued simultaneously with activation (enabling of transfer), the DMA transfer request is immediately output to the bus controller to start the transfer.

Note: If a software request is issued to the channel in which built-in peripheral request is set, the source is cleared for the corresponding peripheral after the transfer is completed. Therefore, the original transfer request can be cleared, so do not issue the software request.

16.4 Transfer Sequence

You can independently set the transfer type and transfer mode to determine the operation sequence after the DMA transfer is activated for each channel (setting of DMACB:TYPE[1:0], MOD[1:0]).

■ Selecting the Transfer Sequence

The following sequence can be selected depending on the register setting:

Burst two-cycle transfer

Block/step two-cycle transfer

■ Burst Two-cycle Transfer

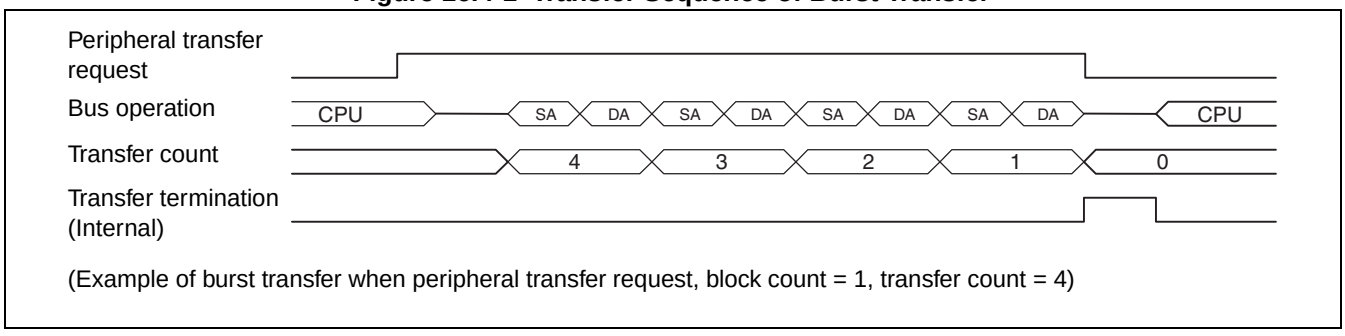
Repeats the specified number of transfers at one transfer source. For the two-cycle transfer, the transfer source/destination address can be specified with 20 bits (for ch.0 to ch.3) or 24 bits (for ch.4).

The transfer source can select the peripheral transfer request/software transfer request.

[Features of burst transfer]

- If one transfer request is accepted, the transfer is repeated until the transfer count register turns to "0". Number of transfers is block size × number of transfers (DMACA:BLK[3:0] × DMACA:DTC[15:0]).
- If another request occurs during the transfer, it is ignored.
- If the reload function of the transfer count register is enabled, the next transfer request is accepted after the transfer is finished.
- If another channel's transfer request with a higher priority is accepted during the transfer, the channel is switched at the boundary of the current block transfer unit, and it is not returned until the transfer request for the channel is cleared.

Figure 16.4-1 Transfer Sequence of Burst Transfer



(Example of burst transfer when peripheral transfer request, block count = 1, transfer count = 4)

■ Step/Block Transfer Two-cycle Transfer

For the step/block transfer (in which the transfer is executed for each transfer request at the specified number of blocks), the transfer source/destination address can be specified with 20 bits (for ch.0 to ch.3) or 24 bits (for ch.4).

● Step transfer

If "1" is set to the block size, the sequence turns to the step transfer sequence.

[Features of step transfer]

- If a transfer request is accepted once, the transfer is executed once and stopped after the transfer request is cleared (the DMA transfer request is turned down to the bus controller).
- If another request occurs during the transfer, it is ignored.
- If another channel's transfer request with a higher priority is accepted during the transfer, the channel is switched after the transfer is stopped and the new transfer is started. Priority in the step transfer works only if multiple transfer requests occur simultaneously.

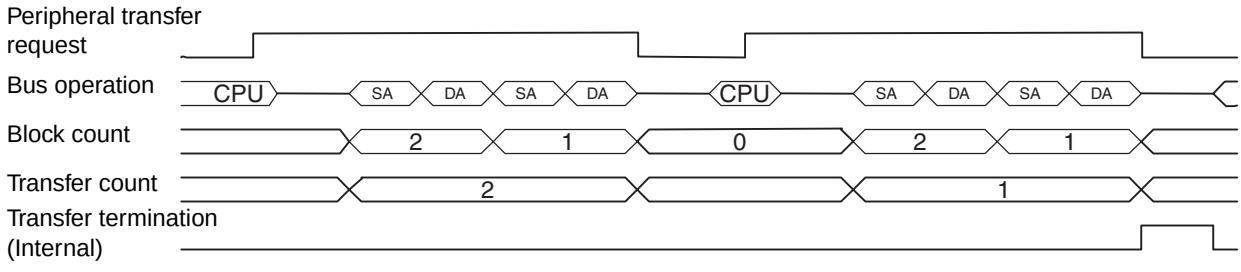
● Block transfer

If other value than "1" is set to the block size, the sequence turns to the block transfer sequence.

[Features of block transfer]

The operation is the same except one transfer unit is composed of multiple numbers (blocks) of the transfer cycle.

Figure 16.4-2 Transfer Sequence of Block Transfer



(Example of block transfer when peripheral transfer request, block count = 2, transfer count=2)

■ General DMA Transfer

● Block size

- One transfer unit of the transfer data is a collection of the data, number of which is specified in the block size specification register (× the data width).
- Since the data transferred in one transfer cycle is fixed to the value specified with the data width, a transfer unit is composed of the number of the transfer cycles with the block size specified value.
- If a transfer request with a higher priority is accepted during the transfer or if the transfer pause request is issued, even the block transfer is stopped only at the end of one transfer unit. This enables the data to be protected for the data block that does not want to be divided or paused, but the response can be downgraded if the block size is large.
- Though it is stopped immediately only when the reset occurs, the content of the data being transferred is not secured.

● Reload operation

This module enables you to set the three types of the reload function for each channel:

(1) Transfer count register reload function

After the specified number of transfers are finished, the initial value is reset in the transfer count register to wait for acceptance of an activation.

Set it if the whole transfer sequence is repeated.

If the reload is not specified, the count register value remains "0" after the specified number of transfers is completed, and the subsequent transfer is not performed.

(2) Reload function of the transfer source address register

After the specified number of transfers are finished, the initial value is reset in the transfer source address register.

Set if the transfer is repeated from the fixed area in the transfer source address area.

If the reload is not specified, the transfer source address register value is the next address after the specified number of transfers are completed. It is used if the address area is not fixed.

(3) Reload function of the transfer destination address register

After the specified number of transfers are finished, the initial value is reset in the transfer destination address register.

Set if the transfer is repeated to the fixed area in the transfer destination address area.

(Description shown below is the same as (2))

- Only If the reload function of the transfer source/destination register is enabled, the re-activation is not performed after the specified number of the transfers are completed. Each of the address register values is only reset.

Note: Special case of operation mode and reload operation

- If it is preferable that processing stops when data transfer ends and starts after input is detected again, do not specify reload.
- If the burst/block/step transfer mode is used for the transfer, the data is not transferred when the transfer is completed, until the transfer is halted after reloaded and then a new transfer request input is detected.

■ Addressing Mode

The transfer destination/source address is specified independently for each of the transfer channel.

The following methods are provided to specify the address. The transfer sequence should be used to set it.

● Address register specification

For the two-cycle transfer mode, set the transfer source address in the transfer source address setting register (DMASA) and the transfer destination address in the transfer destination address setting register (DMADA).

[Features of address register]

ch.0 to ch.3 are registers with 20-bit length, and ch.4 is a register with 24-bit length.

[Functions of address register]

- Read for each access and sent to the address bus.
- At the same time, the address for the next access is calculated with the address counter, and the address register is updated with the address of the calculation result.
- The address calculation is selected from addition/subtraction independently for each channel/transfer destination/transfer source. Increment/decrement width of the address depends on the value of the address count size specification register (DMACB:SASZ, DASZ).
- If the reload function is disabled, the calculated address is left in the last address in the address register after the transfer is completed.
- If the reload function is enabled, the initial value of the address is reloaded.

Notes:

- If an overflow or underflow occurs as a result of 20-bit or 24-bit length full address calculation, transfer on the channel continues. Set each channel to prevent an overflow or underflow from being generated.
- Do not set the address of the register of DMAC itself in the address register.

■ Data Type

The data length (data width) transferred in one transfer should be selected from one of the following:

- Byte
- Half word
- Word

Since the word boundary specification is kept for the DMA transfer, the different lower bit is ignored if the specified address does not match the data length in the transfer source/destination address specification.

- **Word** . . . . . Actual access address is 4 bytes in which the lower 2 bits start with "00_B".
- **Half word** . . . Actual access address is 2 bytes in which the lower 1 bit starts with "0_B".
- **Byte** . . . . . Actual access address matches the address specification.

If the lower bit of the transfer source address does not match that of the transfer destination address, the address as is set is output on the internal address bus, but the address is fixed on each of the transfer targets on the bus, based on the rule described above.

■ Transfer Count Control

The transfer count is specified within a range of 16-bit length (1 to 65536) at maximum. The transfer count specification value is set in the transfer count register (DMACA:DTC).

The register value is stored in the temporary storage buffer when the transfer is started, and decremented by the transfer counter. When this counter value turns to "0", it is detected as completion of specified number of the transfers, and the transfer for the channel is stopped or re-activation is waited (when the reload is specified).

[Features of transfer count registers]

- Each of the registers has 16-bit length.
- Every register has its reload register.
- If activated when the register value is "0_B", the transfer is repeated 65536 times.

[Reload operation]

- Available only if the register has the reload function and the function is enabled.
- The initial value of the count register is saved to the reload register when the transfer is activated.
- If the count turns to "0" using the transfer counter, the transfer is notified as completed, and the initial value is read from the reload register and written to the count register.

■ CPU Control

If the DMA transfer request is accepted, DMA issues the transfer request to the bus controller.

The bus controller passes the right to use the internal bus to DMA at the break of the bus operation to start the DMA transfer.

● DMA transfer and interrupt

- During DMA transfer, interrupts are generally not accepted until the transfer ends. And if the DMA transfer request occurs when the interrupt is being processed, the transfer request is accepted and the process of the interrupt is paused until the transfer is completed.
- If, as an exception, an interrupt request with a higher priority than the hold suppression level specified in the interrupt controller occurs, DMAC temporarily turns down the transfer request to the bus controller at the boundary of the transfer unit (1 block), and pauses the transfer until the interrupt request is cleared. The transfer request is internally kept during this time. If the interrupt request is cleared, then DMAC issues the transfer request to the bus controller to obtain the right to use the bus and resume the DMA transfer.

● DMA suppression

- If the interrupt source with a higher priority occurs during the DMA transfer, the FR family halts the DMA transfer and branches to the appropriate interrupt routine. This mechanism is valid as long as the interrupt request exists. However, the suppression mechanism does not work if the interrupt source is cleared, resulting that the DMA transfer resumes in the interrupt process routine. Therefore, use the DMA suppression function, if you do not want the DMA transfer to be resumed in the process routine of the interrupt source with a level which halts the DMA transfer. The DMA suppression function is activated when other value than "0000_B" is written to the DMAH[3:0] bit in the DMA overall control register, and stopped when "0000_B" is written.
- This function is mainly used in the interrupt process routine. Before the interrupt source is cleared in the interrupt process routine, the content of the DMA suppression register is incremented by 1. This prevents the subsequent DMA transfer from being performed. After the interrupt process is addressed, the content of the DMAH[3:0] bit is decremented by 1 before the return. If there are multiple interrupts, the content of the DMAH[3:0] bit is not "0000_B" yet, meaning that the DMA transfer continues to be suppressed. If there are not multiple interrupts, the content of the DMAH[3:0] bit is "0", meaning that the DMA transfer request is immediately enabled.

● Notes

- As number of the register's bits are 4 bits, this function cannot be used for multiple interrupts with more than 15 levels.
- The priority order of the DMA tasks should be placed at least 15-level higher than other interrupt levels.

■ Start of Operation

Starting of the DMA transfer is controlled independently for each channel, but you must enable all the channels' operations before that.

● All the channels' operation enabled

Before activating each of the DMAC channels, you must enable all the channels' operation using the DMA operation enable bit (DMACR:DMAE) in advance. If not enabled, any activation settings or occurred transfer requests are invalid.

● Transfer activation

Use the operation enable bit in the control register of each channel to activate the transfer operation. When the transfer request is accepted for the activated channel, the DMA transfer operation is started in the specified mode.

● Activation from pausing state

If each or all the channels control is paused before the activation, the state remains after activating the transfer operation. If a transfer request is issued at this time, the request is accepted and maintained. The transfer is started at the point where the pausing is deactivated.

■ Acceptance and Transfer of Transfer Request

- When activated, the transfer request specified for each channel starts to be sampled.
- If the peripheral interrupt activation is selected, the DMAC transfers are continued until the transfer request is cleared, and if cleared, the transfer is stopped at a transfer unit (peripheral interrupt activation).
  Since the peripheral interrupt level is treated as detected, the interrupt should be performed with the interrupt cleared by DMA.
- The transfer request is always accepted, even when other channel's request is accepted to perform the transfer. The channel to transfer for each transfer unit is determined based on a judgment of the priority.

■ Peripheral Interrupt Clear by DMA

- This DMA has a function to clear the peripheral interrupt. This function works when the peripheral interrupt is selected for the DMA activation source (when IS[4:0]=$1XXXX_B$).
- The peripheral interrupt is cleared only for the specified activation source. That is, only the peripheral function specified in IS[4:0] is cleared.

● Timing of occurrence for interrupt clear

- The timing of occurrence depends on the transfer mode (see the section "16.5 Operation Flowcharts").

[Block/step transfer]

If the block transfer is selected, a clear signal occurs for each block (step) transfer.

[Burst transfer]

If the burst transfer is selected, the clear signal occurs when the specified number of transfers are completed.

■ Pause

The DMA transfer pauses if:

- Setting of pausing by writing to the control register (Each channel independently or all the channels simultaneously)

  If pause is set by the pause bit, the corresponding channel's transfer is stopped until the pause deactivation is enabled again. Pause should be checked in the DSS bit.

  If the pause is deactivated, the transfer is resumed.

- Hold suppression level interrupt process

  If an interrupt request with a higher level than the hold suppression level occurs, all the channels being transferred are paused at the boundary of the transfer unit, and the NMI/interrupt is processed first after releasing the bus right. The transfer request accepted during the processing of the interrupt is maintained.

  The channel in which the request is retained resumes the transfer after the interrupt process is completed.

■ Termination/Stop of Operation

Termination of the DMA transfer is controlled independently for each channel, but you can disable all the channels' operations.

- Transfer termination

  If the reload operation is invalid, the transfer is stopped when the transfer count register turns to "0". The subsequent transfer requests become invalid after "normal termination" is displayed in the end code (the DMACA:DENB bit is cleared).

  If the reload operation is valid, the initial value is reloaded when the transfer count register turns to "0". A transfer request is being waited again after "normal termination" is displayed in the end code (the DMACA:DENB bit is not cleared).

- All the channels' operation disabled

  If the DMA operation enable bit DMAE is used to disable all the channels' operations, all the DMAC operations are stopped, including those of running channels. If all the channels' DMA operations are enabled again after that, a transfer is not performed unless it is re-activated for each channel. In this case, interrupt does not occur at all.

■ Stop By Error

As some cases where the transfer is stopped due to other source than the normal termination by completion of the specified number of transfers, there are stop and halt by various errors.

● Occurrence of Transfer Stop Request from Peripheral Circuit

Some peripheral circuits that output the transfer request can generate the transfer stop request when an abnormality (such as receive/send error around the communication system) is found.

If this transfer stop request is received, DMAC stops the corresponding channel's transfer, displaying "transfer stop request" in the end code.

See the explanation of the transfer source selection bits bit28 to bit24 (IS4 to IS0) of the DMACA register for information on whether there is a transfer stop request in the peripheral circuit.

See the specification of each peripheral circuit for details on the condition for generation of each transfer stop request.

■ DMAC Interrupt Control

The following interrupts can be output for each DMAC channel, independently of the peripheral interrupt, which becomes a transfer request.

- Transfer end interrupt . . . . Occurs only for normal termination.
- Error interrupt . . . . . Transfer stop request from the peripheral circuit (Error due to peripheral)

These interrupts are all output according to the content of the end code.

The interrupt request can be cleared by writing "000_B" to DSS2 to DSS0 (end code) of DMACS.

The end code must be cleared by writing "000_B" for reactivation.

If the reload operation is valid, it is automatically reactivated, but the end code is not cleared in this case, meaning that the data remains until a new end code is written when the next transfer is finished.

There is only one type of the end source that can be displayed in the end code, so if the multiple sources occur simultaneously, the result of the priority judgment is displayed. Interrupt that occurs in this case follows the displayed end code.

The priority of the end code is shown in the following (from the top with the highest priority):

- Reset
- Clear by writing "000_B"
- Peripheral stop request
- Normal termination
- Channel selection and control

■ DMA Transfer during the Sleep

- DMAC can operate during the sleep mode.
- If the operation during the sleep mode is expected, you should consider the following:
  (1) As CPU is being stopped, the DMAC register cannot be changed. The setting should be finished before entering the sleep mode.
  (2) The sleep mode is deactivated by interrupt, and if interrupt around the DMA activation source is selected, the interrupt controller must be used to disable the interrupt.

Similarly, if you do not want to deactivate the sleep mode in the DMA end interrupt, the interrupt should be disabled.

■ Channel Selection and Control

You can set up to five channels simultaneously for the transfer channel. Each of the channel can be set independently for each function.

● Priority among channels

Only one channel is simultaneously available for the DMA transfer, and the priority is set among the channels.

The priority setting has two modes, fix/rotation, and is selected for each channel group (described later).

- Fix mode

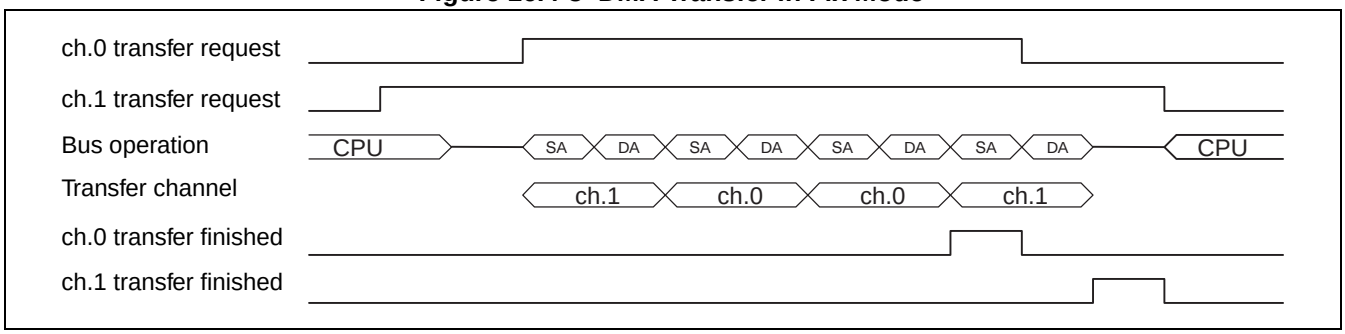
  Fixed from the smallest channel number.

  (ch.0 > ch.1 > ch.2 > ch.3 > ch.4)

  If a transfer request with a higher priority is accepted during the transfer, the transfer channel is switched to the channel with the higher priority at the end of one transfer unit (number specified in the block size specification register × data width).

  After the higher priority transfers are finished, the original channel's transfer is resumed.

Figure 16.4-3 DMA Transfer in Fix Mode

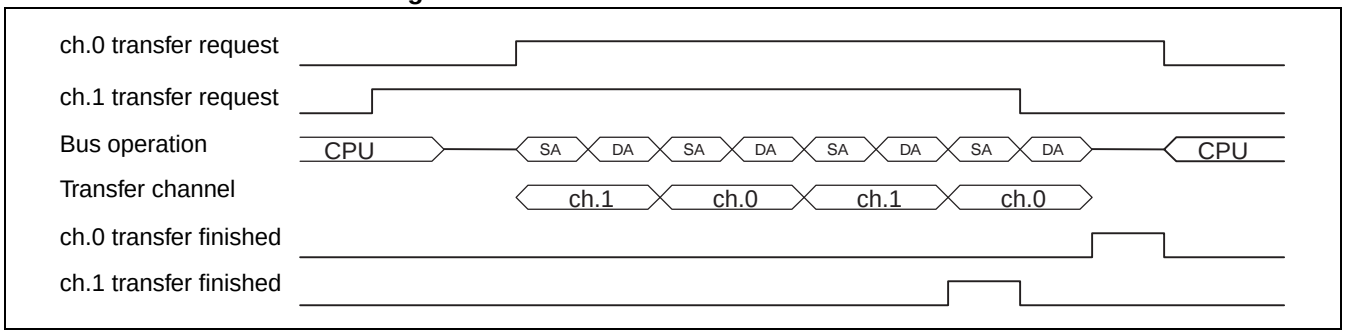


- Rotation Mode (Only between ch.0 and ch.1)

  The initial state is set the same as (1) after the operation is enabled, but the priority of the channel changes for each transfer completion. Therefore, if the transfer request is output simultaneously, the channels switches for each transfer unit.

  This is a mode that is useful if the continuous/burst transfer is set.

Figure 16.4-4 DMA Transfer in Rotation Mode



● Channel group

The priority is selected in the following unit.

Mode	Priority level	Remarks
Fix	ch.0 > ch.1	-
Rotation	ch.0 > ch.1 ↑↓ ch.0 < ch.1	The initial state is the top row. It is reversed when the transfer occurs for the top row.

MB91313A Series

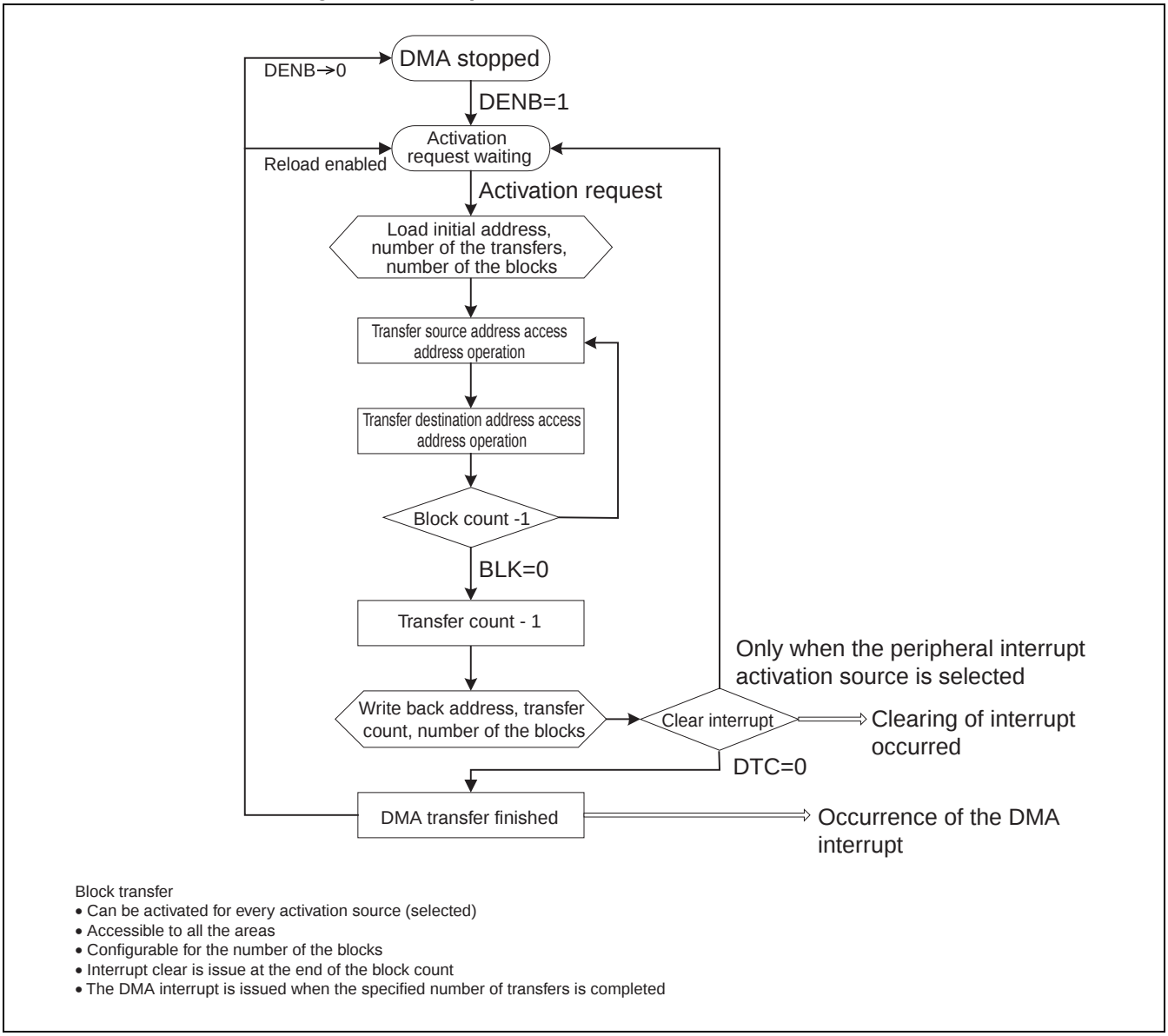
16.5 Operation Flowcharts

This section contains operation flowcharts for the following operation modes:

- **Block transfer**
- **Burst transfer**

■ Block Transfer

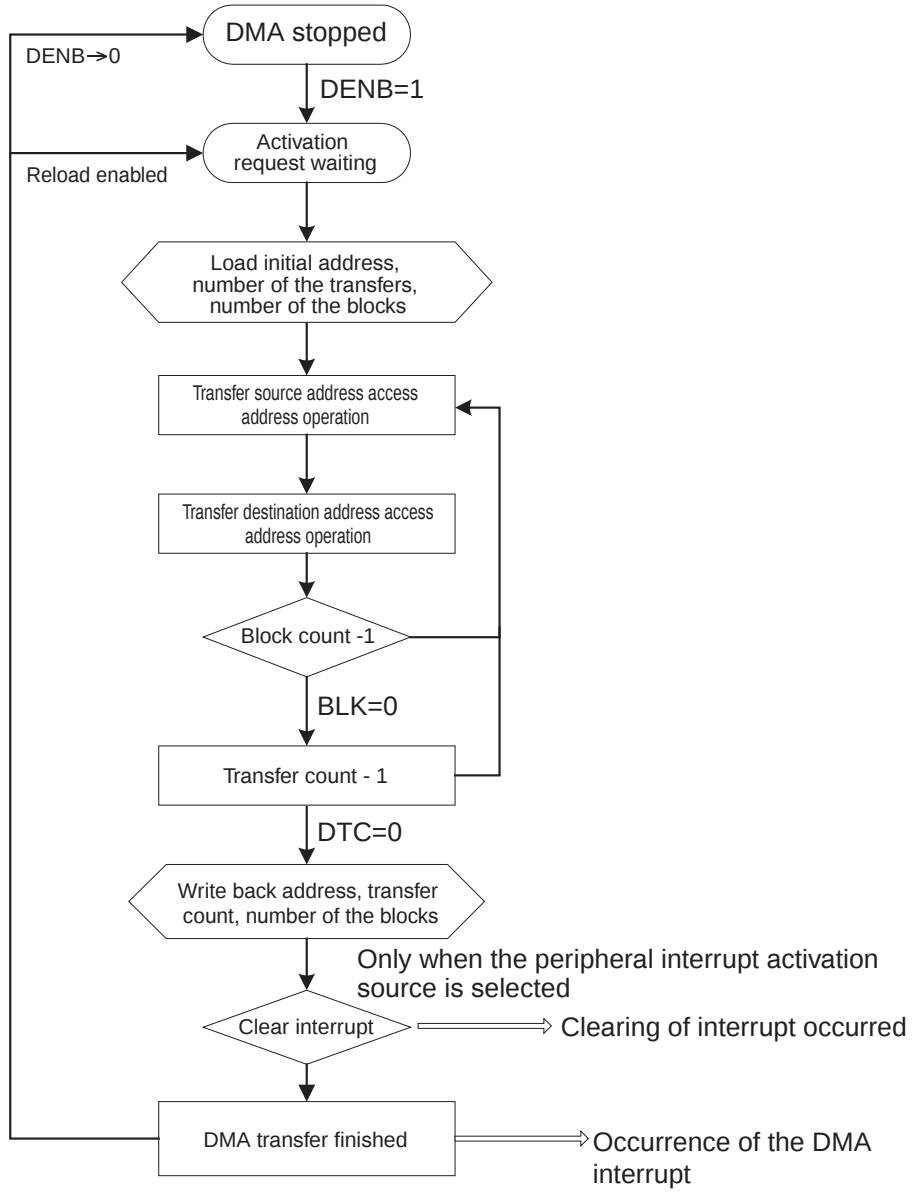
Figure 16.5-1 Operation Flowchart of Block Transfer



Block transfer
- Can be activated for every activation source (selected)
- Accessible to all the areas
- Configurable for the number of the blocks
- Interrupt clear is issue at the end of the block count
- The DMA interrupt is issued when the specified number of transfers is completed

■ Burst Transfer

Figure 16.5-2 Operation Flowchart of Burst Transfer



Burst transfer
- Can be activated for every activation source (selected)
- Accessible to all the areas
- Configurable for the number of the blocks
- Interrupt clear and the DMA interrupt are issued when the specified number of transfer is completed.

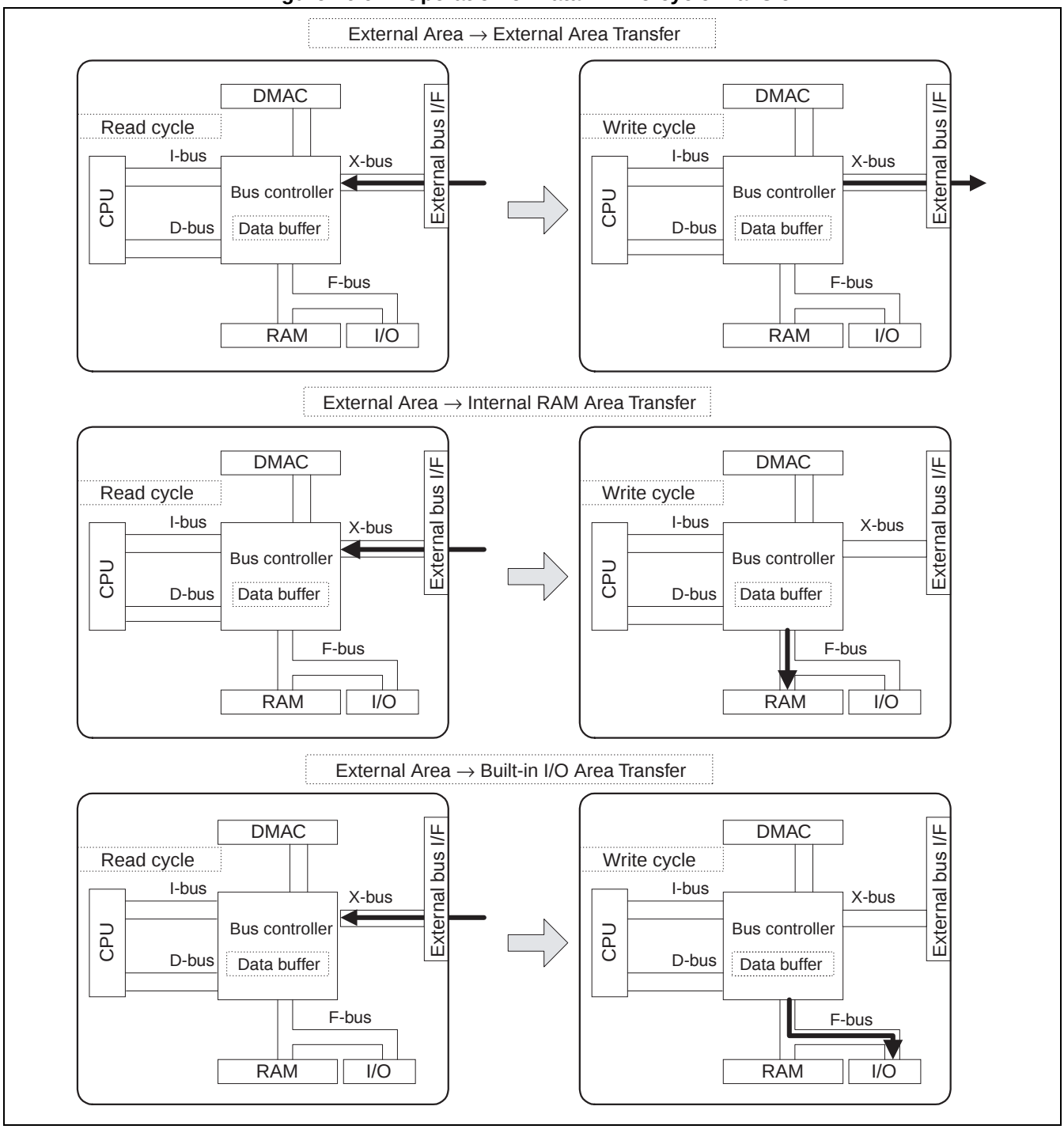
MB91313A Series

16.6 Data Bus

This section shows transfer examples of data bus.

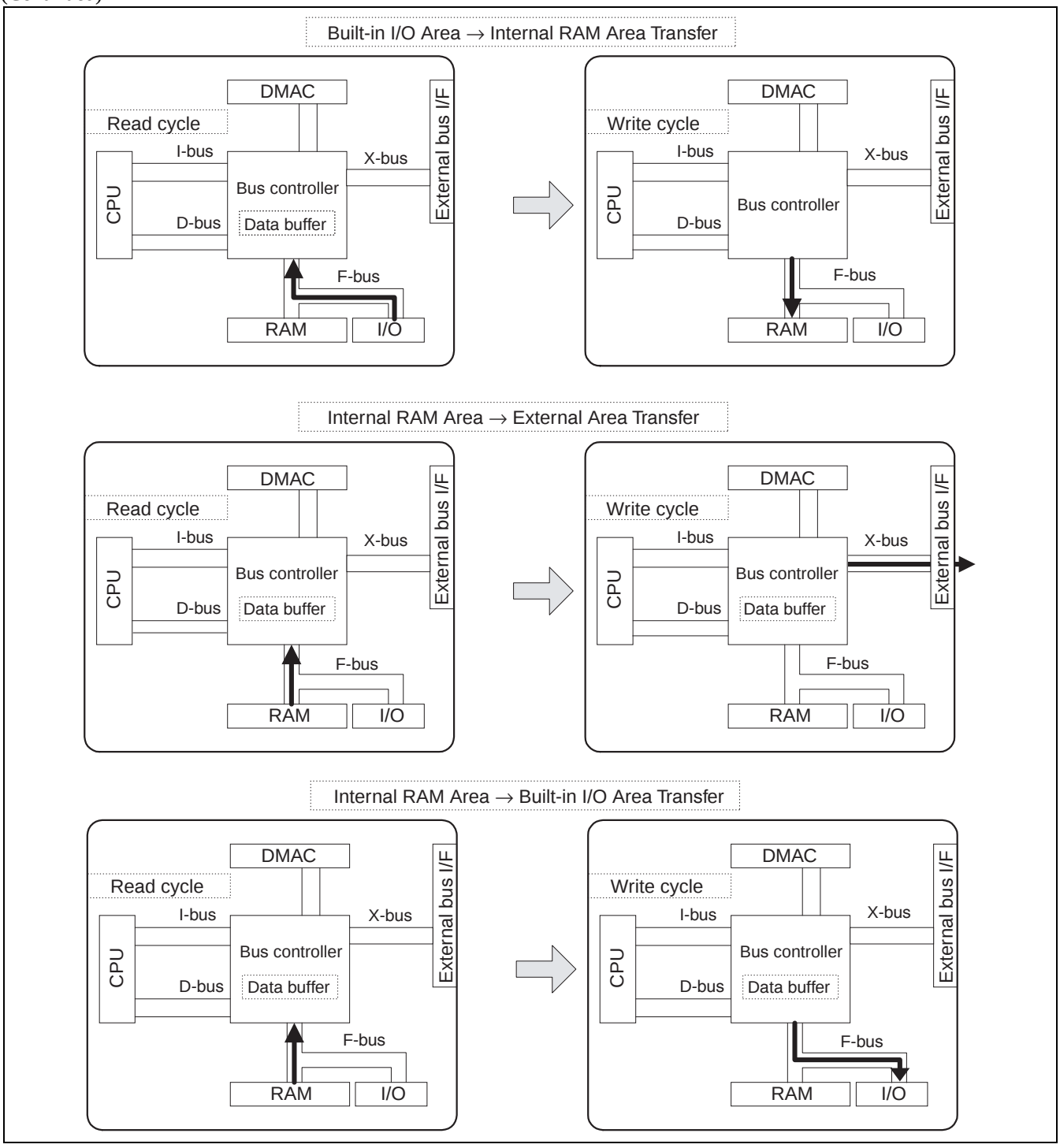
■ Operation of Data in Two-cycle Transfer

Figure 16.6-1 Operation of Data in Two-cycle Transfer



(Continued)

(Continued)


Built-in I/O Area → Internal RAM Area Transfer
DMAC
Read cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
Data buffer
F-bus
RAM
I/O
DMAC
Write cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
F-bus
RAM
I/O
Internal RAM Area → External Area Transfer
DMAC
Read cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
Data buffer
F-bus
RAM
I/O
DMAC
Write cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
Data buffer
F-bus
RAM
I/O
Internal RAM Area → Built-in I/O Area Transfer
DMAC
Read cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
Data buffer
F-bus
RAM
I/O
DMAC
Write cycle
I-bus
X-bus
External bus I/F
CPU
Bus controller
D-bus
Data buffer
F-bus
RAM
I/O

CHAPTER 17

REMOTE CONTROL RECEPTION

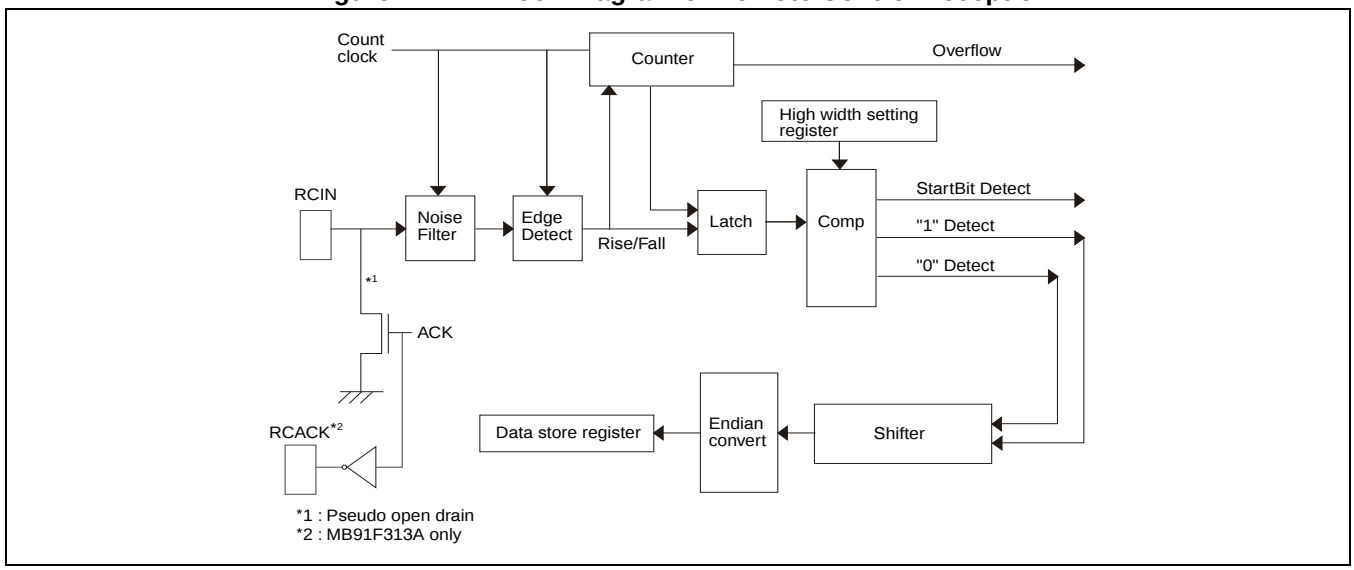
This chapter describes the functions and operations of HDMI-CEC reception, ACK automatic a response, and the remote control reception.

17.1 Overview of Remote Control Reception

This Product is equipped with HDMI-CEC reception, ACK automatic response, and remote-control reception function.

■ Block Diagram of Remote Control Reception

Figure 17.1-1 Block Diagram of Remote Control Reception



17.2 Register of Remote Control Reception

This section describes the configurations and functions of the registers used for remote control reception.

■ Register Configuration of Remote Control Reception

Table 17.2-1 shows a list of remote control reception registers.

Table 17.2-1 Registers of the Remote Control Reception

Address	Register Abbreviation	Register Name
$000X0_H$	RCCR	Remote control reception control register *1
$000X1_H$	RCST	Remote control reception interrupt control register *1
$000X2_H$	RCSHW	Start bit high width setting register *1
$000X3_H$	RCDAHW	High width setting register A *1
$000X4_H$	RCDBHW	High width setting register B *1
$000X5_H$	Reserved	
$000X6_H$	RCADR1	Device address setting register 1 *1
$000X7_H$	RCADR2	Device address setting register 2 *1
$000X8_H$	RCDTHH	Data store register HH
$000X9_H$	RCDTHL	Data store register HL
$000XA_H$	RCDTLH	Data store register LH
$000XB_H$	RCDTLL	Data store register LL
$000XC_H$	RCCKD	Clock divided register *2
$000XD_H$		

X = 18_H : (ch.0), 19_H : (ch.1)

*1 : 8-bit access only

*2 : 16-bit access only

17.2.1 Remote Control Reception Control Register (RCCR)

Shows the bit configuration of the remote control reception control register (RCCR).

■ Remote Control Reception Control Register (RCCR)

Figure 17.2-1 Bit Configuration of Remote Control Reception Control Register (RCCR)

bit	7	6	5	4	3	2	1	0
	THSEL	Reserved	Reserved	Reserved	ADRCE	MOD1	MOD0	EN
Attribute	R/W							
Initial value	0	-	-	-	0	0	0	0

R/W: Readable/writable

Note:

This register allows only 8-bit access.

[bit 7]: THSEL

This is the threshold select bit.

Initial value is "0".

Set the criteria to check for "0" and "1" in the high width setting register A/B.

Status	THSEL	
	0	1
W > width A W < width B	"0" data	"1" data
W > width A W ≥ width B	"1" data	"0" data

[bit 6 to bit 4]: Reserved

Write	Ignored.
Read	The read value is "0".

[bit 3]: ADRCE

Enable address comparison bit.

The initial value is "0" (disable comparison). Comparison of receive address and device address is enabled when set to "1".

When comparison is enabled, an ACK/OVF interrupt occurs only when the addresses match.

In CEC mode, an ACK is returned when the address match is detected. In the case of broadcast address, match is assumed, but no ACK is returned.

Set "0" if the mode is other than SIRCS mode or HDMI-CEC mode.

[bit 2,bit 1]: MOD1,MOD0

These bits are used to set the remote control reception operation mode.

MOD1	MOD0	Function
0	0	SIRCS mode [Initial Value]
0	1	Forbidden to set
1	0	NEC/AEHA mode (repeat signal not supported)
1	1	HDMI-CEC mode

Except for SIRCS mode (MOD1=1), the input signal is internally reversed.

High width comparison is applied to Low width.

[bit 0]: EN

This is the operation enable bit.

Remote control reception operation starts when this bit is set to "1".

Initial value is "0" (stopped).

Do not change the following registers and bits when this bit is "1" (operating).

RCCR register's THSEL bit, ADRCE bit, and MOD bit

RCST register's OVFSEL

RCSHW, RCDAHW, RCDBHW, RCADR1, RCADR2, and RCCKD register

17.2.2 Remote Control Reception Interrupt Control Register (RCST)

Figure 17.2-2 shows the bit configuration of the remote control reception interrupt control register (RCST).

■ Remote Control Reception Interrupt Control Register (RCST)

Figure 17.2-2 Bit Configuration of Remote Control Reception Interrupt Control Register (RCST)

bit	7	6	5	4	3	2	1	0
	STIE	ACKIE	OVFIE	OVFSEL	ST	ACK	EOM	OVF
Attribute	R/W							
Initial value	0							

R/W: Readable/writable

Note:

This register allows only 8-bit access.

[bit 7]: STIE

This bit enables start bit interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 6]: ACKIE

This bit enables ACK interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

It is valid only in CEC mode.

[bit 5]: OVFIE

This bit enables counter overflow interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

This interrupt occurs only when there is an overflow after a start bit is detected.

No interrupt occurs if start bit is undetected.

[bit 4]: OVFSEL

This bit is used to set the overflow detection condition.

Value	Description
0	Overflow occurs when the counter reaches 128 clock counts.
1	Overflow occurs when the counter reaches 256 clock counts.

[bit 3]: ST

This bit indicates the detection of start bit.

Value	Description
0	Start bit undetected
1	Start bit detected

Cleared when "0" is written.

An interrupt occurs when a start bit is detected while the STIE bit is "1".

[bit 2]: ACK

This bit indicates the detection of ACK.

Value	Description
0	ACK undetected
1	ACK detected

Cleared when "0" is written.

An interrupt occurs when an ACK is detected while the ACKIE bit is "1".

When address comparison is enabled, an interrupt occurs only when the addresses match.

It is valid only in CEC mode.

[bit 1]: EOM

This bit indicates the detection of EOM.

Value	Description
0	EOM undetected
1	EOM detected

Cleared when "0" is written.
It is valid only in CEC mode.

[bit 0]: OVF

This bit indicates the detection of counter overflow.

Value	Description
0	Counter overflow undetected
1	Counter overflow detected

When address comparison is enabled, an interrupt occurs only when the addresses match.
Cleared when "0" is written.
In SIRCS mode, the OVF flag is not set until the second byte is received.

17.2.3 Device Address Configuration Registers 1, 2 (RCADR1, RCADR2)

Figure 17.2-3 shows the bit configuration of the device address configuration registers 1, 2 (RCADR1, RCADR2).

■ Device Address Configuration Registers 1, 2 (RCADR1, RCADR2)

Figure 17.2-3 Bit Configuration of Device Address Configuration Registers 1, 2 (RCADR1, RCADR2)

bit	7	6	5	4	3	2	1	0
	Reserved			RCADR1,2				
Attribute				R/W				
Initial value	-	-	-	0	0	0	0	0

R/W: Readable/writable

Note:

This register allows only 8-bit access.

[bit 7 to bit 5]: Reserved

Write	Ignored.
Read	The read value is "0".

[bit 4 to bit 0]: RCADR1, 2

This is the register to set the device side (receive side) address.
The address set in this register is compared with the Remote Control Reception device address and HDMI-CEC destination.
Do not set $0F_H$ (broadcast address) in this register in HDMI-CEC mode.

17.2.4 Start Bit High Width Configuration Register (RCSHW)

Figure 17.2-4 shows the bit configuration of the start bit high width configuration register (RCSHW).

■ Start Bit High Width Configuration Register (RCSHW)

Figure 17.2-4 Bit Configuration of Start Bit High Width Configuration Register (RCSHW)

bit	7						0
	RCSHW						
Attribute	R/W						
Initial value	0						

R/W: Readable/writable

Note:

This register allows only 8-bit access.

This register is used to set the High duration of the start bit.

A start bit is detected when the width of the received High exceeds the set value.

If the High width of the received signal is less than the set value, no start bit is detected and start bit High detection wait state is reentered.

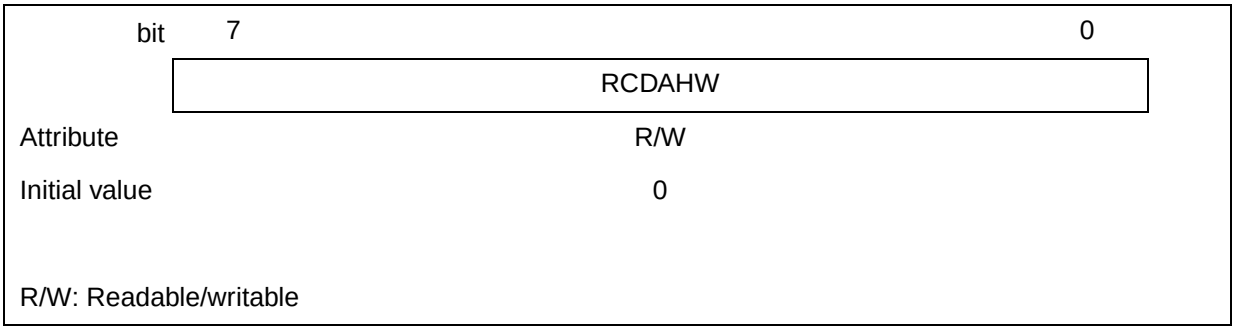
Set RCSHW $\leq$ 127 (value not exceeding overflow detection) if OVFSEL=0.

17.2.5 "H" Width Configuration Register A (RCDAHW)

Figure 17.2-5 shows the bit configuration of the "H" Width Configuration register A (RCDAHW).

■ "H" Width Configuration Register A (RCDAHW)

Figure 17.2-5 Bit Configuration of "H" Width Configuration A (RCDAHW)



Note:

This register allows only 8-bit access.

This is register A used to set the "H" duration.

The value set in this register must be such that $2 \leq$ RCDAHW < RCDBHW.

Also, in CEC mode, set so that RCDAHW < 46 (less than ACK response pulse width).

17.2.6 "H" Width Configuration Register B (RCDAHW)

Figure 17.2-6 shows the bit configuration of the "H" Width Configuration register B (RCDBHW).

■ "H" Width Configuration Register B (RCDAHW)

Figure 17.2-6 Bit Configuration of "H" Width Configuration Register B (RCDBHW)

bit	7						0
	RCDBHW						
Attribute	R/W						
Initial value	0						

R/W: Readable/writable

Note:

This register allows only 8-bit access.

This is register B used to set the "H" duration.

Do not set value less than RCCDAHW.

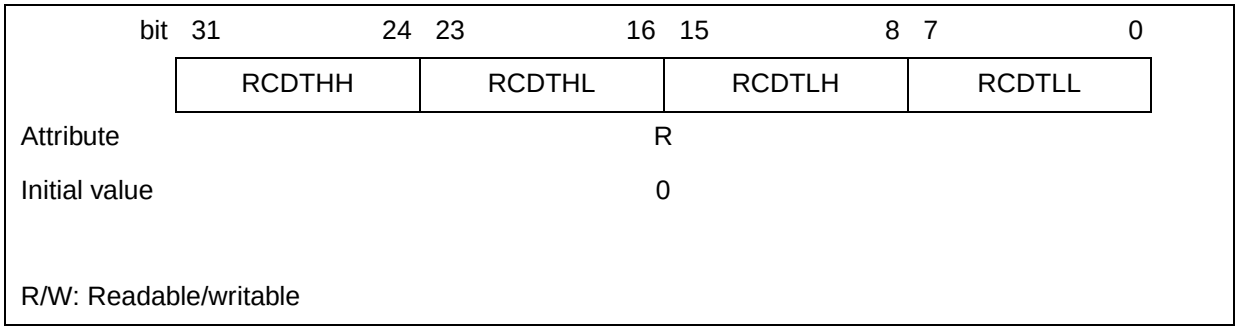
Make sure RCCDAHW < RCCDBHW < RCSHW is satisfied.

17.2.7 Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Figure 17.2-7 shows the bit configuration of the data storage registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL).

■ Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Figure 17.2-7 Bit Configuration of Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)



These are the registers to store the received data.

In CEC mode, the received data is stored in RCDTHH.

In remote control mode, data is stored starting from RCDTHH as each 8 bits are received.

When a counter overflow interrupt occurs, the bits received up to that point are stored MSB justified.

When the EN bit of RCCR register is "0", the value read from these registers is undefined.

If a signal exceeding 4 bytes is received, the excess data is ignored and not reflected in the registers.

17.2.8 Clock Division Configuration Register (RCCKD)

Figure 17.2-8 shows the bit configuration of the clock division configuration register (RCCKD).

■ Clock Division Configuration Register (RCCKD)

Figure 17.2-8 Bit Configuration of Clock Division Configuration Register (RCCKD)

bit	15 – 13	12	11 – 0
	Reserved	CKSEL	CKDIV
Attribute	R/W	R/W	R/W
Initial value	-	0	0

R/W: Readable/writable

Note:

This register allows only 16-bit access.

[bit 15 to bit 13]: Reserved

Write	Ignored.
Read	The read value is "0".

[bit 12]: CKSEL

This bit selects the operating clock.

Value	Description
0	Clock dividing the resource clock is selected.
1	Source oscillation clock (32kHz) is selected.

To select a sub-clock by CKSEL bit of Clock Division Configuration Register (RCCKD), write "1" in PLL2EN bit of Clock Source Control register (CLKP) to oscillate the sub-clock.

[bit 11 to bit 0]: CKDIV

These bits are used to set the resource clock division rate.
Division rate is CKDIV+1.
From 1 division (no division) to 4096 divisions can be set (only if CKSEL=0).

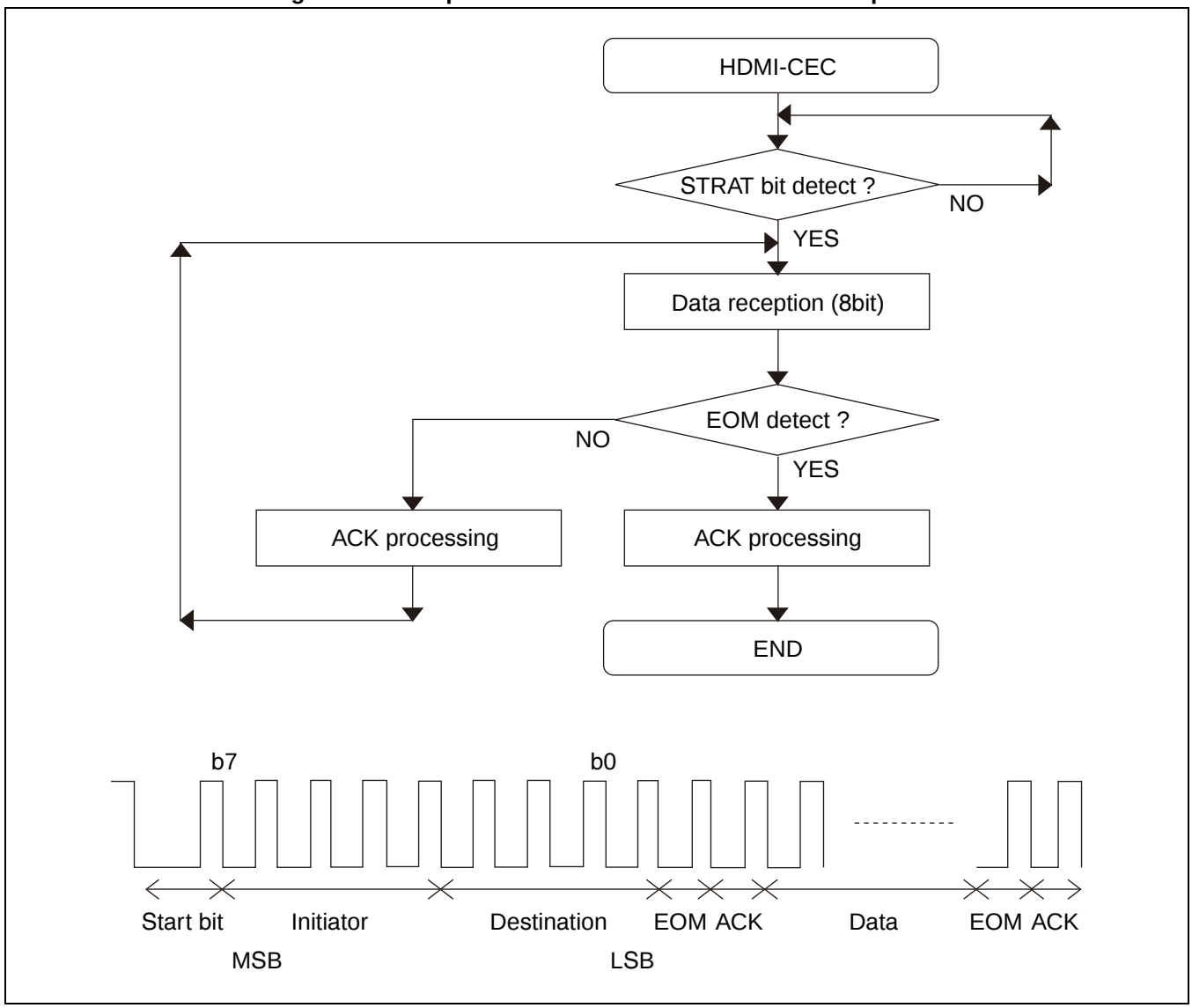
17.3 Explanation of Operations and Setting Procedure Examples

This section describes the remote control reception operation. A setting examples for each operation status are also shown.

■ Operation Flowchart of Remote Control Reception

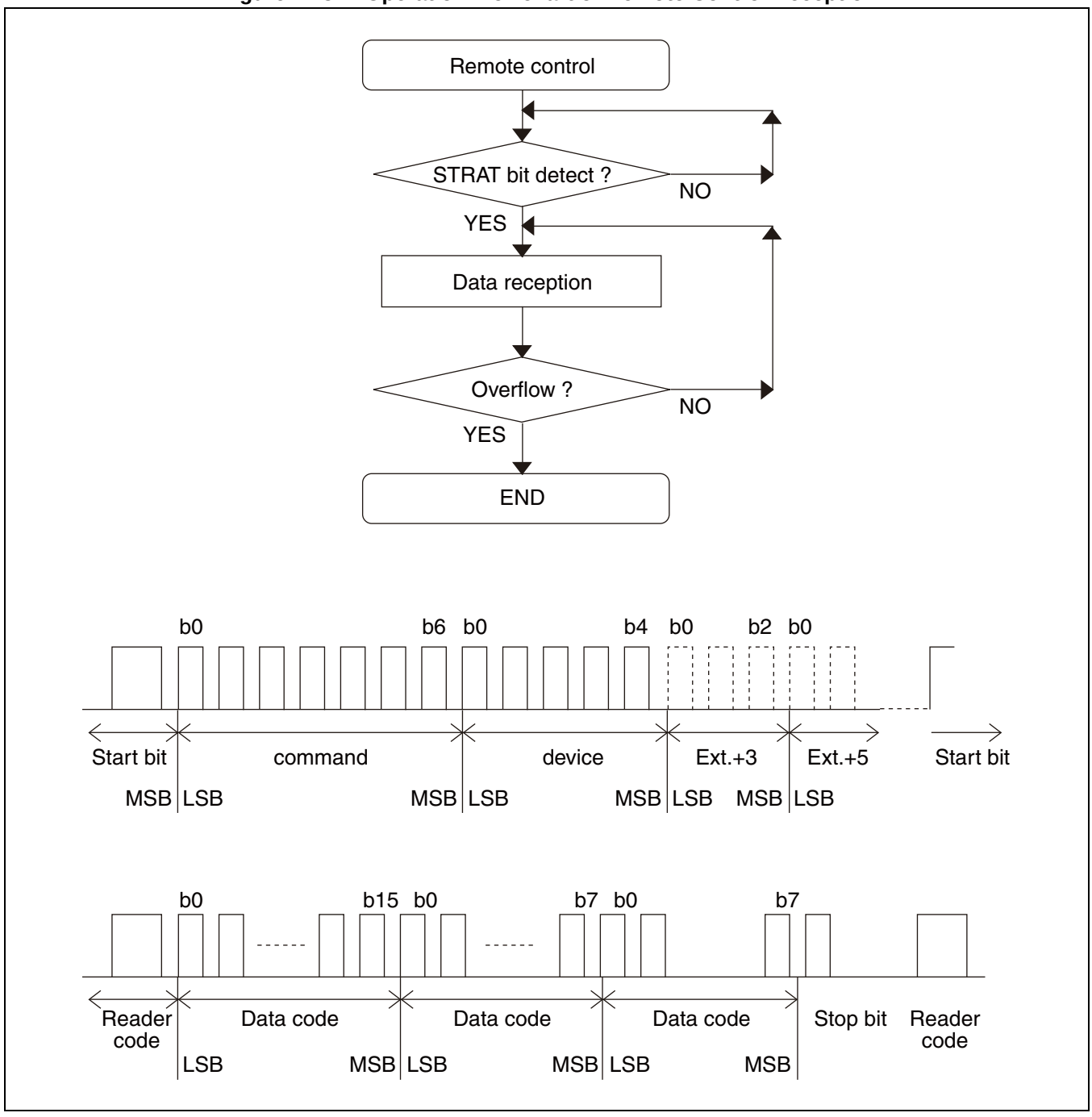
● HDMI-CEC Reception

Figure 17.3-1 Operation Flowchart of HDMI-CEC Reception



● Remote Control Reception

Figure 17.3-2 Operation Flowchart of Remote Control Reception



■ Setting Example of Remote Control Reception

Table 17.3-1 Setting Example of HDMI-CEC

Registers	Settings	Remarks
Remote Control Reception control register	MOD=11, THSEL=1, ADRCE=1	
Remote Control Reception interrupt control register	ACKIE=1, OVFSEL=1, OVFIE=1	(7.8 ms)
Start bit high width configuration register	114	3.5 ms
"H" width configuration register A	13	0.4 ms
"H" width configuration register B	42	1.3 ms

Table 17.3-2 Setting Example of Remote Control (SIRCS)

Registers	Settings	Remarks
Remote Control Reception control register	MOD=00, THSEL=0, ADRCE=1	
Remote Control Reception interrupt control register	ACKIE=0, OVFSEL=0, OVFIE=1	3.9 ms
Start bit high width configuration register	76	2.3 ms
"H" width configuration register A	17	0.52 ms
"H" width configuration register B	37	1.1 ms

Table 17.3-3 Setting Example of Remote Control (NEC)

Registers	Settings	Remarks
Remote Control Reception control register	MOD=10, THSEL=0	
Remote Control Reception interrupt control register	ACKIE=0, OVFSEL=1, OVFIE=1	7.8 ms
Start bit high width configuration register	144	4.4 ms
"H" width configuration register A	15	0.46 ms
"H" width configuration register B	52	1.6 ms

CHAPTER 18

FLASH MEMORY

This chapter provides an outline of flash memory and explains its register configuration, register functions, and operations.

Code: CM71-00512-2E

Page: 500, 500, 501, 501, 502, 505, 507, 511, 522, 525, 531

18.1 Outline of Flash Memory

MB91313A series has a built-in 544 KB flash memory.
The internal flash memory can be erased either by sector or in all of the sectors collectively, and can be written (programmed) in halfwords (16-bit units), via the FR-CPU.

■ Outline of Flash Memory

This flash memory is a built-in 544 KB memory operating at 3.3 V. Similar to the discrete flash memory, this flash memory supports writing outside the device using the ROM programmer. In addition to the functions equivalent to the discrete flash memory, when used as built-in ROM for the FR-CPU, the flash memory allows instructions and data to be read from in words (32 bits), enabling a high-speed operation of the device.

This product supports the following features by combining flash memory macros and FR-CPU interface circuits:

- Features for use as CPU memory, for storing programs and data

  Accessibility through 32-bit bus when used as ROM

  Allowing read, write, and erase (automatic algorithm*) by the CPU

- Features equivalent to single flash memory product

  Allowing read or write (automatic algorithm*) by a ROM writer

  *: Automatic algorithm: embedded algorithm

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

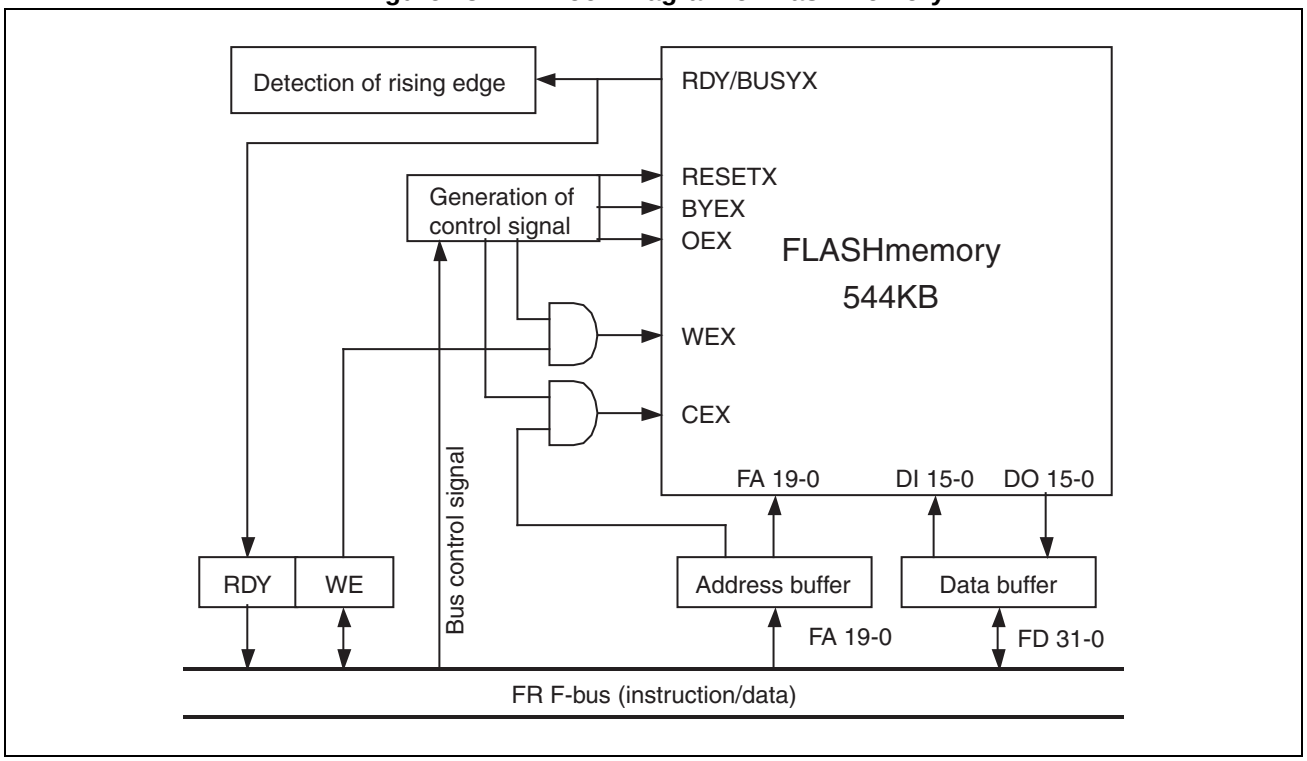
This section explains use of the flash memory accessed from the FR-CPU.

For information on using the flash memory accessed from a ROM writer, see the instruction manual provided with the ROM writer.

■ Block Diagram

Figure 18.1-1 shows a block diagram of flash memory.

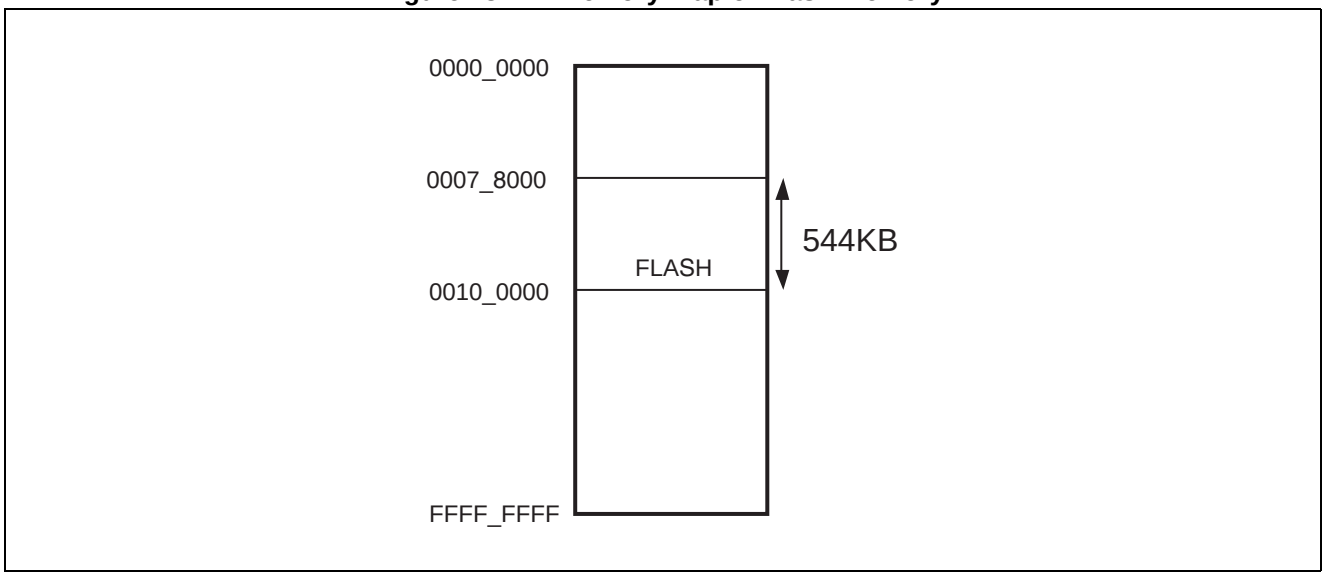
Figure 18.1-1 Block Diagram of Flash Memory



■ Memory Map

Figure 18.1-2 shows a memory map of flash memory.

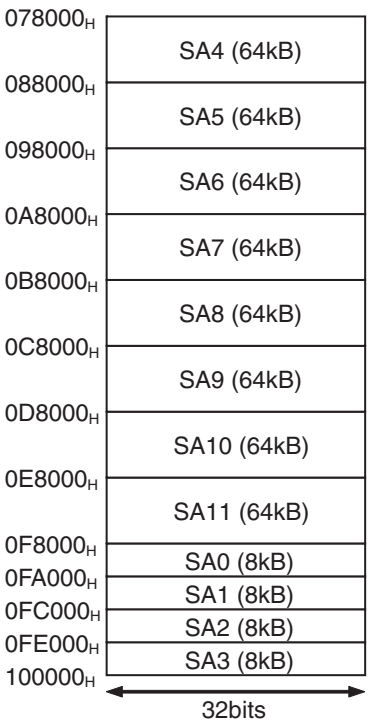
Figure 18.1-2 Memory map of Flash memory



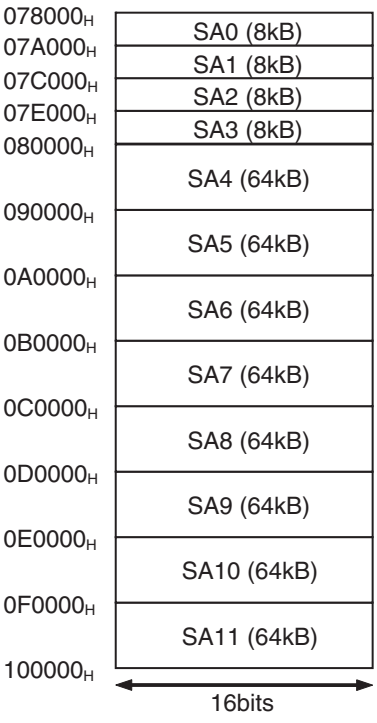
■ Sector Address Table

Sector configuration is different between Flash mode and CPU mode.

❍ Sector map in CPU mode



❍ **Sector map in FLASH mode**



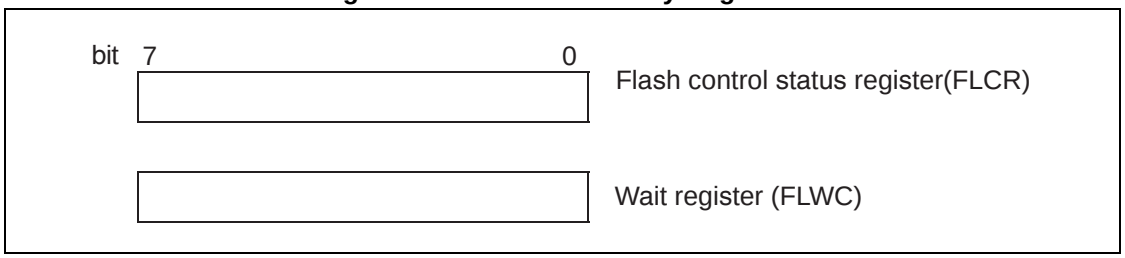
18.2 Flash Memory Registers

The flash memory has two registers: flash control status register (FLCR) and wait register (FLWC).

■ List of Flash Memory Registers

Figure 18.2-1 shows a list of flash memory registers.

Figure 18.2-1 Flash Memory Registers



18.2.1 Flash Control Status Register (FLCR)

The flash control status register (FLCR) indicates the operation status of flash memory.

■ Configuration of Flash Control Status Register (FLCR) (CPU Mode)

This register controls writing to flash memory.

This register can only be accessed in CPU mode. Do not access this register using the Read Modify Write instruction.

Figure 18.2-2 shows the bit configuration of FLCR.

Figure 18.2-2 Bit Configuration of FLCR

FLCR

Address	bit	7	6	5	4	3	2	1	0	Initial value
00007000_H		Reserved	Reserved	Reserved	Reserved	RDY	Reserved	WE	Reserved	$0000X000_B$
		R	R	R	R	R	R	R/W	R	

R/W: Readable/writable
R: Read only

[bit7 to bit4] Reserved: Reserved bits

These are reserved bits.

The read value is always "0000_B".

[bit3] RDY: Ready bit

This bit indicates the operation status of the automatic algorithm (data write/erase).

The read data indicates the flash memory status as listed in the table below.

0	Data writing or erasing is in process, flash memory is not ready to accept a new Data write/Erase command, and no data can be read from a flash memory address.
1	Flash memory is ready to accept a new Data write/Erase command and data can be read from a flash memory address.

- This bit is not initialized during a reset.
- Only read operation is possible, but write operation does not affect this bit.

[bit2] Reserved bit

This is a reserved bit. Always set this bit to "0".

[bit1] WE : Write Enable bit

This bit controls the writing of data and commands to flash memory in CPU mode.

In flash memory mode, writing is enabled regardless of this bit.

0	Writing to flash memory is disabled and data is read from flash memory in 32-bit access mode.
1	Writing to flash memory is enabled and data is read from flash memory in 16-bit access mode.

This bit is initialized to "0" during reset.

Notes:

- If this is overwritten, confirm that the RDY bit has stopped the automatic algorithm. When the RDY bit is set to "0", the value of this bit cannot be charged.
- When WE=1, do not respond to the instruction access request and only respond to the data access request.

[bit0] Reserved bit

This is a reserved bit. Always set this bit to "0".

18.2.2 Wait Register (FLWC)

The wait register (FLWC) controls the wait state for flash memory access in CPU mode.

■ Bit Configuration of Wait Register (FLWC)

The wait register (FLWC) consists of the following bits:

Figure 18.2-3 Bit Configuration of Wait Register (FLWC)

FLWC

Address	bit	7	6	5	4	3	2	1	0	Initial value
007004_H		FAC1	FAC0	WTW2	WTW1	WTW0	WTC2	WTC1	WTC0	01011011_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

[bit7, bit6] FAC1, FAC0: Access control bits

These bits are set to control internal pulse generation for flash memory control. The ATDIN/EQIN pulse width can be set by setting these bits.

Please set the value matched to setting (WTC2 to WTC0) of the wait cycle of the read access.

FAC1	FAC0	ATDIN width	EQIN width	Remarks
0	0	0.5 clock	1.0 clock	Read 1 wait
0	1	1.0 clock	1.5 clock	Read 2, 3 wait [Initial value]
1	0	1.5 clock	2.0 clock	Setting disabled
1	1	2.0 clock	2.5 clock	Setting disabled

[bit5 to bit3] WTW2 to WTW0: Write wait cycle bits

WTW2	WTW1	WTW0	Wait cycle	Remarks
0	0	0	-	Setting disabled
0	0	1	1	Setting disabled
0	1	0	2	Setting disabled
0	1	1	3	[Initial value]
1	0	0	4	Setting disabled
1	0	1	5	Setting disabled
1	1	0	6	Setting disabled
1	1	1	7	Setting disabled

- This bit is initialized to "011_B" at a reset.
- Be sure to set always"011_B".

[bit2 to bit0] WTC2, WTC1, WTC0: Wait cycle bits

WTC2	WTC1	WTC0	Wait cycle	Remarks
0	0	0	-	Setting disabled
0	0	1	1	-
0	1	0	2	-
0	1	1	3	Initial value
1	0	0	4	Setting disabled
1	0	1	5	Setting disabled
1	1	0	6	Setting disabled
1	1	1	7	Setting disabled

- These bits are initialized to "011_B" at a reset.

18.3 Flash Memory Access Modes

The following two types of access mode are available for the FR-CPU:

- **FR-CPU ROM mode:**
  One word (32 bits) data can be read but not written in a single cycle.
- **FR-CPU Programming mode:**
  Word (32 bits) access is prohibited but reading/writing data with half-word (16 bits) is enabled.

■ FR-CPU ROM Mode (32 Bits, Read Only)

In this mode, the flash memory serves as FR-CPU internal ROM. This mode enables to read one word (32 bits) in one cycle but does not enable to write to flash memory or to start the automatic algorithm.

❍ Mode specification

When specifying this mode, set the WE bit of the flash control status register (FLCR) to "0".

This mode is always set after releasing a reset at CPU run time.

This mode can be set only when the CPU is running.

❍ Detailed operation

In this mode, one word (32 bits) can be read from the flash memory area in one cycle.

❍ Restrictions

- Address assignment and endians in this mode differ from those for writing with the ROM writer.
- In this mode, commands and data cannot be written to flash memory together.

■ FR-CPU Programming Mode (16 Bits, Read/Write)

This mode enables data to be written and erased. As one word (32 bits) cannot be accessed in one cycle, program execution in flash memory is disabled in this mode.

❍ Mode specification

When specifying this mode, set the WE bit of the flash control status register (FLCR) to "1".

When a reset is released at CPU run time, the WE bit indicates "0". To set to this mode, set the WE bit to "1". If the WE bit is set again to "0" through a writing operation or because of a reset, the device enters FR-CPU ROM mode.

When the RDY bit of the flash control status register (FLCR) is "0", the WE bit cannot be overwritten. When overwriting the WE bit, ensure that the RDY bit is set to "1".

❍ **Detailed operation**

Half word (16 bits) data can be read from the flash memory area in one cycle.

The automatic algorithm can be started by writing a command to flash memory.

When the automatic algorithm starts, data can be written to or erased from flash memory. For details on the automatic algorithm, see "18.4 Automatic Algorithm of Flash Memory".

❍ **Restrictions**

- Address assignment and endians in this mode differ from those for writing with the ROM writer.
- This mode inhibits reading data in words (32 bits).

■ Automatic Algorithm Execution Status

When the automatic algorithm is started in FR-CPU programming mode, the operation status of the automatic algorithm can be checked by reading the RDY bit in the flash control status register (FLCR).

When the RDY bit is set to "0", data is being written or erased with the automatic algorithm, and no write or erase command can be accepted. Moreover, data cannot be read from any address in flash memory.

Data read with the RDY bit set to "0" is a hardware sequence flag to indicate flash memory status.

18.4 Automatic Algorithm of Flash Memory

The flash memory automatic algorithm can be started using a Reset, Write, Chip Erase, or Sector Erase command. The Sector Erase command can suspend and resume the automatic algorithm.

■ Command Sequence

To start the automatic algorithm, write one to six half-words (16 bits) data in sequence. This is called the command.

If the address and data to be written are invalid or are written in an incorrect sequence, the flash memory is reset to read mode.

Table 18.4-1 lists commands that can be used to write data to or erase data from flash memory.

Table 18.4-1 Command Sequence Table

Command Sequence	Bus Write Cycle	1st Bus Write cycle		2nd Bus Write cycle		3rd Bus Write cycle		4th Bus Read/write cycle		5th Bus Write cycle		6th Bus Write cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	$XXXX_H$	$F0F0_H$	---	---	---	---	---	---	---	---	---	---
Reset	3	$DAAA8_H$	$AAAA_H$	$D5554_H$	5555_H	$DAAA8_H$	$F0F0_H$	---	---	---	---	---	---
Data writing	4	$DAAA8_H$	$AAAA_H$	$D5554_H$	5555_H	$DAAA8_H$	$A0A0_H$	PA	PD	---	---	---	---
Chip erase	6	$DAAA8_H$	$AAAA_H$	$D5554_H$	5555_H	$DAAA8_H$	8080_H	$DAAA8_H$	$AAAA_H$	$D5554_H$	$AAAA_H$	$DAAA8_H$	1010_H
Sector erase	6	$DAAA8_H$	$AAAA_H$	$D5554_H$	5555_H	$DAAA8_H$	8080_H	$DAAA8_H$	$AAAA_H$	$D5554_H$	$AAAA_H$	SA	3030_H
Sector erase suspend		Suspend erase during sector erase with input of Address= $XXXX_H$, data = $B0B0_H$											
Sector erase resume		Resume erase after Sector erase suspend, with input of Address= $XXXX_H$, data = 3030_H											
Continuous mode	3	$DAAA8_H$	$AAAA_H$	$D5554_H$	5555_H	$DAAA8_H$	2020_H	---	---	---	---	---	---
Continuous writing	2	$XXXXX_H$	$A0A0_H$	PA	PD	---	---	---	---	---	---	---	---
Continuous mode Reset	2	$XXXXX_H$	9090_H	$XXXXX_H$	$F0F0_H$ or 0000_H	---	---	---	---	---	---	---	---

RA: Read address
PA: Write address
SA: Sector address (Specify any address in the sector)
RD: Read data
PD: Write data

❍ Reset command

Set flash memory into Read/Reset mode.

The flash memory remains in reading state until another command is entered.

When the power is turned on, flash memory is automatically set to the read or reset state. In this case, data can be read without a reset command of the automatic algorithm.

To return to read/reset mode after the time limit is exceeded, issue a Reset command. Data is read from flash memory in the next read cycle.

❍ **Program (Data write)**

In FR-CPU programming mode, data is basically written in half-word units. The data write operation is performed in four cycles of bus operation. Writing to memory starts when the last write cycle is performed.

After executing automatic algorithm of data writing, it becomes unnecessary to control the flash memory externally. The flash memory itself internally generates write pulses to check the margin of the cells to which data is written. The data polling function compares bit7 of the original data with data polling flag (DQ7) of the written data, and if these bits are the same, the automatic write operation ends (see "■Hardware Sequence Flag" in "18.5 Execution Status of the Automatic Algorithm"). The automatic write operation then returns to the read/reset mode and accepts no more write addresses. After that, the flash memory requests the next valid address. In this manner, the data polling function indicates a write operation in memory.

During a write operation, all commands written to the flash memory are ignored. If a hardware reset starts during write operation, the data at the address for writing may become invalid.

Writing operations can be performed in any address sequence and outside of sector boundaries. However, write operations cannot change a data item "0" to "1". If a "0" is overwritten with a "1", the data polling algorithm either determines that the elements are defective, or that "1" has been written. In the latter case, however, the respective data item is read as "0" in reset or read/reset mode. A data item "0" can be changed to "1" only after an erase operation.

❍ **Chip Erase**

The Chip Erase command sequence ("erase all sectors simultaneously") is executed in six access cycles.

During the Chip Erase command sequence, the user does not have to write to flash memory before the erase operation. When the automatic algorithm of chip erasure is executed, flash memory checks cell states by writing a pattern of zeros before automatically erasing the contents of all cells (preprogram). In this operation, flash memory does not have to be controlled externally.

The automatic erase operation starts with the write operation of the command sequence and ends when data polling flag (DQ7) is set to "1", where flash memory returns to the read/reset mode. The chip erase time can be expressed as follows: time for sector erase × number of all sectors + time for writing to the chip (preprogram).

❍ Sector Erase

The Sector Erase command sequence is executed in six access cycles. The Sector Erase command is entered in the sixth cycle for starting the sector erase operation. The next Sector Erase command can be accepted within a time-out period of 40 µs to 160 µs after the last Sector Erase command is written.

Multiple Sector Erase commands can be accepted during the six bus cycles of the writing operation. During the command sequence, Sector Erase commands (3030_H) for sectors whose contents are to be erased simultaneously are written consecutively to the addresses for these sectors. The sector erase operation itself starts from the end of the time-out period of 40 µs to 160 µs after the last Sector Erase command is written. When the contents of multiple sectors are erased simultaneously, the subsequent Sector Erase commands must be input within the 40 µs time-out period to ensure that they are accepted. For checking whether the succeeding Sector Erase command is valid, read sector erase timer flag (DQ3) (see "■Hardware Sequence Flag" in "18.5 Execution Status of the Automatic Algorithm").

During the time-out period, any command other than Sector Erase and Temporarily Stop Erase is reset at read time, and the preceding command sequence is ignored. In the case of the Temporary Stop Erase command, the contends of the sector are erased again and the erase operation is completed.

Any combination and number (from 0 to 6) of sector addresses can be entered in the sector erase buffers.

The user does not have to write to flash memory before the sector erase operation.

Flash memory automatically writes to all cells in a sector whose data is automatically erased (preprogram). When the contents of a sector are erased, the other cells remain intact. In these operations, flash memory does not have to be controlled externally.

The automatic sector erase operation starts from the end of the 40 µs to 160 µs time-out period after the last Sector Erase command is written. When data polling flag (DQ7) is set to "1", the automatic sector erase operation ends and flash memory returns to the read/reset mode. At this time, other commands are ignored.

The data polling function is enabled for any sector address in which data has been erased. The time required for erasing the data of multiple sectors can be expressed as follows: time for sector erase + time for sector write (preprogram) × number of erased sectors.

❍ Temporarily Stop Erase

The Temporarily Stop Erase command temporarily stops the automatic algorithm in flash memory when the user is erasing the data of a sector, thereby making it possible to write data to and read data from the other sectors. This command is valid only during the sector erase operation and ignored during chip erase and write operations.

The Temporarily Stop Erase command ($B0B0_H$) is effective only during sector erasure operation that includes the sector erase time-out period after a Sector Erase command (3030_H) is issued.

When this command is entered within the time-out period, waiting for time-out ends and the erase operation is suspended. The erase operation is restarted when a Restart Erase command was written. Temporarily Stop Erase and Restart Erase commands can be entered with any address.

When a Temporarily Stop Erase command is entered during sector erase operation, the flash memory needs a maximum of 20 µs to stop the erase operation. When flash memory enters temporary erase stop mode, a RDY bit in FLCR register and data polling flag (DQ7) output "1", and toggle bit flag (DQ6) stops to toggle. For checking whether the erase operation has stopped, enter the address of the sector whose data is being erased and read the values of toggle bit flag (DQ6) and data polling flag (DQ7). At this time, another Temporarily Stop Erase command entry is ignored.

When the erase operation stops, flash memory enters the temporary erase stop and read mode. Data reading is enabled in this mode for sectors that are not subject to temporary erase. Other than that, there is no difference from the standard read operation.

After the temporary erase stop and read mode is entered, the user can write to flash memory by writing a Write command sequence. The write mode in this case is the temporary erase stop and write mode. In this mode, data write operations become valid for sectors that are not subject to temporary erase stop. Other than that, there is no difference from the standard byte writing operation. The temporary toggle bit flag (DQ6) can be used to detect this operation.

Note that toggle bit flag (DQ6) can be read from any address, but data polling flag (DQ7) must be read from write addresses.

To restart the sector erase operation, a Restart Erase command (3030_H) must be entered. Another Restart Erase command entry is ignored in this case. On the other hand, a Temporarily Stop Erase command can be entered after flash memory restarts the erase operation.

18.5 Execution Status of the Automatic Algorithm

Data write/erasure operation of flash memory is performed by automatic algorithm. The operation status of automatic algorithm can be checked by hardware sequence flag and RDY bit.

■ RDY bit

The flash memory uses RDY bit in flash memory status register (FLCR) in addition to the hardware sequence flag to indicate whether the automatic algorithm is running or ends.

When the value of the RDY bit is "0", the flash memory is executing a data write or erase operation, where new Data Write and Erase commands are not accepted.

When the value of the RDY bit is "1", the flash memory is in read/data write or erase operation wait state.

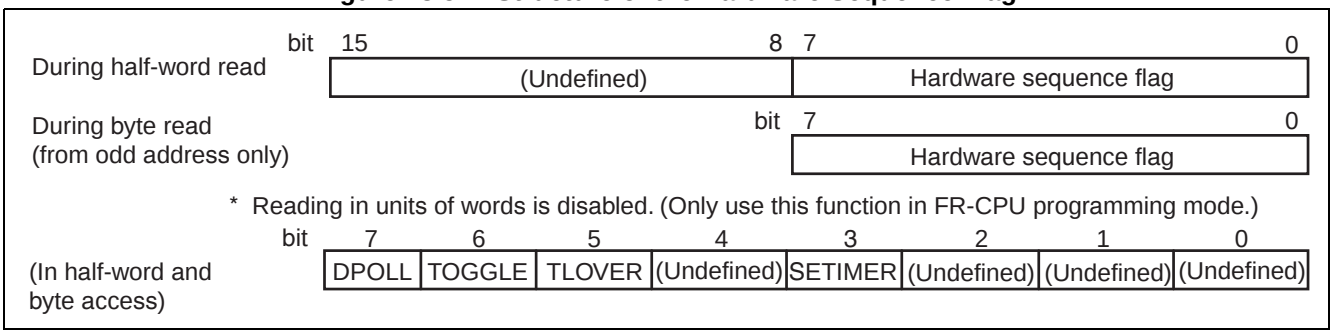
■ Hardware Sequence Flag

For obtaining the hardware sequence flag as data, read an arbitrary address (an odd address in byte access) from flash memory when the automatic algorithm is executed. The data contains five validity bits which indicate the status of the automatic algorithm.

When the automatic algorithm is executed for ROM1, specify an address in ROM1. When executed for ROM2, specify an address in ROM2.

Figure 18.5-1 shows the structure of the hardware sequence flag.

Figure 18.5-1 Structure of the Hardware Sequence Flag



The hardware sequence flag becomes invalid in FR-CPU ROM mode. Always use FR-CPU programming mode and read only in half-words or bytes.

Table 18.5-1 lists the possible statuses of the hardware sequence flag.

Table 18.5-1 Statuses of the Hardware Sequence Flag

Status			DPOLL	TOGGLE	TLOVER	SETIMR
Executing	Data write		Reverse data	Toggle	0	0
	Chip erase		0	Toggle	0	1
	Sector erase	Time-out duration	1	Toggle	0	0
		Erase duration	0	Toggle	0	1
	Temporary erase stop mode	Read (from sectors in temporary erase stop)	1	1	0	0
		Read (from sectors not in temporary erase stop)	Data	Data	Data	Data
		Data write (to sectors not in temporary erase stop)	Reverse data	Toggle*	0	0
Time limit exceeded	Data write		Reverse data	Toggle	1	0
	Chip/sector erase		0	Toggle	1	1
	Write operation during temporary erase stop		0	Toggle	1	1

*: TOGGLE toggles continuous read operations from any address.

The hardware sequence flags are explained below.

[bit7] DPOLL: Data polling flag (DQ7)

This flag is used with the data polling function to report that the automatic algorithm is being executed or terminated.

- Automatic data write operation status

  When read access is performed while the automatic algorithm of data write is being executed, flash memory outputs the inversion of bit7 of the last data written regardless of the address indicated by the address signal. When read access is performed at the end of the automatic write algorithm, flash memory outputs bit7 of the read data to the address indicated by the address signal.

- Chip erase operation status

  When read access is performed while the automatic algorithm of chip erase is being executed, flash memory outputs "0", regardless of the address indicated by the address signal. Similarly, flash memory outputs "1" at the end of the chip erase.

- During a sector erase operation:

  When read access is performed from the sector being erased during execution of the sector erase automatic algorithm, it outputs "0". Due to restrictions on the function in this series, the flash memory outputs "1" for 40 μs to 160 μs after the sector erase command is issued, and then outputs "0".

  After the sector erase is terminated, the flash memory outputs "1".

  For restrictions on the data polling flag (DQ7) and how to avoid erroneous judgment of sector erase completion, see section "18.7 Restrictions on Data Polling Flag (DQ7) and How to Avoid Erroneous Judgment".

- Temporary sector erase stop status

  When read access is performed during temporary sector erase stop status, flash memory outputs "1" when the address indicated by the address signal is included in the sector in erase status. If the address is not included in the sector in erase status, flash memory outputs bit7 of the read value to the address. For checking whether a sector is in temporary sector erase stop status and which sector is in erase status, read this bit and the toggle bit flag.

Note:

Read access to a specified address is ignored while the automatic algorithm is active. Values can be output to other bits after data polling flag operation terminates in data read operation. Therefore, when data is to be read after terminating the automatic algorithm, confirm that data polling is terminated in the current read access.

[bit6] TOGGLE: Toggle bit flag (DQ6)

This flag is used with the toggle bit function to mainly report that the automatic algorithm is being executed or terminated as well as data polling flag.

- Write or chip/sector erase operation status

  When continuous read operations are performed while the automatic algorithm of data write or chip/sector erase is being executed, flash memory outputs "1" and "0" toggle results to bit6 regardless of the address indicated by the address signal. When continuous read operations are performed at the end of the automatic algorithm of write or chip/sector erase algorithm, flash memory stops toggle operation of bit6 and outputs bit6 (DATA: 6) of the data read from the address indicated by the address signal.

  If a write target sector is protected from overwriting during a write operation, the toggle bit tries to toggle for about 2 μs and stops toggling without changing data. If all selected sectors are protected from overwriting, the toggle bit tries to toggle for about 100 μs and the system returns to read/reset status without changing data.

- Temporary sector erase stop status

  When a read operation is performed during a temporary sector erase stop operation, flash memory outputs "1" if the address indicated by the address signal is included in the sector in erase state. If the address is not included in the sector in erase state, flash memory outputs the data of bit6 of the read value at the address indicated by the address signal.

[bit5] TLOVER: Timing limit over flag (DQ5)

This flag is used to report that a time (number of internal pulses) specified internally with flash memory is exceeded while the automatic algorithm is being executed.

- Write or chip/sector erase operation status

  When read access is performed within a specified time (necessary for write or erase) after activating the automatic algorithm of write or chip/sector erase, flash memory outputs "0". If read access is performed beyond the specified time, flash memory outputs "1". Because these output operations are not affected by whether the automatic algorithm is being executed or terminated, these operations can be used to check whether data write or erase operation is successful. If flash memory outputs "1" while the automatic algorithm is being executed with the data polling function or toggle bit function, consider the write operation to be unsuccessful.

  For example, when "1" is written to a flash memory address where "0" is written, failure occurs. Flash memory is locked and the automatic algorithm is not terminated. Thus, valid data is not output from the data polling flag. The toggle bit flag does not stop toggling, the time limit is exceeded, and "1" is output to the TLOVER flag. This status indicates that flash memory was not used correctly, not that it was defective. Execute a Reset command.

[bit3] SETIMR: Sector erase timer flag (DQ3)

This flag is used to report that sector erasure is being awaited after starting a Sector Erase command.

- Sector erase operation status

  When read access is performed within a sector erase time-out period after starting a Sector Erase command, flash memory outputs "0" regardless of the address indicated by the address signal of the target sector. If read access is performed beyond the time-out period, flash memory outputs "1" regardless of the address.

  When "1" is set in this flag while the data polling or toggle bit function indicates that the automatic algorithm is being executed, an internally controlled erase operation has started. The writing of subsequent sector erase code and commands other than Erase Temporary Stop is ignored until erase operation terminates.

  When this flag is "0", flash memory accepts another sector erase code entry. In this case, it is recommended to check the status of this flag by software before writing the succeeding sector erase code. If this flag is "1" at the second time of status check, the additional sector erase code may not be accepted.

- Sector erase operation status

  When a read operation is performed during a temporary sector erase stop operation, flash memory outputs "1" if the address indicated by the address signal is included in the sector that is subject to the erase operation. If the address is not included in the sector that is subject to the erase operation, flash memory outputs the data of bit3 of the read value at the address indicated by the address signal.

18.6 Data Writing to and Erasing from Flash Memory

This section explains how to issue a command to start the automatic algorithm for following operations in flash memory.

• Reset • Data write • Chip erase • Sector erase • Temporary sector erase stop • Sector erase restart

■ Data Writing/Erase

Automatic algorithm (data write, chip erase, sector erase, temporary sector erase stop, sector erase restart) of flash memory is activated by writing the command sequence into bus. The write cycles for each bus must always be executed continuously. Termination of the automatic algorithm can be checked with the data polling function and toggle bit function. Flash memory is set again into read/reset status after the automatic algorithm terminates normally.

18.6.1 Read/Reset Status

This section explains the procedure to set flash memory into read/reset status by issuing reset command.

■ Read/Reset Status

To set the flash memory read/reset status, issue the reset command sequence table continuously to target sector in the flash memory.

A bus operation is performed one or three times with a Reset command sequence. There is no essential difference between these two sequences. Read/reset status is the initial status of flash memory, and flash memory is set in this status at power-on or when a command terminates normally. In this status, the system waits for a command other than Read/Reset to be entered.

Data can be read using normal read access in this status. Programs can be accessed from the CPU the same way the programs in mask ROM are accessed. The Read/Reset command is not necessary for reading data in normal read access. This command is required, however, to initialize the automatic algorithm if a command does not terminate normally.

18.6.2 Data Writing

This section explains the procedure to write data to flash memory by issuing data write command.

■ Data Writing

To activate the automatic algorithm of data write, issue the data write command in command sequence table continuously to target sector in flash memory. The automatic algorithm and automatic writing start when writing data to the target address terminates in the fourth cycle.

❍ How to specify address

Only even-numbered addresses can be specified in write data cycles. If an odd-numbered address is specified, data cannot be written correctly. In other words, data must be written to even-numbered addresses in units of half-words.

Data can be written by freely specifying the order of addresses where data is to be written. Moreover, data can be written beyond sector boundaries. Note that items of data can only be written with each write command in units of half-words.

❍ Notes on writing data

Data "0" cannot be changed to "1" in a write operation. If data "1" is overwritten, the data polling algorithm or toggle operation does not terminate, and the flash memory device is considered defective. An error is assumed with the time limit over flag if the specified write time is exceeded, or if only data "1" is apparently written, although data "0" is read in read/reset status. Data "0" can be changed to "1" only with an erase operation. All commands are ignored during automatic writing. If a hardware reset is activated during writing, the data being written is not guaranteed.

❍ Write procedure

Figure 18.6-1 shows an example of the write procedure.

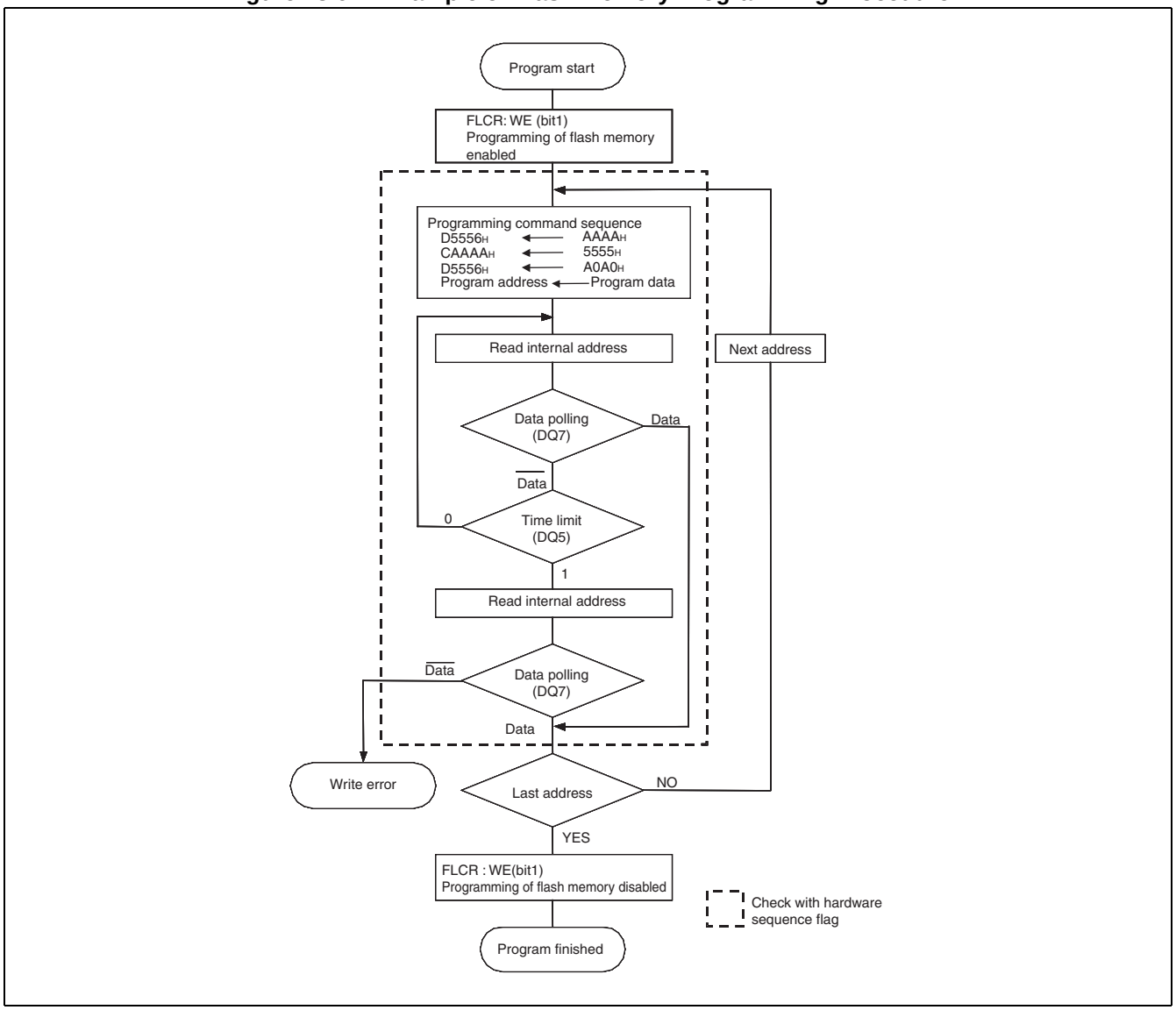
The status of the automatic algorithm in flash memory can be checked using the hardware sequence flag. In the example in Figure 18.6-1, the data polling flag (DQ7) is used to check for termination of the write operation.

Data for the flag check is read from the address where the last data was written.

The data polling flag (DQ7) changes together with the timing limit over flag (DQ5). Therefore, data polling flag (DQ7) must be rechecked even though timing limit over flag (DQ5) is set to "1".

The toggle bit flag (DQ6) also stops toggling simultaneously when the value of timing limit over flag (DQ5) is changed to "1". Therefore, this flag must also be rechecked.

Figure 18.6-1 Example of Flash Memory Programming Procedure



18.6.3 Erasing Data (Chip Erase)

This section explains the procedures to erase all items of data in flash memory by issuing chip erase commands.

■ Erasing Data (Chip Erase)

To erase all data in flash memory, issue the chip erase command in command sequence table to the target section in flash memory.

Six bus operations are necessary to execute a chip erase operation. The operation starts when the sixth write cycle is completed. The user need not write any value to flash memory before chip erase operation. Flash memory automatically writes "0" to erase all cells.

18.6.4 Erasing Data (Sector Erase)

This section explains the procedures to erase specified sectors in flash memory by issuing sector erase commands. Erasure in sector units is possible and two or more sectors can be specified with this command.

■ Sector Erase

To erase the sectors, issue the sector erase command continuously to the target sector in the flash memory.

❍ How to specify sectors

A sector erase operation can be performed with six bus operations. A 40µs to 160µs sector erase time-out period starts when a sector erase code (3030_H) is written to an even-numbered address accessible in the target sector in the sixth cycle. To erase another sector, a sector code (3030_H) must be written in the same cycle the same way.

❍ Note on specifying two or more sectors

A sector erase operation starts when the 40 µs to 160 µs sector erase time-out period terminates after the final sector erase code is written. Therefore, when two or more sectors are to be specified, the address and erase code of each target sector must be entered within 40 µs (in the sixth cycle of the command sequence) after specifying the preceding sector. Note that an address and erase code not entered within 40 µs may not be accepted. The sector erase timer flag (DQ3) of hardware sequence flag can be used to check the validity of a written sector erase code. The address at which the read sector erase time is written should indicate the target sector.

❍ Sector erase procedure

The hardware sequence flag can be used to check the status of the automatic algorithm in flash memory.

Figure 18.6-2 shows an example of the sector erase procedure. In this example, the toggle bit flag (DQ6) is used to check for termination of the erase operation.

Note that data for the flag check is read from the sector to be erased.

The toggle bit flag (DQ6) stops toggling simultaneously when the value of the timing limit over flag (DQ5) changes to "1". Therefore, the toggle bit flag (DQ6) must be rechecked even though the timing limit over flag (DQ5) is set to "1".

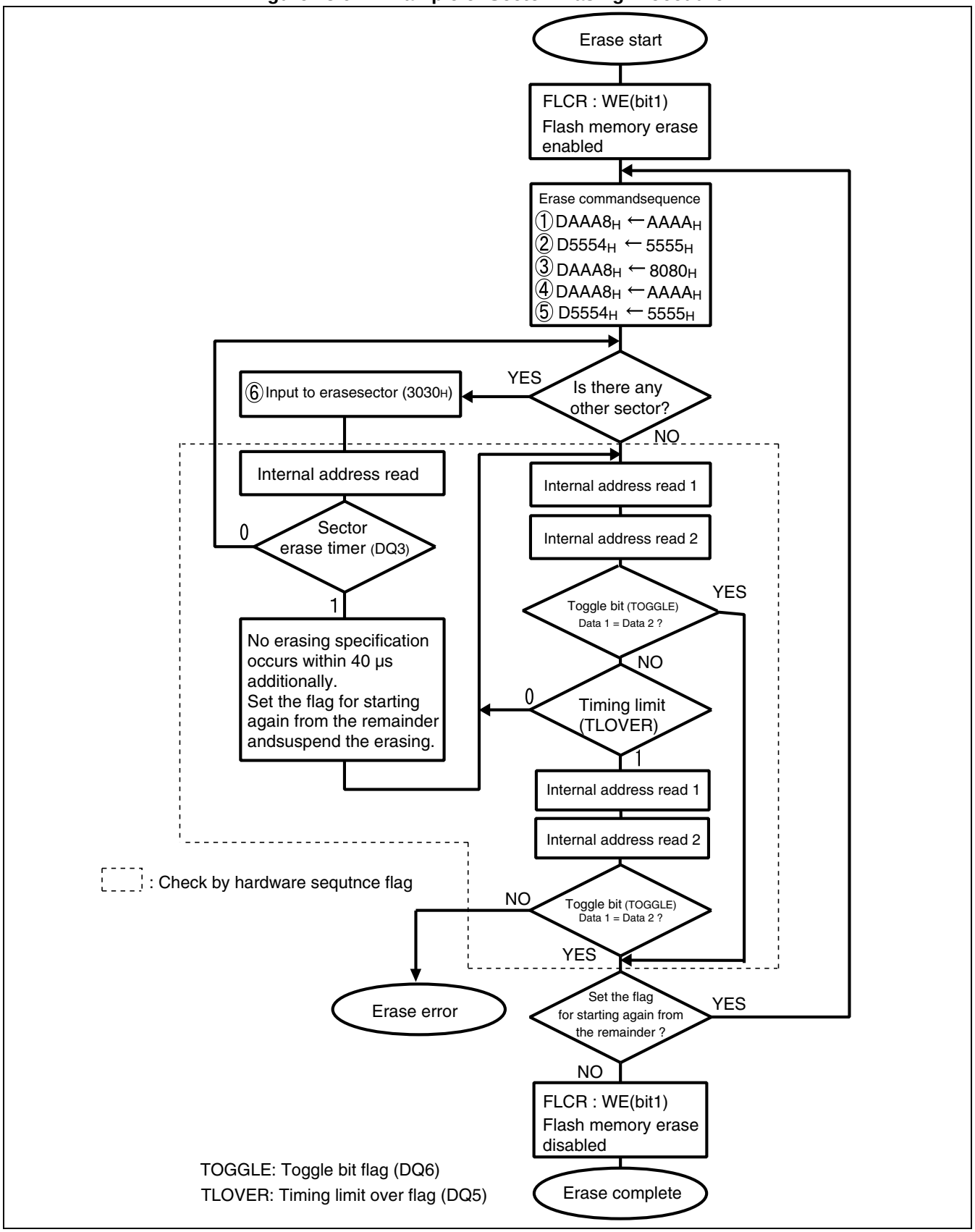
Because the data polling flag (DQ7) also changes with the timing limit over flag (DQ5), it must also be rechecked.

■ Restrictions on Data Polling Flag (DQ7)

Due to restrictions on the function in this series, the data polling flag(DQ7) outputs "1" for 40 µs to 160 µs after the sector erase command is issued, and then outputs "0". After the sector erase is terminated, the flash memory outputs "1".

For restrictions on the data polling flag (DQ7) and how to avoid erroneous judgment of sector erase completion, see section "18.7 Restrictions on Data Polling Flag (DQ7) and How to Avoid Erroneous Judgment".

Figure 18.6-2 Example of Sector Erasing Procedure



18.6.5 Temporary Sector Erase Stop

This section explains the procedures to stop the sector erase operation temporarily by issuing temporary sector erase stop command. When sector erase operation is temporarily stopped by this command, the data can be read from a sector not being erased by using this command.

■ Temporary Sector Erase Stop

To stop the sector erase temporarily, issue the temporary sector erase command in command sequence table to target sector in flash memory.

This command stops erase operation temporarily, so, data read from a sector not being erased becomes available. Data can only be read from the sector; data cannot be written there. This command is only effective during sector erasure that includes an erase time-out period. It is ignored during chip erase operation and write operation.

A sector erase operation is stopped temporarily by writing a temporary erase stop code ($B0B0_H$). The address where the temporary erase stop code is written should indicate an address in flash memory. A Temporary Sector Erase Stop command issued during temporary erase stop status is ignored.

If a Temporary Sector Erase Stop command is entered during a sector erase time-out period, the sector erase time-out is immediately canceled and erase operation in progress is stopped. If a Temporary Sector Erase Stop command is entered during a sector erase operation after the sector erase time-out period elapses, sector erase operation is stopped temporarily after up to 20 µs elapse.

18.6.6 Sector Erase Restart

This section explains how to restart the temporarily stopped sector erase operation by issuing the sector erase restart command.

■ Sector Erase Restart

To restart the sector erase which is temporarily stopped, issue the sector erase restart command in command sequence table to target sector in flash memory.

Restart operation starts when an erase restart code (3030_H) is written. The address where the erase restart code is written should indicate an address in flash memory.

Sector Erase Restart commands issued during a sector erase operation are ignored.

18.7 Restrictions on Data Polling Flag (DQ7) and How to Avoid Erroneous Judgment

This series has some restrictions on how to use the data polling flag (DQ7) during execution of the automatic sector erase algorithm. This section describes such restrictions and how to avoid erroneous judgment of sector erase completion.

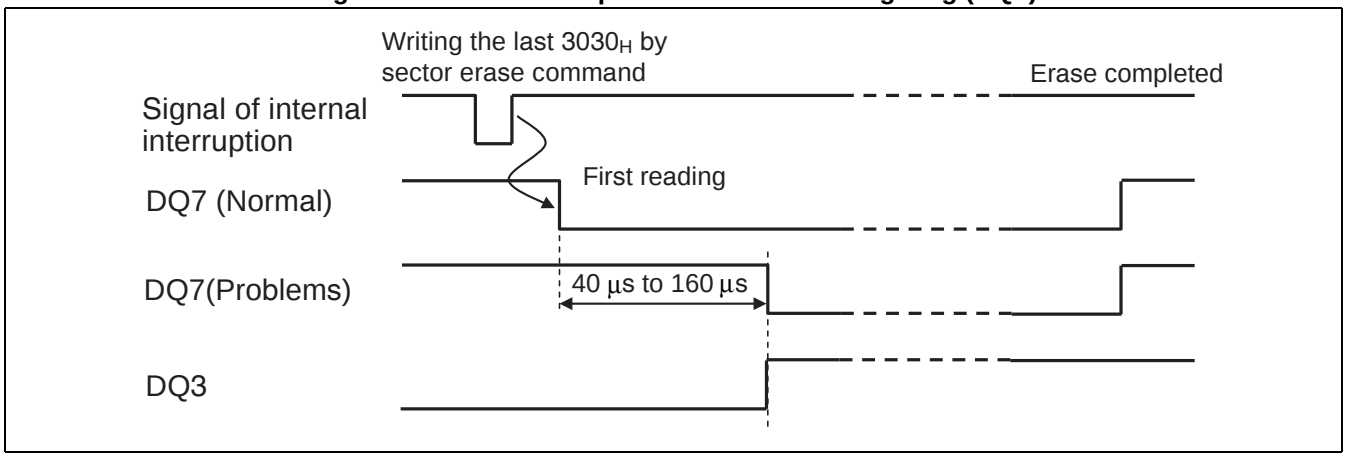
■ Description of Problems due to Restrictions

The data polling flag (DQ7) is used to indicate that the execution of the automatic algorithm is currently in progress or completed, by using the data polling function. In its original operation, as shown in Figure 18.7-1, DQ7 outputs "0" after the sector erase command is issued when the automatic algorithm is being started, and returns to "1" upon the completion of the erase operation. Therefore, the DQ7 polling algorithm indicates the completion of the erase operation by outputting "1".

In this series, DQ7 continues to output "1" for 40 μs to 160 μs, after the Sector Erase command is issued, and then it outputs "0". When the erase operation is completed, it then returns to "1". For this reason, if the sector erase polling is started while "1" is still being output immediately after the sector erase command is issued, the erroneous judgment that the erase operation has been completed may occur, although the erase operation has not actually started.

The timing for DQ7 to change from "1" to "0" after the sector erase command is accepted is the same as the timing for the sector erase timer flag (DQ3), which indicates the sector erase timeout period, to change from "0" to "1".

Figure 18.7-1 Actual Operation of Data Polling Flag (DQ7)



The following or other problems may occur, as a result of the erroneous judgment that the erase operation has been completed,

(1) Runaway or abnormal operation may occur, because the value of the sequence flag is read from the flash memory even when the CPU attempts to fetch instruction/data; therefore, the value of the program cannot be read properly.

(2) If the next command is issued after the erroneous judgment that the sector erase operation has been completed occurs, the first command may be cancelled, resulting in a return to the read state, or the next command may not be accepted.

■ How to Avoid Problems

Use one of the following methods to avoid the problems.

- Polling using the toggle bit flag (DQ6)

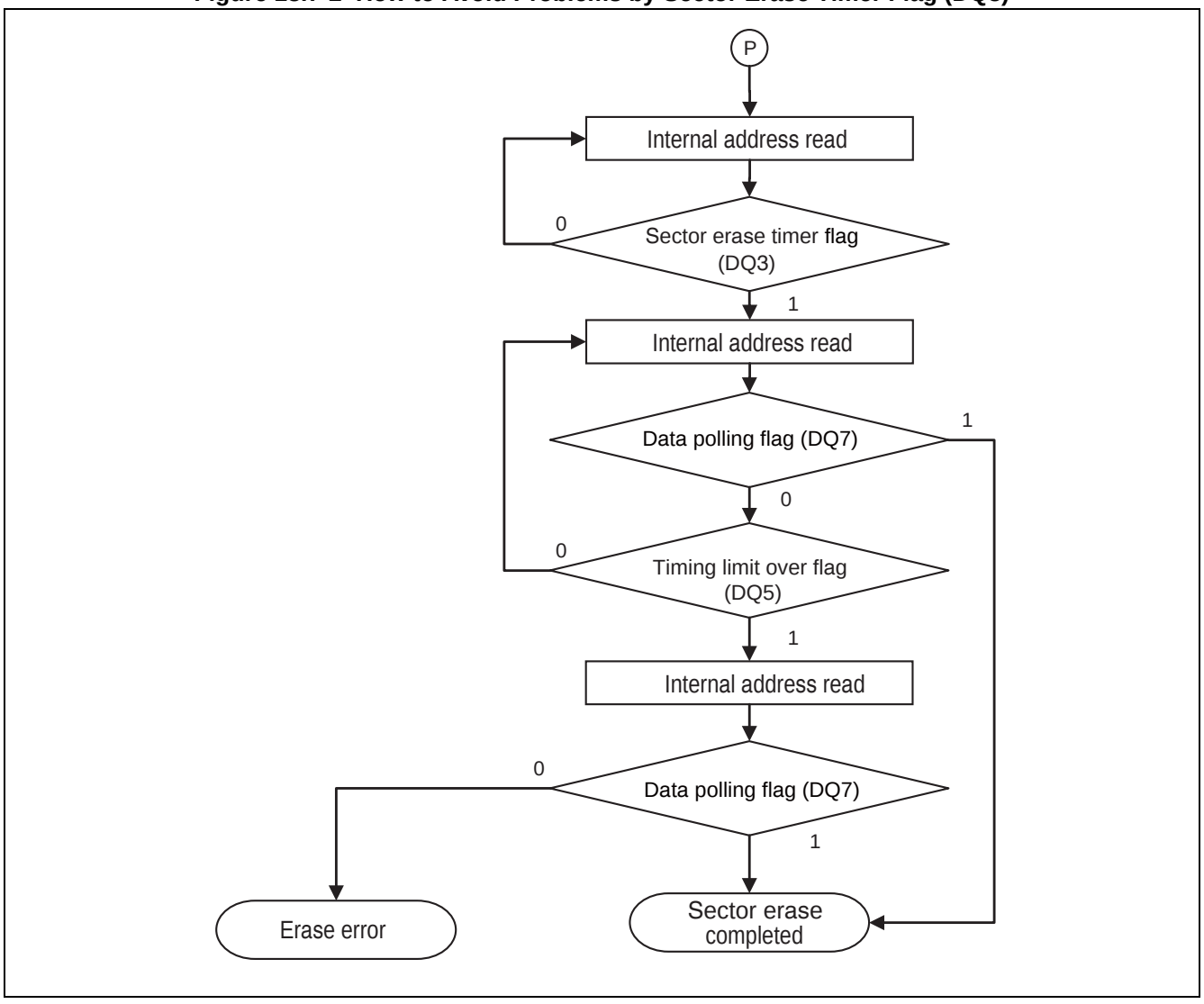
  Determine the state of the automatic algorithm using DQ6, as shown in Figure 18.6-2 in "18.6.4 Erasing Data (Sector Erase)"

  In the same manner as the data polling flag (DQ7), the toggle bit flag (DQ6) indicates that the automatic algorithm is being executed or has terminated by the toggle bit function.

- Starting polling of DQ7 after the sector erase timeout period elapses

  Before starting the polling of DQ7, wait for 160µs or more by software after the sector erase command is issued, or wait until DQ3 is set to "1" (end of the sector erase timeout period). Figure 18.7-2 shows the judgment method using DQ3 after the sector erase command is issued.

Figure 18.7-2 How to Avoid Problems by Sector Erase Timer Flag (DQ3)

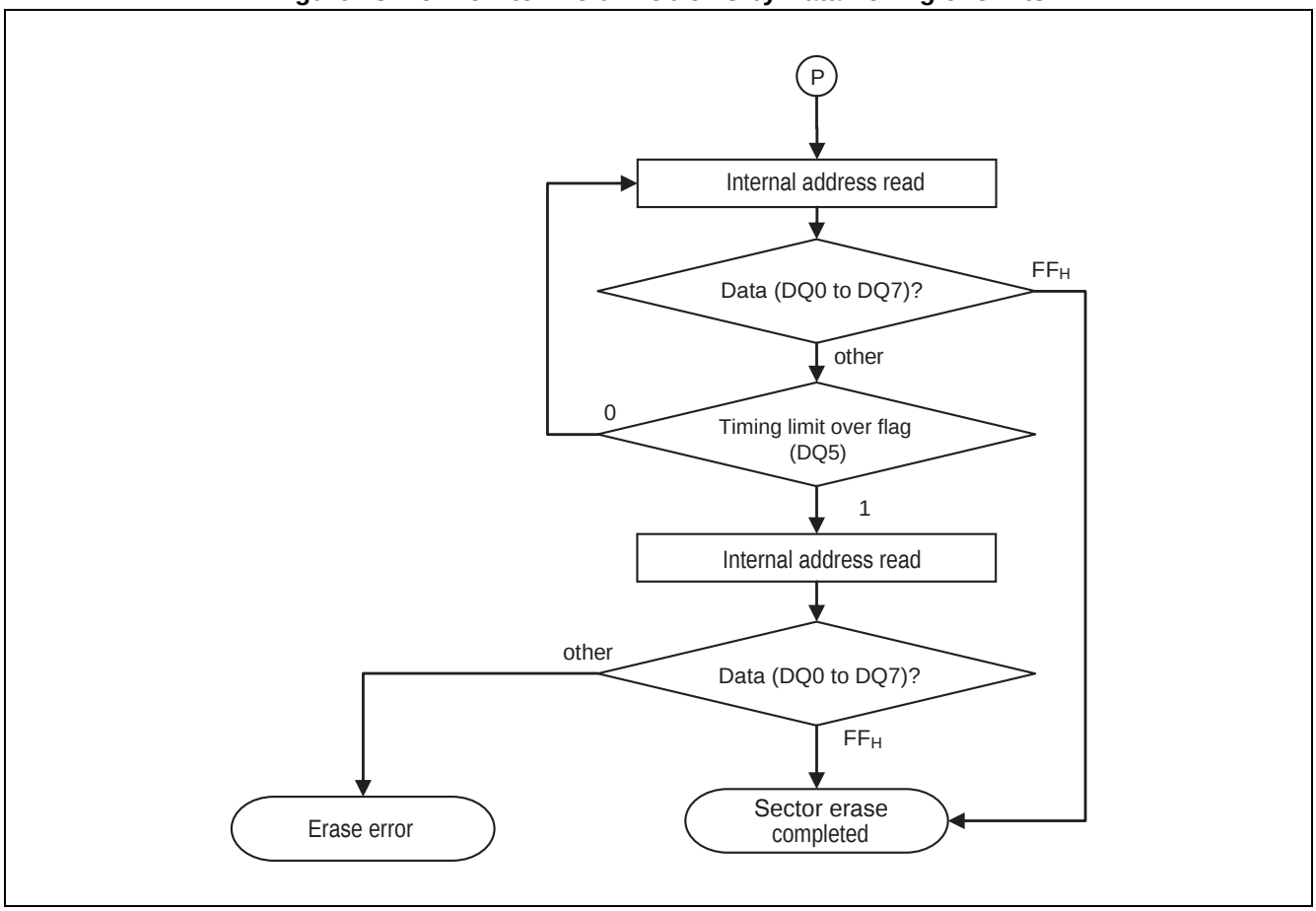


- Data polling using the 8 bits of hardware sequence flags

Make a judgment by data polling using the 8 bits of hardware sequence flags, rather than using only the polling of DQ7.

Figure 18.7-3 shows the judgment method using the data polling of the 8 bits after the sector erase command is issued.

Figure 18.7-3 How to Avoid Problems by Data Polling of 8 Bits



18.8 Restriction and Notes

This section explains the notes on programming flash memory.

■ Restrictions and Notes

(1) Do not execute the write access for the following area (I-bus)

0000_0300_H to 0000_037F_H
0000_03E4_H to 0000_03E7_H
0000_8000_H to 0000_BFFF_H
0001_0000_H to 0001_FFFF_H

(2) Do not execute the write access to the FLASH memory at WE=0 of FLCR register.

(3) Do not execute the continuous write access for the FLASH memory at WE=1 of FLCR register. Be sure to open more than "NOP"1 instruction.

Example: Command write to FLASH (command sequence) => FLASH read

```
ldi     #0xAAAA, r0
ldi     #0x5555, r1
ldi     #0xAAAAA, r6
ldi     #0xA5555, r7
ldi     #0xA0A0, r8
ldi     #PA, r2
ldi     #PD, r3

sth     r0, @r6
nop                 : Be sure to open more than "NOP"1 instruction
sth     r1, @r7
nop                 : Be sure to open more than "NOP"1 instruction
sth     r8, @r6
nop                 : Be sure to open more than "NOP"1 instruction
sth     r3, @r2
nop                 : Be sure to open more than "NOP"1 instruction
```

(4) The write access is enabled only the halfword to FLASH memory at the CPU mode. Do not execute the byte write access.

(5) Do not execute the branch instruction to the FLASH memory area after rewriting the WE of the FLCR register. When branching to the FLASH area, obey the following program example once the FLCR register value is read.

```
STB     R1,     @R2     / / WE=OFF
LDUB    @R2,    R1      / / FLCR value dummy read
BRA     _flash_address  / / Branch to the FLASH memory
```

(6) The sector protect cannot be used.

(7) Do not guarantee the read value immediately after rewriting to FLASH. Before reading is executed after writing, be sure to insert the dummy read.

```
STH     r0,   @r1       / / FLASH write
LDUH    @r2,    r4      / / Dummy read
LDUH    @r3     r4      / / Polling data read
```

CHAPTER 19

MB91313A

SERIAL PROGRAMMING CONNECTION

This chapter explains the serial programming connection for MB91313A.

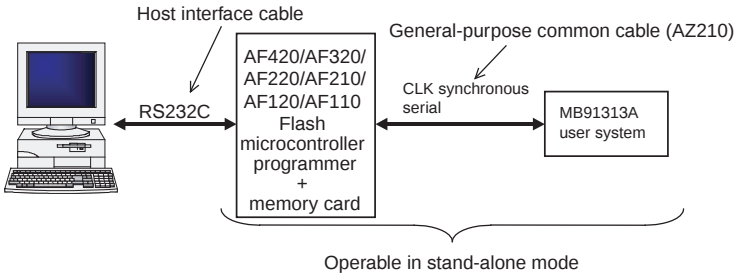
19.1 Basic Configuration of Serial Programming

MB91313A supports serial on-board programming (Fujitsu standard) into flash ROM. This section summarizes its specifications.

■ Basic Configuration of Serial Programming Connection

Fujitsu standard serial on-board programming uses the AF420/AF320/AF220/AF210/AF120/AF110 flash microcontroller programmers by Yokogawa Digital Computer Corporation. The following figure shows the basic configuration for the MB91313A serial programming connection.

It is possible to choose between the program operated in single-chip mode and the program operated in internal ROM external bus mode and to write.

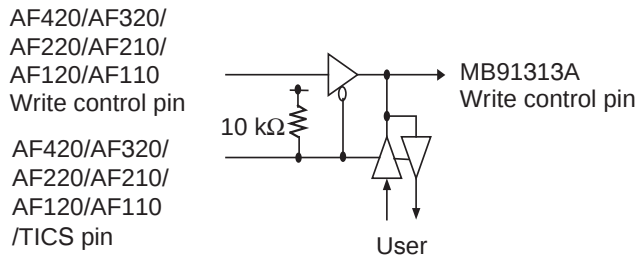


Note:

Contact Yokogawa Digital Computer Corporation for the functions and use of the flash microcontroller programmer and for information on the general-purpose common cable (AZ210) for connection and relevant connectors.

■ Pins Used for Fujitsu-standard Serial On-board Programming

Pin	Function	Supplement
MD2, MD1, MD0	Mode pins	Controls the programming mode. Flash serial programming mode: MD2, MD1, MD0=H, L, L
P54/ASX P55/RDX	Write program start pins	Sets P54="L", P55="H" (clock synchronous mode) during the flash serial reprogramming.
INITX, TRSTX	Reset pins	-
ICD3 (SIN0)	Serial data input pin	Use the ch.0 resource of UART as an interface for the serial on-board programming communication via the DSU pin.
ICS0 (SOT0)	Serial data output pin	
ICD2 (SCK0)	Serial clock input pin	
V_{CC}	Supply-voltage feeder pin	Supply the programming voltage from the user system.
V_{SS}	GND pin	Use this pin also to GND of the flash microcontroller programmer.



Notes:

1. The control circuit shown above is required for the user system to use the Pxx, ICD3, ICS0, and ICD2 pins. (During serial programming, the user circuit can be isolated by the /TICS signal of the flash microcontroller programmer.)
2. Connect the user system to the flash microcontroller programmer with the user power supply off.

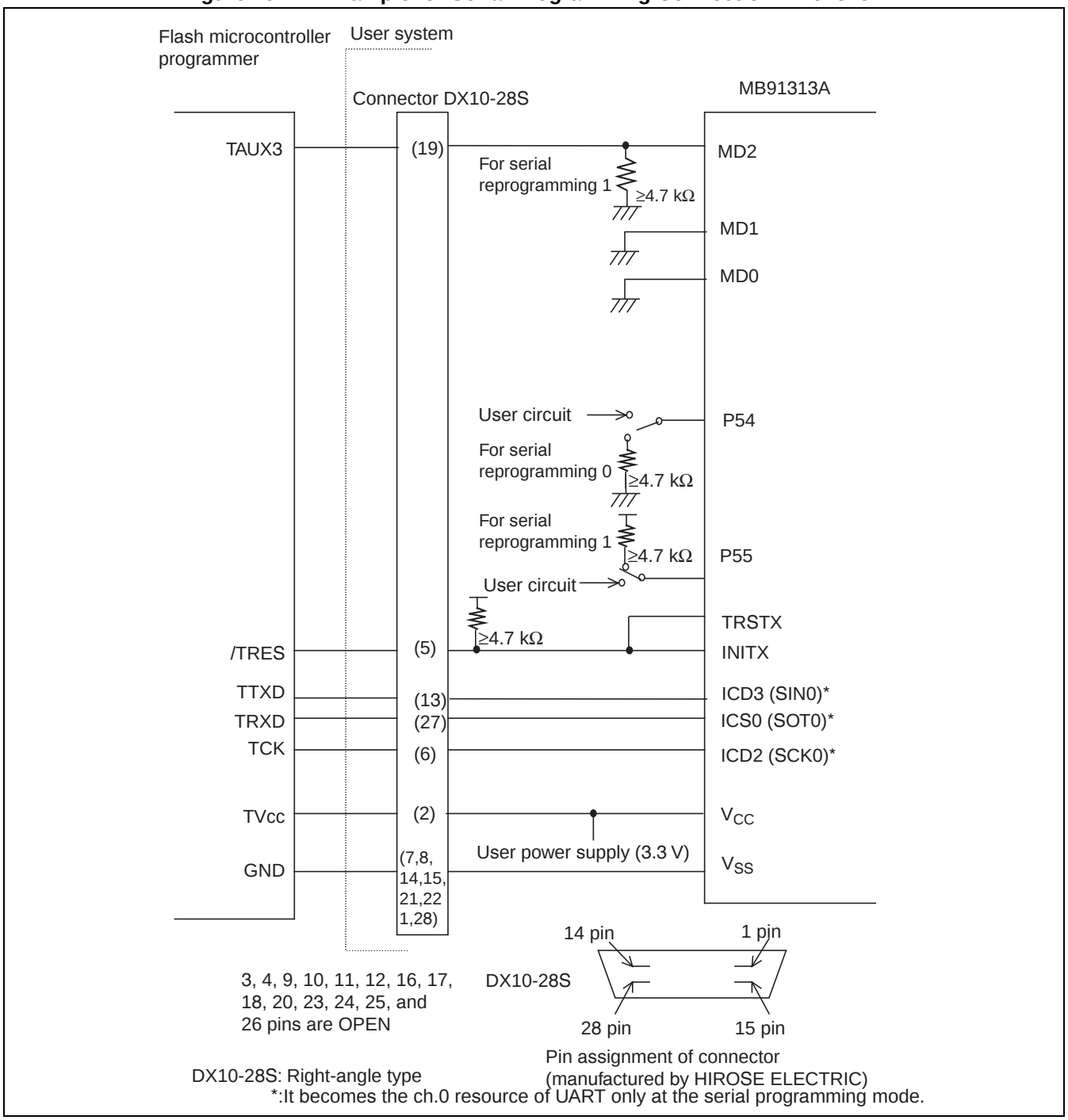
19.2 Example of Serial Programming Connection

This section indicates an example the serial programming connection.

■ Example of Serial Programming Connection

Figure 19.2-1 shows an example of the serial programming connection.

Figure 19.2-1 Example for Serial Programming Connection MB91313A



19.3 System Configuration of Flash Microcontroller Programmer

This section indicates the system configuration of the flash microcontroller programmer (manufactured by Yokogawa Digital Computer Corporation).

■ System Configuration of Flash Microcontroller Programmer

Model			Function	
Main unit	AF220	/AC4P	Ethernet interface model	/100V to 220V power adapter
	AF210	/AC4P	Standard model	/100V to 220V power adapter
	AF120	/AC4P	Single-key Ethernet interface model	/100V to 220V power adapter
	AF110	/AC4P	Single-key model	/100V to 220V power adapter

Contact: Yokogawa Digital Computer Corporation
Phone: (81)-42-333-6224

19.4 Additional Notes

This section explains the note for the serial programming connection for MB91313A.

■ Note

The port state during flash memory programming by the serial programmer is the same as the reset state, except for the pin being used for programming.

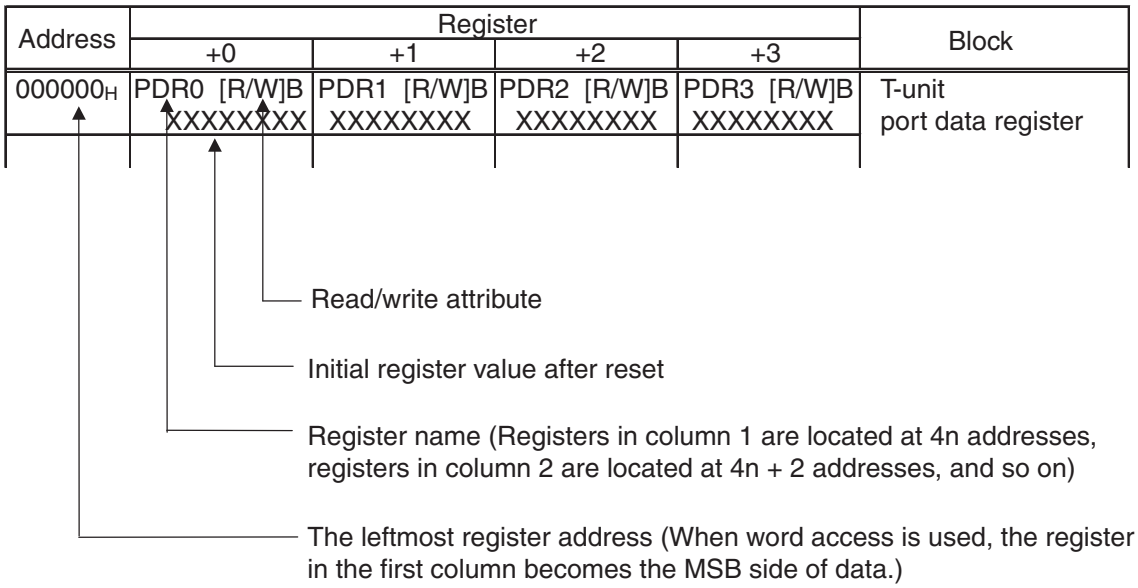
APPENDIX

This appendix explains the details not covered in the main topics and other referential information for programming about the I/O map, interrupt vector, pin status of the CPU state, notes and instructions list when using the little endian area.

APPENDIX A I/O Map

This section shows the correspondence between memory space area and each register of peripheral resources.

■ How to Read I/O Map



The bit value of a register shows an initial value as follows:

"1": Initial value "1"

"0": Initial value "0"

"X": Initial value "Undefine"

"-": No physical register exists at that location.

Appendix A-1 I/O Map (1 / 10)

Address	Register				Block
	0	1	2	3	
000000_H	PDR0 [R/W] XXXXXXXX	PDR1 [R/W] XXXXXXXX	PDR2 [R/W] XXXXXXXX	PDR3 [R/W] XXXXXXXX	Port data register
000004_H	PDR4 [R/W] XXXXXXXX	PDR5 [R/W] XXXXXXXX	PDR6 [R/W] ----XXXX	Reserved	
000008_H	Reserved				
$00000C_H$	PDRC [R/W] XXXXXXXX	PDRD [R/W] XXXXXXXX	PDRE [R/W] XXXXXXXX	PDRF [R/W] XXXXXXXX	
000010_H to $00001C_H$	Reserved				Reserved
000020_H	ADCTH [R/W] 00000000	ADCTL [R/W] 00000000	ADCH [R/W] 00000000_00000000		10-bit A/D converter
000024_H	ADAT0 [R] XXXXXX00_00000000		ADAT1 [R] ------XX XXXXXXXX		
000028_H	ADAT2 [R] XXXXXX00_00000000		ADAT3 [R] XXXXXX00_00000000		
$00002C_H$	ADAT4 [R] XXXXXX00_00000000		ADAT5 [R] XXXXXX00_00000000		
000030_H	ADAT6 [R] XXXXXX00_00000000		ADAT7 [R] XXXXXX00_00000000		
000034_H	ADAT8 [R] XXXXXX00_00000000		ADAT9 [R] XXXXXX00_00000000		
000038_H to $00003C_H$	Reserved				Reserved
000040_H	EIRR0 [R/W] 00000000	ENIR0 [R/W] 00000000	ELVR0 [R/W] 00000000 00000000		External interrupt 0 to 7
000044_H	DICR [R/W] -------0	HRCL [R,R/W] ---11111	Reserved		Delay / I-unit
000048_H	TMRLR0 [W] XXXXXXXX XXXXXXXX		TMR0 [R] XXXXXXXX XXXXXXXX		Reload timer 0
$00004C_H$	Reserved		TMCSR0 [R,RW] 00000000 00000000		
000050_H	TMRLR1 [W] XXXXXXXX XXXXXXXX		TMR1 [R] XXXXXXXX XXXXXXXX		Reload timer 1
000054_H	Reserved		TMCSR1 [R,RW] 00000000 00000000		
000058_H	TMRLR2 [W] XXXXXXXX XXXXXXXX		TMR2 [R] XXXXXXXX XXXXXXXX		Reload timer 2
$00005C_H$	Reserved		TMCSR2 [R,RW] 00000000 00000000		

Appendix A-1 I/O Map (2 / 10)

Address	Register				Block
	0	1	2	3	
000060_H	SCR0 [R,R/W] 0--00000	SMR0 [W,R/W] 000-0000	SSR0 [R,R/W] 0-000011	ESCR0 [R/W] --000000	Serial interface 0 FIFO 0
000064_H	RDR0/TRD0 [R/W] -------- 00000000 : RDR0 -------- 11111111 : TDR0		BGR01 [R/W] 00000000	BGR00 [R/W] 00000000	
000068_H	ISMK0 [R/W] 01111110	IBSA [R/W] 00000000	FCR01 [R/W] 00-00100	FCR00 [R/W] 00000000	
$00006C_H$	FBYTE01 [R/W] 00000000	FBYTE00 [R/W] 00000000	Reserved		
000070_H	SCR1 [R,R/W] 0--00000	SMR1 [W,R/W] 000-0000	SSR1 [R,R/W] 0-000011	ESCR1 [R/W] --000000	Serial interface 1 FIFO 1
000074_H	RDR1/TDR1 [R/W] -------- 00000000 : RDR1 -------- 11111111 : TDR1		BGR11 [R/W] 00000000	BGR10 [R/W] 00000000	
000078_H	ISMK1 [R/W] 01111110	IBSA1 [R/W] 00000000	FCR11 [R/W] 00-00100	FCR10 [R/W] 00000000	
$00007C_H$	FBYTE10 [R/W] 00000000	FBYTE00 [R/W] 00000000	Reserved	Reserved	
000080_H	SCR2 [R,R/W] 0--00000	SMR2 [W,R/W] 000-0000	SSR2 [R,R/W] 0-000011	ESCR2 [R/W] --000000	Serial interface 2
000084_H	RDR2/TDR2 [R/W] -------- 00000000 : RDR2 -------- 11111111 : TDR2		BGR21 [R/W] 00000000	BGR20 [R/W] 00000000	
000088_H	ISMK2 [R/W] 01111110	IBSA2 [R/W] 00000000	FCR21 [R/W] 00000000	FCR20 [R/W] 00000000	
$00008C_H$	FBYTE21 [R/W] 00000000	FBYTE20 [R/W] 00000000	Reserved		
000090_H	SCR3 [R,R/W] 0--00000	SMR3 [W,R/W] 000-0000	SSR3 [R,R/W] 0-000011	ESCR3 [R/W] --000000	Serial interface 3
000094_H	RDR3/TDR3 [R/W] -------- 00000000 : RDR3 -------- 11111111 : TDR3		BGR31 [R/W] 00000000	BGR30 [R/W] 00000000	
000098_H	ISMK3 [R/W] 01111110	IBSA3 [R/W] 00000000	Reserved		
$00009C_H$	Reserved				
$0000A0_H$	SCR4 [R,R/W] 0--00000	SMR4 [W,R/W] 00000000	SSR4 [R,R/W] 0-000011	ESCR4 [R/W] --000000	Serial interface 4
$0000A4_H$	RDR4/TDR4 [R/W] -------- 00000000 : RDR4 -------- 11111111 : TDR4		BGR41 [R/W] 00000000	BGR40 [R/W] 00000000	
$0000A8_H$	ISMK4 [R/W] 01111110	IBSA4 [R/W] 00000000	Reserved		
$0000AC_H$	Reserved				

MB91313A Series

Appendix A-1 I/O Map (3 / 10)

Address	Register				Block
	0	1	2	3	
$0000B0_H$	SCR5 [R,R/W] 0--00000	SMR5 [W,R/W] 00000000	SSR5 [R,R/W] 0-000011	ESCR5 [R/W] --000000	Serial interface 5
$0000B4_H$	RDR5/TDR5 [R/W] -------- 00000000 : RDR5 -------- 11111111 : TDR5		BGR51 [R/W] 00000000	BGR50 [R/W] 00000000	
$0000B8_H$	ISMK5 [R/W] 01111110	IBSA5 [R/W] 00000000	Reserved		
$0000BC_H$	Reserved				
$0000C0_H$	EIRR1 [R/W] 00000000	ENIR1 [R/W] 00000000	ELVR1 [R/W] 00000000 00000000		Ext. INT 8 to INT 15
$0000C4_H$	EIRR2 [R/W] 00000000	ENIR2 [R/W] 00000000	ELVR2 [R/W] 00000000 00000000		Ext. INT 16 to INT 23
$0000C8_H$ to $0000CC_H$	Reserved				Reserved
$0000D0_H$	PWCCL [R/W] 0000--00	PWCCH [R/W] 0--00000	Reserved		PWC
$0000D4_H$	PWCD [R] XXXXXXXX XXXXXXXX		Reserved		
$0000D8_H$	PWCC2 [R/W] 000-----	Reserved			
$0000DC_H$	PWCUD [R/W] XXXXXXXX XXXXXXXX		Reserved		
$0000E0_H$ to $0000EC_H$	Reserved				Reserved
$0000F0_H$	T0LPCR [R/W] -----000	T0CCR [R/W] 0-000000	T0TCR [R/W] 00000000	T0R [R/W] ---00000	Multifunction timer
$0000F4_H$	T0DRR [R/W] XXXXXXXX XXXXXXXX		T0CRR [R/W] XXXXXXXX XXXXXXXX		
$0000F8_H$	T1LPCR [R/W] -----000	T1CCR [R/W] 0-000000	T1TCR [R/W] 00000000	T1R [R/W] ---00000	
$0000FC_H$	T1DRR [R/W] XXXXXXXX XXXXXXXX		T1CRR [R/W] XXXXXXXX XXXXXXXX		
000100_H	T2LPCR [R/W] -----000	T2CCR [R/W] 0-000000	T2TCR [R/W] 00000000	T2R [R/W] ---00000	
000104_H	T2DRR [R/W] XXXXXXXX XXXXXXXX		T2CRR [R/W] XXXXXXXX XXXXXXXX		
000108_H	T3LPCR [R/W] -----000	T3CCR [R/W] 0-000000	T3TCR [R/W] 00000000	T3R [R/W] ---00000	
$00010C_H$	T3DRR [R/W] XXXXXXXX XXXXXXXX		T3CRR [R/W] XXXXXXXX XXXXXXXX		
000110_H	TMODE [R/W] 00000000 00000000		Reserved		
000114_H to $00011C_H$	Reserved				Reserved

Appendix A-1 I/O Map (4 / 10)

Address	Register				Block
	0	1	2	3	
000120_H	PDUT0 [W] XXXXXXXX XXXXXXXX		PCSR0 [W] XXXXXXXX XXXXXXXX		PPG0
000124_H	PTMR0 [W] 11111111 11111111		PCNH0 [R/W] 0000000-	PCNL0 [R/W] 0000000-	
000128_H	PDUT1 [W] XXXXXXXX XXXXXXXX		PCSR1 [W] XXXXXXXX XXXXXXXX		PPG1
$00012C_H$	PTMR1 [W] 11111111 11111111		PCNH1 [R/W] 0000000-	PCNL1 [R/W] 0000000-	
000130_H	PDUT2 [W] XXXXXXXX XXXXXXXX		PCSR2 [W] XXXXXXXX XXXXXXXX		PPG2
000134_H	PTMR2 [W] 11111111 11111111		PCNH2 [R/W] 0000000-	PCNL2 [R/W] 0000000-	
000138_H	PDUT3 [W] XXXXXXXX XXXXXXXX		PCSR3 [W] XXXXXXXX XXXXXXXX		PPG3
$00013C_H$	PTMR3 [W] 11111111 11111111		PCNH3 [R/W] 0000000-	PCNL3 [R/W] 0000000-	
000140_H to 000144_H	Reserved				Reserved
000148_H	TMRLR3 [W] XXXXXXXX XXXXXXXX		TMR3 [R] XXXXXXXX XXXXXXXX		Reload timer 3
$00014C_H$	Reserved		TMCSR3 [R,RW] 00000000 00000000		
000150_H	TMRLR4 [W] XXXXXXXX XXXXXXXX		TMR4 [R] XXXXXXXX XXXXXXXX		Reload timer 4
000154_H	Reserved		TMCSR4 [R,RW] 00000000 00000000		
000158_H	TMRLR5 [W] XXXXXXXX XXXXXXXX		TMR5 [R] XXXXXXXX XXXXXXXX		Reload timer 5
$00015C_H$	Reserved		TMCSR5 [R,RW] 00000000 00000000		
000160_H to $00017C_H$	Reserved				Reserved
000180_H	RCCR0 [R/W] 0---0000	RCST0 [R/W] 00000000	RCSHW0 [R/W] 00000000	RCDAHW0 [R/W] 00000000	Remote control 0
000184_H	RCDBHW0 [R/W] 00000000	Reserved	RCADR01 [R/W] 00000000	RCADR02 [R/W] 00000000	
000188_H	RCDT0HH [R] 00000000	RCDT0HL [R] 00000000	RCDT0LH [R] 00000000	RCDT0LL [R] 00000000	
$00018C_H$	RCCKD0 [R/W] 00000000 00000000		Reserved		
000190_H	RCCR1 [R/W] 0---0000	RCST1 [R/W] 00000000	RCSHW1 [R/W] 00000000	RCDAHW1 [R/W] 00000000	Remote control 1
000194_H	RCDBHW1 [R/W] 00000000	Reserved	RCADR11 [R/W] 00000000	RCADR12 [R/W] 00000000	
000198_H	RCDT1HH [R] 00000000	RCDT1HL [R] 00000000	RCDT1LH [R] 00000000	RCDT1LL [R] 00000000	
$00019C_H$	RCCKD1 [R/W] 00000000 00000000		Reserved		

MB91313A Series

Appendix A-1 I/O Map (5 / 10)

Address	Register				Block
	0	1	2	3	
$0001A0_H$ to $0001AC_H$	Reserved				Reserved
$0001B0_H$	SCR6 [R,R/W] 0--00000	SMR6 [W,R/W] 000-0000	SSR6 [R,R/W] 0-000011	ESCR6 [R/W] --000000	Serial interface 6
$0001B4_H$	RDR6/TDR6 [R/W] -------- 00000000 : RDR6 -------- 11111111 : TDR6		BGR61 [R/W] 00000000	BGR60 [R/W] 00000000	
$0001B8_H$	ISMK6 [R/W] 01111110	IBSA6 [R/W] 00000000	Reserved		
$0001BC_H$	Reserved				
$0001C0_H$	SCR7 [R,R/W] 0--00000	SMR7 [W,R/W] 000-0000	SSR7 [R,R/W] 0-000011	ESCR7 [R/W] --000000	Serial interface 7
$0001C4_H$	RDR7/TDR7 [R/W] -------- 00000000 : RDR7 -------- 11111111 : TDR7		BGR71 [R/W] 00000000	BGR70 [R/W] 00000000	
$0001C8_H$	ISMK7 [R/W] 01111110	IBSA7 [R/W] 00000000	Reserved		
$0001CC_H$	Reserved				
$0001D0_H$	SCR8 [R,R/W] 0--00000	SMR8 [W,R/W] 000-0000	SSR8 [R,R/W] 0-000011	ESCR8 [R/W] --000000	Serial interface 8
$0001D4_H$	RDR8/TDR8 [R/W] -------- 00000000 : RDR8 -------- 11111111 : TDR8		BGR81 [R/W] 00000000	BGR80 [R/W] 00000000	
$0001D8_H$	ISMK8 [R/W] 01111110	IBSA8 [R/W] 00000000	Reserved		
$0001DC_H$	Reserved				
$0001E0_H$	SCR9 [R,R/W] 0--00000	SMR9 [W,R/W] 000-0000	SSR9 [R,R/W] 0-000011	ESCR9 [R/W] --000000	Serial interface 9
$0001E4_H$	RDR9/TDR9 [R/W] -------- 00000000 : RDR9 -------- 11111111 : TDR9		BGR91 [R/W] 00000000	BGR90 [R/W] 00000000	
$0001E8_H$	ISMK9 [R/W] 01111110	IBSA9 [R/W] 00000000	Reserved		
$0001EC_H$	Reserved				
$0001F0_H$	SCRA [R,R/W] 0--00000	SMRA [W,R/W] 000-0000	SSRA [R,R/W] 0-000011	ESCRA [R/W] --000000	Serial interface 10
$0001F4_H$	RDRA/TDRA [R/W] -------- 00000000 : RDRA -------- 11111111 : TDRA		BGRA1 [R/W] 00000000	BGRA0 [R/W] 00000000	
$0001F8_H$	ISMKA [R/W] 01111110	IBSAA [R/W] 00000000	Reserved		
$0001FC_H$	Reserved				

Appendix A-1 I/O Map (6 / 10)

Address	Register				Block
	0	1	2	3	
000200_H	DMACA0 [R/W] 00000000 00000000 00000000 00000000				DMAC
000204_H	DMACB0 [R/W] 00000000 00000000 00000000 00000000				
000208_H	DMACA1 [R/W] 00000000 00000000 00000000 00000000				
$00020C_H$	DMACB1 [R/W] 00000000 00000000 00000000 00000000				
000210_H	DMACA2 [R/W] 00000000 00000000 00000000 00000000				
000214_H	DMACB2 [R/W] 00000000 00000000 00000000 00000000				
000218_H	DMACA3 [R/W] 00000000 00000000 00000000 00000000				
$00021C_H$	DMACB3 [R/W] 00000000 00000000 00000000 00000000				
000220_H	DMACA4 [R/W] 00000000 00000000 00000000 00000000				
000224_H	DMACB4 [R/W] 00000000 00000000 00000000 00000000				
000228_H to $00023C_H$	Reserved				
000240_H	DMACR [R/W] 0XX00000 XXXXXXXX XXXXXXXX XXXXXXXX				
000244_H to $0003EC_H$	Reserved				Reserved
$0003F0_H$	BSD0 [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Bit search module
$0003F4_H$	BSD1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0003F8_H$	BSDC [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0003FC_H$	BSRR [R] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000400_H	DDR0 [R/W] B,H 00000000	DDR1 [R/W] B,H 00000000	DDR2 [R/W] B,H 00000000	DDR3 [R/W] B,H 00000000	Data direction register
000404_H	DDR4 [R/W] B,H 00000000	DDR5 [R/W] B,H 00000000	DDR6 [R/W] B,H ----0000	Reserved	
000408_H	Reserved				
$00040C_H$	DDRC [R/W] B,H -----000	DDRD [R/W] B,H 00000000	DDRE [R/W] B,H 00000000	Reserved	
000410_H	Reserved				
000414_H to $00041C_H$	Reserved				Reserved

MB91313A Series

Appendix A-1 I/O Map (7 / 10)

Address	Register				Block
	0	1	2	3	
000420_H	PFR0 [R/W] B,H 00000000	PFR1 [R/W] B,H 00000000	PFR2 [R/W] B,H 00000000	PFR3 [R/W] B,H 00000000	Port function register
000424_H	PFR4 [R/W] B,H 00000000	PFR5 [R/W] B,H 00000000	PFR6 [R/W] B,H ----0000	Reserved	
000428_H	Reserved				
$00042C_H$	PFRC [R/W] B,H -----000	PFRD [R/W] B,H 00000000	PFRE [R/W] B,H 00000000	–	
000430_H	Reserved				
000434_H to $00043C_H$	Reserved				Reserved
000440_H	ICR00 [R,R/W] -11111	ICR01 [R,R/W] -11111	ICR02 [R,R/W] -11111	ICR03 [R,R/W] -11111	Interrupt control unit
000444_H	ICR04 [R,R/W] -11111	ICR05 [R,R/W] -11111	ICR06 [R,R/W] -11111	ICR07 [R,R/W] -11111	
000448_H	ICR08 [R,R/W] -11111	ICR09 [R,R/W] -11111	ICR10 [R,R/W] -11111	ICR11 [R,R/W] -11111	
$00044C_H$	ICR12 [R,R/W] -11111	ICR13 [R,R/W] -11111	ICR14 [R,R/W] -11111	ICR15 [R,R/W] -11111	
000450_H	ICR16 [R,R/W] -11111	ICR17 [R,R/W] -11111	ICR18 [R,R/W] -11111	ICR19 [R,R/W] -11111	
000454_H	ICR20 [R,R/W] -11111	ICR21 [R,R/W] -11111	ICR22 [R,R/W] -11111	ICR23 [R,R/W] -11111	
000458_H	ICR24 [R,R/W] -11111	ICR25 [R,R/W] -11111	ICR26 [R,R/W] -11111	ICR27 [R,R/W] -11111	
$00045C_H$	ICR28 [R,R/W] -11111	ICR29 [R,R/W] -11111	ICR30 [R,R/W] -11111	ICR31 [R,R/W] -11111	
000460_H	ICR32 [R,R/W] -11111	ICR33 [R,R/W] -11111	ICR34 [R,R/W] -11111	ICR35 [R,R/W] -11111	
000464_H	ICR36 [R,R/W] -11111	ICR37 [R,R/W] -11111	ICR38 [R,R/W] -11111	ICR39 [R,R/W] -11111	
000468_H	ICR40 [R,R/W] -11111	ICR41 [R,R/W] -11111	ICR42 [R,R/W] -11111	ICR43 [R,R/W] -11111	
$00046C_H$	ICR44 [R,R/W] -11111	ICR45 [R,R/W] -11111	ICR46 [R,R/W] -11111	ICR47 [R,R/W] -11111	
000470_H to $00047C_H$	Reserved				Reserved
000480_H	RSRR [R,R/W] 10000000	STCR [R/W] 00110011	TBCR [R/W] 00XXXX00	CTBR [W] XXXXXXXX	Clock control unit
000484_H	CLKR [R/W] 00000000	WPR [W] XXXXXXXX	DIVR0 [R/W] 00000011	DIVR1 [R/W] 00000000	
000488_H	Reserved		OSCCR [R/W] XXXXXXX0	Reserved	
$00048C_H$	WPCR [R/W] 00000000	Reserved			Clock timer
000490_H	OSCR [R/W] 00000000	OSCT [R/W] XXXXXXXX	Reserved		Stabilization wait timer

Appendix A-1 I/O Map (8 / 10)

Address	Register				Block
	0	1	2	3	
000494_H to $0004FC_H$	Reserved				Reserved
000500_H	PCR0 [R/W] 00000000	PCR1 [R/W] 00000000	Reserved		Port pull-up control register
000504_H	Reserved	PCR5 [R/W] 00000000	PCR6 [R/W] ----0000	Reserved	
000508_H to 000510_H	Reserved				
000514_H to $00051C_H$	Reserved				Reserved
000520_H	EPFR0 [R/W] 00000000	EPFR1 [R/W] 00000000	EPFR2 [R/W] 11111111	EPFR3 [R/W] 11111111	Extra port function register
000524_H	EPFR4 [R/W] 11111111	EPFR5 [R/W] 11111111	EPFR6 [R/W] ----1000	Reserved	
000528_H	Reserved				
$00052C_H$	EPFRC [R/W] -----000	EPFRD [R/W] B,H 00000000	EPFRE [R/W] 00000000	EPFRF [R/W] 00000000	
000530_H	Reserved				
000534_H to $00056C_H$	Reserved				Reserved
000570_H	ADER [R/W] H 00000011 11111111		Reserved		EXT/I^2C/AD
000574_H	Reserved				Reserved
000578_H	NSF [R/W] ------00 00000000		Reserved		I^2C noise filter
$00057C_H$ to $00063C_H$	Reserved				Reserved

MB91313A Series

Appendix A-1 I/O Map (9 / 10)

Address	Register				Block
	0	1	2	3	
000640_H	ASR0 [R/W] 00000000 00000000		ACR0 [R/W] 1111XX00 00000000		Ext. bus Interface
000644_H	ASR1 [R/W] 00000000 XXXXXXXX		ACR1 [R/W] 00XX0X00 00X0XXXX		
000648_H	ASR2 [R/W] 00000000 XXXXXXXX		ACR2 [R/W] 00XX0X00 00X0XXXX		
$00064C_H$	ASR3 [R/W] 00000000 XXXXXXXX		ACR3 [R/W] 00XX0X00 00X0XXXX		
000650_H to $00065C_H$	Reserved				
000660_H	AWR0 [R/W] 01110000 01011011		AWR1 [R/W] 0XXX0000 0X0X1XXX		
000664_H	AWR2 [R/W] 0XXX0000 0X0X1XXX		AWR3 [R/W] 0XXX0000 0X0X1XXX		
000668_H to $00067C_H$	Reserved				
000680_H	CSER[R/W] 00000001	Reserved			
000684_H	Reserved				
000688_H to $0007F8_H$	-				Unused
$0007FC_H$	Reserved	MODR [W] XXXXXXXX	Reserved		Mode register
000800_H to $000AFC_H$	-				Unused
$000B00_H$ to $000FFC_H$	Reserved				Reserved

Appendix A-1 I/O Map (10 / 10)

Address	Register				Block
	0	1	2	3	
001000_H	DMASA0 [R/W] 00000000 00000000 00000000 00000000				DMAC
001004_H	DMADA0 [R/W] 00000000 00000000 00000000 00000000				
001008_H	DMASA1 [R/W] 00000000 00000000 00000000 00000000				
$00100C_H$	DMADA1 [R/W] 00000000 00000000 00000000 00000000				
001010_H	DMASA2 [R/W] 00000000 00000000 00000000 00000000				
001014_H	DMADA2 [R/W] 00000000 00000000 00000000 00000000				
001018_H	DMASA3 [R/W] 00000000 00000000 00000000 00000000				
$00101C_H$	DMADA3 [R/W] 00000000 00000000 00000000 00000000				
001020_H	DMASA4 [R/W] 00000000 00000000 00000000 00000000				
001024_H	DMADA4 [R/W] 00000000 00000000 00000000 00000000				
001028_H to $006FFC_H$	Reserved				Reserved
007000_H	FLCR[R/W] 0000X000	Reserved			Flash I/F
007004_H	FLWC[R/W] 00011011	Reserved			

MB91313A Series

APPENDIX B Vector Table

This section indicates the interrupt vector table.

Appendix B-1 Interrupt Vector (1 / 3)

Interrupt Source	Interrupt Number		Interrupt Level	Offset	TBR Default Address	R N	DMAC STOP Source
	Decimal	Hexa-decimal					
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-	-
Mode vector	1	1	-	$3F8_H$	$000FFFF8_H$	-	-
System-reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-	-
System-reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-	-
System-reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-	-
System-reserved	5	5	-	$3E8_H$	$000FFFE8_H$	-	-
System-reserved	6	6	-	$3E4_H$	$000FFFE4_H$	-	-
Coprocessor absence trap	7	7	-	$3E0_H$	$000FFFE0_H$	-	-
Coprocessor error trap	8	8	-	$3DC_H$	$000FFFDC_H$	-	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-	-
System-reserved	10	0A	-	$3D4_H$	$000FFFD4_H$	-	-
System-reserved	11	0B	-	$3D0_H$	$000FFFD0_H$	-	-
Step trace trap	12	0C	-	$3CC_H$	$000FFFCC_H$	-	-
NMI request (tool)	13	0D	-	$3C8_H$	$000FFFC8_H$	-	-
Undefined instruction exception	14	0E	-	$3C4_H$	$000FFFC4_H$	-	-
System-reserved	15	0F	$15(F_H)$ fixation	$3C0_H$	$000FFFC0_H$	-	-
External interrupt 0	16	10	ICR00	$3BC_H$	$000FFFBC_H$	-	-
External interrupt 1	17	11	ICR01	$3B8_H$	$000FFFB8_H$	-	-
External interrupt 2	18	12	ICR02	$3B4_H$	$000FFFB4_H$	-	-
External interrupt 3	19	13	ICR03	$3B0_H$	$000FFFB0_H$	-	-
External interrupt 4	20	14	ICR04	$3AC_H$	$000FFFAC_H$	-	-
External interrupt 5	21	15	ICR05	$3A8_H$	$000FFFA8_H$	-	-
External interrupt 6	22	16	ICR06	$3A4_H$	$000FFFA4_H$	-	-
External interrupt 7	23	17	ICR07	$3A0_H$	$000FFFA0_H$	-	-
Reload timer 0	24	18	ICR08	$39C_H$	$000FFF9C_H$	-	-
Reload timer 1	25	19	ICR09	398_H	$000FFF98_H$	-	-
Reload timer 2	26	1A	ICR10	394_H	$000FFF94_H$	-	-
UART0 RX/I^2C status	27	1B	ICR11	390_H	$000FFF90_H$	0	STOP

Appendix B-1 Interrupt Vector (2 / 3)

Interrupt Source	Interrupt Number		Interrupt Level	Offset	TBR Default Address	R N	DMAC STOP Source
	Decimal	Hexa-decimal					
UART0 TX	28	1C	ICR12	$38C_H$	$000FFF8C_H$	3	-
UART1 RX/I^2C status	29	1D	ICR13	388_H	$000FFF88_H$	1	STOP
UART1 TX	30	1E	ICR14	384_H	$000FFF84_H$	4	-
UART2 RX/I^2C status	31	1F	ICR15	380_H	$000FFF80_H$	2	STOP
UART2 TX	32	20	ICR16	$37C_H$	$000FFF7C_H$	5	-
UART3 RX/UART3 TX/I^2C status	33	21	ICR17	378_H	$000FFF78_H$	-	-
UART4 RX/UART4 TX /I^2C status	34	22	ICR18	374_H	$000FFF74_H$	-	-
UART5 RX/UART5 TX/I^2C status	35	23	ICR19	370_H	$000FFF70_H$	-	-
UART6 RX/UART6 TX /I^2C status	36	24	ICR20	$36C_H$	$000FFF6C_H$	-	-
UART7 RX/UART7 TX /I^2C status	37	25	ICR21	368_H	$000FFF68_H$	-	-
UART8 RX/UART8 TX/I^2C status	38	26	ICR22	364_H	$000FFF64_H$	-	-
UART9 RX/UART9 TX/I^2C status	39	27	ICR23	360_H	$000FFF60_H$	-	-
UART10 RX/UART10 TX/ I^2C status	40	28	ICR24	$35C_H$	$000FFF5C_H$	-	-
A/D Converter	41	29	ICR25	358_H	$000FFF58_H$	-	-
PPG0	42	2A	ICR26	354_H	$000FFF54_H$	12	-
PWC	43	2B	ICR27	350_H	$000FFF50_H$	-	-
HDMI-CEC/Remote Controler0,1	44	2C	ICR28	$34C_H$	$000FFF4C_H$	-	-
Watch timer	45	2D	ICR29	348_H	$000FFF48_H$	-	-
Main clock oscillation wait	46	2E	ICR30	344_H	$000FFF44_H$	-	-
Time-base timer	47	2F	ICR31	340_H	$000FFF40_H$	-	-
Reload timer 3	48	30	ICR32	$33C_H$	$000FFF3C_H$	-	-
Reload timer 4	49	31	ICR33	338_H	$000FFF38_H$	-	-
Reload timer 5	50	32	ICR34	334_H	$000FFF34_H$	-	-
PPG1	51	33	ICR35	330_H	$000FFF30_H$	-	-
PPG2	52	34	ICR36	$32C_H$	$000FFF2C_H$	-	-
PPG3	53	35	ICR37	328_H	$000FFF28_H$	-	-
DMA0	54	36	ICR38	324_H	$000FFF24_H$	-	-
DMA1	55	37	ICR39	320_H	$000FFF20_H$	-	-
DMA2	56	38	ICR40	$31C_H$	$000FFF1C_H$	-	-
DMA3	57	39	ICR41	318_H	$000FFF18_H$	-	-
DMA4	58	3A	ICR42	314_H	$000FFF14_H$	-	-
External interrupt 8 to 15	59	3B	ICR43	310_H	$000FFF10_H$	-	-
External interrupt 16 to 23	60	3C	ICR44	$30C_H$	$000FFF0C_H$	-	-
Multifunction timer 0, 1	61	3D	ICR45	308_H	$000FFF08_H$	-	-

MB91313A Series

Appendix B-1 Interrupt Vector (3 / 3)

Interrupt Source	Interrupt Number		Interrupt Level	Offset	TBR Default Address	R N	DMAC STOP Source
	Decimal	Hexa-decimal					
Multifunction timer 2, 3	62	3E	ICR46	304_H	$000FFF04_H$	-	-
Delayed interrupt	63	3F	ICR47	300_H	$000FFF00_H$	-	-
System-reserved (used in REALOS)	64	40	-	$2FC_H$	$000FFEFC_H$	-	-
System-reserved (used in REALOS)	65	41	-	$2F8_H$	$000FFEF8_H$	-	-
System-reserved	66	42	-	$2F4_H$	$000FFEF4_H$	-	-
System-reserved	67	43	-	$2F0_H$	$000FFEF0_H$	-	-
System-reserved	68	44	-	$2EC_H$	$000FFEEC_H$	-	-
System-reserved	69	45	-	$2E8_H$	$000FFEE8_H$	-	-
System-reserved	70	46	-	$2E4_H$	$000FFEE4_H$	-	-
System-reserved	71	47	-	$2E0_H$	$000FFEE0_H$	-	-
System-reserved	72	48	-	$2DC_H$	$000FFEDC_H$	-	-
System-reserved	73	49	-	$2D8_H$	$000FFED8_H$	-	-
System-reserved	74	4A	-	$2D4_H$	$000FFED4_H$	-	-
System-reserved	75	4B	-	$2D0_H$	$000FFED0_H$	-	-
System-reserved	76	4C	-	$2CC_H$	$000FFECC_H$	-	-
System-reserved	77	4D	-	$2C8_H$	$000FFEC8_H$	-	-
System-reserved	78	4E	-	$2C4_H$	$000FFEC4_H$	-	-
System-reserved	79	4F	-	$2C0_H$	$000FFEC0_H$	-	-
Used for INT instruction	80 to 255	50 to FF	-	$2BC_H$ to 000_H	$000FFEBC_H$ to $000FFC00_H$	-	-

APPENDIX C Pin Status In Each CPU State

This appendix describes the pin status in each CPU state.

Words and phrases used for the pin status have the following meanings.

1. Input enabled

   It means that the input function is allowed to be used.

2. Input 0 fix

   It means that the pin is sending "0" to the internal by blocking the external input at the input gate near the pin.

3. Output Hi-Z

   It means that the transistor for pin drive is disabled and the pin is set to high impedance.

4. Output Retention

   It means that the output status used immediately before becoming this mode is output as it is.

   That is, when internal peripherals are operating, the pin will output by following the peripherals in which the output occurs. When the pin outputs as a port, it will retains the output.

5. Retention of the Status Immediately Before

   It means that the output status immediately before becoming this mode is output as it is, or for input, it means that the input is enabled.

Appendix C-1 Pin Status in Single-Chip Mode (Internal Vector Mode) (1 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial value		HIZ=0	HIZ=1
3	P23	SIN1	P23	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix
4	P24	SOT1/SDA1	P24				
5	P25	SCK1/SCL1	P25				
6	P26	SIN2	P26				
7	P27	SOT2	P27				
8	P30	SCK2	P30				
9	P31	TOT0	P31				
10	P32	TOT1	P32				
11	P33	TOT2	P33				
12	P34	TIN0	P34				
13	P35	TIN1	P35				
14	P36	TIN2	P36				
15	P37	RIN	P37				
16	P40	TMO0/INT16	P40				Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
17	P41	TMO1/INT17	P41				
18	P42	TMO2/INT18	P42				
19	P43	TMO3/INT19	P43				
20	P44	TMI0/INT20	P44				
21	P45	TMI1/INT21/SIN10	P45				
22	P46	TMI2/INT22/ SOT10/SDA10	P46				
23	P47	TMI3/INT23/ SCK10/SCL10	P47				
24	P60	TOT3/TRG2	P60				Output Hi-Z/ Input 0 fix
25	P61	TOT4/TRG3	P61				
26	P62	TOT5	P62				
27	P63	TIN3	P63				
28	P64	TIN4	P64				
29	P65	TIN5	P65				
32	PF0	PF0/RCIN0	PF0				
33	PF1	PF1/RCIN1	PF1				
34	PF2	PF2/RCACK0	PF2				
35	PF3	PF3/RCACK1	PF3				
36	PF4	—	PF4				

Appendix C-1 Pin Status in Single-Chip Mode (Internal Vector Mode) (2 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial value		HIZ=0	HIZ=1
37	PF5	—	PF5	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix
38	PF6	—	PF6				
39	PF7	—	PF7				
45	PD0	AN0	AN0	Output Hi-Z/ Input 0 fix			
46	PD1	AN1	AN1				
47	PD2	AN2	AN2				
48	PD3	AN3	AN3				
49	PD4	AN4	AN4				
50	PD5	AN5	AN5				
51	PD6	AN6	AN6				
52	PD7	AN7	AN7				
53	PE0	AN8/INT0	AN8				Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
54	PE1	AN9/PPG0/INT1	AN9				
55	PE2	PPG1/INT2/ATRG	PE2	Output Hi-Z/ Input enabled			
56	PE3	PPG2/INT3	PE3				
58	—	INITX	INITX	Input enabled	Input enabled	Input enabled	Input enabled
65	—	MD0	MD0				
66	—	MD1	MD1				
67	—	MD2	MD2				
68	PE4	PPG3/INT4	PE4	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
69	PE5	SIN8/INT5	PE5				
70	PE6	SOT8/SDA8/INT6	PE6				
71	PE7	SCK8/SCL8/INT7	PE7				
72	PC0	SIN9	PC0				Output Hi-Z/ Input 0 fix
73	PC1	SOT9/SDA9	PC1				
74	PC2	SCK9/SCL9	PC2				
75	PC3	—	PC3				
76	PC4	PPGA	PC4				
77	PC5	PPGB	PC5				
78	PC6	TRG0	PC6				
79	PC7	TRG1	PC7				
80	—	TRSTX	TRSTX	Input enabled	Input enabled	Input enabled	Input enabled

Appendix C-1 Pin Status in Single-Chip Mode (Internal Vector Mode) (3 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial value		HIZ=0	HIZ=1
81	—	ICD0	ICD0	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Retention of the status immediately before
82	—	ICD1	ICD1				
83	—	ICD2	ICD2				
84	—	ICD3	ICD3				
85	—	ICS0	ICS0	"H" output	"L" output	"L" output	"L" output
86	—	ICS1	ICS1	"L" output	"L" output	"L" output	"L" output
87	—	ICS2	ICS2	"L" output	"L" output	"L" output	"L" output
88	—	ICLK	ICLK	Undefined	"H" output	"H" output	"H" output
89	—	IBREAK	IBREAK	Input enabled	Input enabled	Input enabled	Input enabled

Appendix C-1 Pin Status in Single-Chip Mode (Internal Vector Mode) (4 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial value		HIZ=0	HIZ=1
93	P00	SIN3/INT8	P00	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
94	P01	SOT3/SDA3/INT9	P01				
95	P02	SCK3/SCL3/INT10	P02				
96	P03	SIN4/INT11	P03				
97	P04	SOT4/SDA4/INT12	P04				
98	P05	SCK4/SCL4/INT13	P05				
99	P06	SIN5/INT14	P06				
100	P07	SOT5/SDA5/INT15	P07				
101	P10	SCK5/SCL5	P10				Output Hi-Z/ Input 0 fix
102	P11	SIN6	P11				
103	P12	SOT6/SDA6	P12				
104	P13	SCK6/SCL6	P13				
105	P14	SIN7	P14				
106	P15	SOT7/SDA7	P15				
107	P16	SCK7/SCL7	P16				
108	P17	—	P17				
109	P50	PPG0	P50				
110	P51	PPG1	P51				
111	P52	PPG2	P52				
112	P53	PPG3	P53				
113	P54	—	P54				
114	P55	—	P55				
115	P56	—	P56				
116	P57	—	P57				
117	P20	SIN0	P20				
118	P21	SOT0/SDA0	P21				
119	P22	SCK0/SCL0	P22				

* : P : When set as a general-purpose port F : When specified function is selected

Appendix C-2 Pin Status in Internal ROM External Bus Mode (Internal Vector Mode) (1 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial Value		HIZ=0	HIZ=1
3	P23	SIN1	P23	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix
4	P24	SOT1/SDA1	P24				
5	P25	SCK1/SCL1	P25				
6	P26	SIN2	P26				
7	P27	SOT2	P27				
8	P30	SCK2	P30				
9	P31	TOT0	P31				
10	P32	TOT1	P32				
11	P33	TOT2	P33				
12	P34	TIN0	P34				
13	P35	TIN1	P35				
14	P36	TIN2	P36				
15	P37	RIN	P37				
16	P40	TMO0/INT16	P40				Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
17	P41	TMO1/INT17	P41				
18	P42	TMO2/INT18	P42				
19	P43	TMO3/INT19	P43				
20	P44	TMI0/INT20	P44				
21	P45	TMI1/INT21/SIN10	P45				
22	P46	TMI2/INT22/ SOT10/SDA10	P46				
23	P47	TMI3/INT23/ SCK10/SCL10	P47				
24	P60	TOT3/TRG2	P60				Output Hi-Z/ Input 0 fix
25	P61	TOT4/TRG3	P61				
26	P62	TOT5/RDY	P62		P: Retention of the status immediately before F: Retention of the status immediately before (TOT5) CLK output (CLK)	P: Retention of the status immediately before F: Retention of the status immediately before (TOT5) "H" or "L" output (CLK)	

Appendix C-2 Pin Status in Internal ROM External Bus Mode (Internal Vector Mode) (2 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial Value		HIZ=0	HIZ=1
27	P63	TIN3/CLK	P63	Output Hi-Z/ Input enabled	P: Retention of the status immediately before F: Retention of the status immediately before (TIN3) RDY input (RDY)	P: Retention of the status immediately before F: Retention of the status immediately before (TIN3) Retention of the status immediately before (RDY)	Output Hi-Z/ Input 0 fix
28	P64	TIN4	P64		Retention of the status immediately before	Retention of the status immediately before	
29	P65	TIN5	P65				
32	PF0	PF0/RCIN0	PF0				
33	PF1	PF1/RCIN1	PF1				
34	PF2	PF2/RCACK0	PF2				
35	PF3	PF3/RCACK1	PF3				
36	PF4	—	PF4				
37	PF5	—	PF5				
38	PF6	—	PF6				
39	PF7	—	PF7				
45	PD0	AN0	AN0	Output Hi-Z/ Input 0 fix			
46	PD1	AN1	AN1				
47	PD2	AN2	AN2				
48	PD3	AN3	AN3				
49	PD4	AN4	AN4				
50	PD5	AN5	AN5				
51	PD6	AN6	AN6				
52	PD7	AN7	AN7				
53	PE0	AN8/INT0	AN8				Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
54	PE1	AN9/PPG0/INT1	AN9				
55	PE2	PPG1/INT2/ATRG	PE2	Output Hi-Z/ Input enabled			
56	PE3	PPG2/INT3	PE3				
58	—	INITX	INITX	Input enabled	Input enabled	Input enabled	Input enabled
65	—	MD0	MD0				
66	—	MD1	MD1				
67	—	MD2	MD2				

Appendix C-2 Pin Status in Internal ROM External Bus Mode (Internal Vector Mode) (3 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial Value		HIZ=0	HIZ=1
68	PE4	PPG3/INT4	PE4	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Output Hi-Z/ Input 0 fix (Can only be set as an input when external interrupts are enabled.)
69	PE5	SIN8/INT5	PE5				
70	PE6	SOT8/SDA8/INT6	PE6				
71	PE7	SCK8/SCL8/INT7	PE7				
72	PC0	SIN9	PC0				Output Hi-Z/ Input 0 fix
73	PC1	SOT9/SDA9	PC1				
74	PC2	SCK9/SCL9	PC2				
75	PC3	—	PC3				
76	PC4	PPGA	PC4				
77	PC5	PPGB	PC5				
78	PC6	TRG0	PC6				
79	PC7	TRG1	PC7				
80	—	TRSTX	TRSTX	Input enabled	Input enabled	Input enabled	Input enabled
81	—	ICD0	ICD0	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before	Retention of the status immediately before
82	—	ICD1	ICD1				
83	—	ICD2	ICD2				
84	—	ICD3	ICD3				
85	—	ICS0	ICS0	"H" output	"L" output	"L" output	"L" output
86	—	ICS1	ICS1	"L" output	"L" output	"L" output	"L" output
87	—	ICS2	ICS2	"L" output	"L" output	"L" output	"L" output
88	—	ICLK	ICLK	Undefined	"H" output	"H" output	"H" output
89	—	IBREAK	IBREAK	Input enabled	Input enabled	Input enabled	Input enabled

Appendix C-2 Pin Status in Internal ROM External Bus Mode (Internal Vector Mode) (4 / 4)

Pin No.	Pin name	Function	At initialize (INITX=0)		At sleep*	At stop*	
			Function	Initial Value		HIZ=0	HIZ=1
93	P00	AD00	P00	Output Hi-Z/ Input enabled	P: Retention of the status immediately before F: Output Retention or Hi-Z	P: Retention of the status immediately before F: Output Retention or Hi-Z	Output Hi-Z/ Input 0 fix
94	P01	AD01	P01				
95	P02	AD02	P02				
96	P03	AD03	P03				
97	P04	AD04	P04				
98	P05	AD05	P05				
99	P06	AD06	P06				
100	P07	AD07	P07				
101	P10	AD08	P10				
102	P11	AD09	P11				
103	P12	AD10	P12				
104	P13	AD11	P13				
105	P14	AD12	P14				
106	P15	AD13	P15				
107	P16	AD14	P16				
108	P17	AD15	P17				
109	P50	CS0X/PPG0	P50		P: Retention of the status immediately before F: "H" output	P: Retention of the status immediately before F: "H" output	
110	P51	CS1X/PPG1	P51				
111	P52	CS2X/PPG2	P52				
112	P53	CS3X/PPG3	P53				
113	P54	ASX	P54				
114	P55	RDX	P55				
115	P56	WR0X	P56				
116	P57	WR1X	P57				
117	P20	SIN0	P20		Retention of the status immediately before	Retention of the status immediately before	
118	P21	SOT0/SDA0	P21				
119	P22	SCK0/SCL0	P22				

* : P : When set as a general-purpose port F : When specified function is selected

MB91313A Series

Appendix C-3 Pin status in serial programming mode (1 / 3)

Pin No.	Pin name	Function	At initialize (INITX=0)		P54="L" *	
			Function	Initial Value	P55="H"	P55="L"
3	P23	SIN1	P23	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
4	P24	SOT1/SDA1	P24			
5	P25	SCK1/SCL1	P25			
6	P26	SIN2	P26			
7	P27	SOT2	P27			
8	P30	SCK2	P30			
9	P31	TOT0	P31			
10	P32	TOT1	P32			
11	P33	TOT2	P33			
12	P34	TIN0	P34			
13	P35	TIN1	P35			
14	P36	TIN2	P36			
15	P37	RIN	P37			
16	P40	TMO0/INT16	P40			
17	P41	TMO1/INT17	P41			
18	P42	TMO2/INT18	P42			
19	P43	TMO3/INT19	P43			
20	P44	TMI0/INT20	P44			
21	P45	TMI1/INT21/SIN10	P45			
22	P46	TMI2/INT22/SOT10/SDA10	P46			
23	P47	TMI3/INT23/SCK10/SCL10	P47			
24	P60	TOT3/TRG2	P60			
25	P61	TOT4/TRG3	P61			
26	P62	TOT5	P62			
27	P63	TIN3	P63			
28	P64	TIN4	P64			
29	P65	TIN5	P65			
32	PF0	PF0/RCIN0	PF0			
33	PF1	PF1/RCIN1	PF1			
34	PF2	PF2/RCACK0	PF2			
35	PF3	PF3/RCACK1	PF3			
36	PF4	—	PF4			
37	PF5	—	PF5			
38	PF6	—	PF6	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
39	PF7	—	PF7			

Appendix C-3 Pin status in serial programming mode (2 / 3)

Pin No.	Pin name	Function	At initialize (INITX=0)		P54="L" *	
			Function	Initial Value	P55="H"	P55="L"
45	PD0	AN0	AN0	Output Hi-Z/ Input 0 fix	Output Hi-Z/ Input 0 fix	Output Hi-Z/ Input 0 fix
46	PD1	AN1	AN1			
47	PD2	AN2	AN2			
48	PD3	AN3	AN3			
49	PD4	AN4	AN4			
50	PD5	AN5	AN5			
51	PD6	AN6	AN6			
52	PD7	AN7	AN7			
53	PE0	AN8/INT0	AN8			
54	PE1	AN9/PPG0/INT1	AN9			
55	PE2	PPG1/INT2/ATRG	PE2	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
56	PE3	PPG2/INT3	PE3			
58	—	INITX	INITX	Input enabled	Input enabled	Input enabled
65	—	MD0	MD0			
66	—	MD1	MD1			
67	—	MD2	MD2			
68	PE4	PPG3/INT4	PE4	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
69	PE5	SIN8/INT5	PE5			
70	PE6	SOT8/SDA8/INT6	PE6			
71	PE7	SCK8/SCL8/INT7	PE7			
72	PC0	SIN9	PC0			
73	PC1	SOT9/SDA9	PC1			
74	PC2	SCK9/SCL9	PC2			
75	PC3	—	PC3			
76	PC4	PPGA	PC4			
77	PC5	PPGB	PC5			
78	PC6	TRG0	PC6			
79	PC7	TRG1	PC7			
80	—	TRSTX	TRSTX	Input enabled	Input enabled	Input enabled
81	—	ICD0	ICD0	Output Hi-Z/ Input enabled	Retention of the status immediately before	Retention of the status immediately before
82	—	ICD1	ICD1			
83	—	SCK0	SCK0		Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
84	—	SIN0	SIN0			
85	—	SOT0	SOT0	"H" output	Output state	Output state

MB91313A Series

Appendix C-3 Pin status in serial programming mode (3 / 3)

Pin No.	Pin name	Function	At initialize (INITX=0)		P54="L" *	
			Function	Initial Value	P55="H"	P55="L"
86	—	ICS1	ICS1	"L" output	"L" output	"L" output
87	—	ICS2	ICS2	"L" output	"L" output	"L" output
88	—	ICLK	ICLK	Undefined	"H" output	"H" output
89	—	IBREAK	IBREAK	Input enabled	Input enabled	Input enabled
93	P00	SIN3/INT8	P00	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
94	P01	SOT3/SDA3/INT9	P01			
95	P02	SCK3/SCL3/INT10	P02			
96	P03	SIN4/INT11	P03			
97	P04	SOT4/SDA4/INT12	P04			
98	P05	SCK4/SCL4/INT13	P05			
99	P06	SIN5/INT14	P06			
100	P07	SOT5/SDA5/INT15	P07			
101	P10	SCK5/SCL5	P10			
102	P11	SIN6	P11			
103	P12	SOT6/SDA6	P12			
104	P13	SCK6/SCL6	P13			
105	P14	SIN7	P14			
106	P15	SOT7/SDA7	P15			
107	P16	SCK7/SCL7	P16			
108	P17	—	P17			
109	P50	PPG0	P50			
110	P51	PPG1	P51			
111	P52	PPG2	P52			
112	P53	PPG3	P53			
113	P54	—	P54			
114	P55	—	P55			
115	P56	—	P56			
116	P57	—	P57			
117	P20	—	P20			
118	P21	—	P21	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
119	P22	—	P22			

* : P : When set as a general-purpose port F : When specified function is selected

APPENDIX D Instruction Lists

FR family instruction lists are shown below.

[How to read instruction lists]

Mnemonic	Type	OP	CYC	NZVC	Operation	Remarks
ADD Rj, Rj	A	AG	1	CCCC	Ri + Rj -> Rj	
*ADD #s5, Rj	C	A4	1	CCCC	Ri + s5 -> Ri	
,	,	,	,	,	,	
,	,	,	,	,	,	
(1) (2)	(3)	(4)	(5)	(6)	(7)	

(1) Instruction name

An asterisk (*) indicates an extended instruction that is not contained in the CPU specifications and is obtained by extension or addition by the assembler.

(2) Symbols indicating addressing modes that can be specified for the operand.

For the meaning of symbols, see "Addressing Mode Symbols (on the next page)".

(3) Instruction format

(4) Instruction code in hexadecimal notation

(5) Number of machine cycles

a: Memory access cycle. It may be extended by the Ready function.

b: Memory access cycle. It may be extended by the Ready function.
However, the cycle is interlocked if the instruction immediately after refers to a targeted register for LD operation, and the number of execution cycles is increased by 1.

c: Interlocked if the instruction immediately after is an instruction that reads or writes to R15, SSP, or USP, or an instruction in instruction format A. The number of execution cycles is increased by 1 and so it becomes 2.

d: Interlocked if the instruction immediately after refers to MDH/MDL. The number of execution cycles is increased to 2.
The minimum cycle number is 1 for each case a, b, c, and d.

(6) Indicating flag changes

Flag change
C: Change
-: No change
0: Clear
1: Set

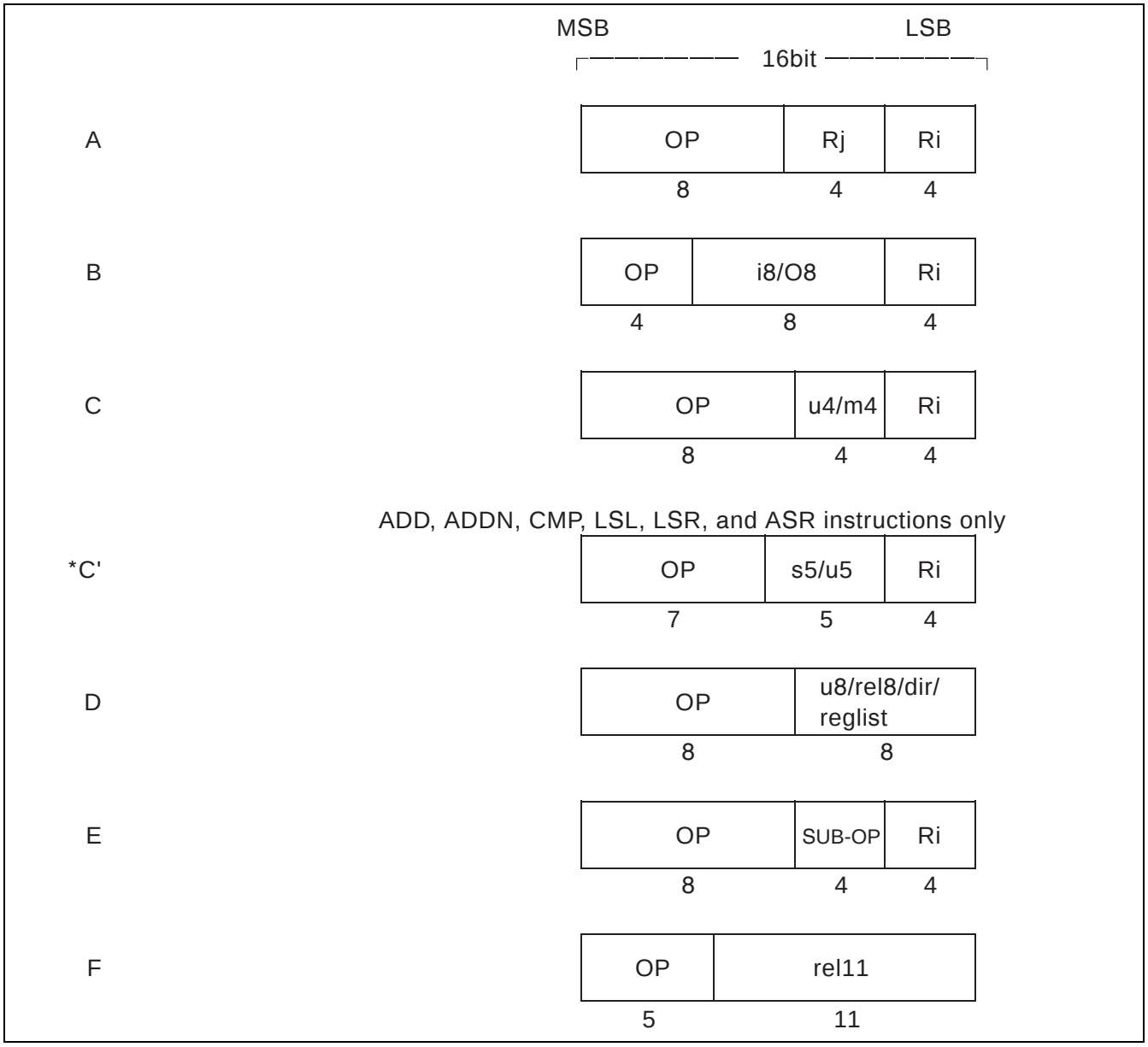
Flag meaning
N: Negative flag
Z: Zero flag
V: Over flag
C: Carry flag

(7) Instruction operation

● Addressing mode symbols

Ri	: Register direct (R0 to R15, AC, FP, SP)
Rj	: Register direct (R0 to R15, AC, FP, SP)
R13	: Register direct (R13, AC)
Ps	: Register direct (Program status register)
Rs	: Register direct (TBR, RP, SSP, USP, MDH, MDL)
Cri	: Register direct (CR0 to CR15)
CRj	: Register direct (CR0 to CR15)
#i8	: Unsigned 8-bit immediate (-128 to 255)
	Note: -128 to -1 is handled as 128 to 255.
#i20	: Unsigned 20-bit immediate (-0X80000 to 0XFFFFF)
	Note: -0X7FFFF to -1 is handled as 0X7FFFF to 0XFFFFF.
#i32	: Unsigned 32-bit immediate (-0X80000000 to 0XFFFFFFFF)
	Note: -0X80000000 to -1 is handled as 0X80000000 to 0XFFFFFFFF.
#s5	: Signed 5-bit immediate (-16 to 15)
#s10	: Signed 10-bit immediate (-512 to 508, multiples of 4 only)
#u4	: Unsigned 4-bit immediate (0 to 15)
#u5	: Unsigned 5-bit immediate (0 to 31)
#u8	: Unsigned 8-bit immediate (0 to 255)
#u10	: Unsigned 10-bit immediate (0 to 1020, multiples of 4 only)
@dir8	: Unsigned 8-bit direct address (0 to 0XFF)
@dir9	: Unsigned 9-bit direct address (0 to 0X1FE, multiples of 2 only)
@dir10	: Unsigned 10-bit direct address (0 to 0X3FC, multiples of 4 only)
label9	: Signed 9-bit branch address (-0X100 to 0XFC, multiples of 2 only)
label12	: Signed 12-bit branch address (-0X800 to 0X7FC, multiples of 2 only)
label20	: Signed 20-bit branch address (-0X80000 to 0X7FFFF)
label32	: Signed 32-bit branch address (-0X80000000 to 0X7FFFFFFF)
@Ri	: Register indirect (R0 to R15, AC, FP, SP)
@Rj	: Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj)	: Register relative indirect (Rj: R0 to R15, AC, FP, SP)
@(R14,disp10)	: Register relative indirect (disp10: -0X200 to 0X1FC, multiples of 4 only)
@(R14,disp9)	: Register relative indirect (disp9: -0X100 to 0XFE, multiples of 2 only)
@(R14,disp8)	: Register relative indirect (disp8: -0X80 to 0X7F)
@(R15,udisp6)	: Register relative indirect (udisp6: 0 to 60, multiples of 4 only)
@Ri+	: Register indirect with post-increment (R0 to R15, AC, FP, SP)
@R13+	: Register indirect with post-increment (R13, AC)
@SP+	: Stack pop
@-SP	: Stack push
(reglist)	: Register list

● Instruction format



Appendix Table D-1 Addition and Subtraction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	Ri+Rj->Ri	The assembler treats the highest-order 1 bit as the sign.
*ADD #s5, Ri	C’	A4	1	CCCC	Ri+s5->Ri	
ADD #u4, Ri	C	A4	1	CCCC	Ri+extu(i4)->Ri	Zero extension
ADD2 #u4, Ri	C	A5	1	CCCC	Ri+extu(i4)->Ri	Minus extension
ADDN Rj, Ri	A	A7	1	CCCC	Ri+Rj+c->Ri	Addition with carry
ADDN Rj, Ri	A	A2	1	----	Ri+Rj->Ri	The assembler treats the highest-order 1 bit as the sign.
*ADDN #s5, Ri	C’	A0	1	----	Ri+s5->Ri	
ADDN #u4, Ri	C	A0	1	----	Ri+extu(i4)->Ri	Zero extension
ADDN2 #u4, Ri	C	A1	1	----	Ri+extu(i4)->Ri	Minus extension
SUB Rj, Ri	A	AC	1	CCCC	Ri-Rj->Ri	
SUBC Rj, Ri	A	AD	1	CCCC	Ri-Rj-c->Ri	Subtraction with carry
SUBN Rj, Ri	A	AE	1	----	Ri-Rj->Ri	

Appendix Table D-2 Comparison Operation

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	Ri-Rj	The assembler treats the highest-order 1 bit as the sign.
*CMP #s5, Ri	C’	A8	1	CCCC	Ri-s5	
CMP #u4, Ri	C	A8	1	CCCC	Ri-extu(i4)	Zero extension
CMP2 #u4, Ri	C	A9	1	CCCC	Ri-extu(i4)	Minus extension

Appendix Table D-3 Logic Operation

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
AND Rj, Ri	A	82	1	CC--	Ri & = Rj	-	Word
AND Rj, @Ri	A	84	1+2a	CC--	(Ri) & = Rj	❍	Word
ANDH Rj, @Ri	A	85	1+2a	CC--	(Ri) & = Rj	❍	Halfword
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	❍	Halfword
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	❍	Halfword
EORB Rj, @Ri	A	9E	1+2a	CC--	(Ri) ^ = Rj	❍	Byte

Appendix Table D-4 Bit Manipulation Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
BANDL #u4, @Ri	C	80	1+2a	----	(Ri)&=(0xF0+u4)	❍	Low-order 4 bits are manipulated.
BANDH #u4, @Ri	C	81	1+2a	----	(Ri)&=((u4<<4)+0x0F)	❍	High-order 4 bits are manipulated.
*BAND #u8, @Ri *1				----	(Ri)&=u8	-	
BORL #u4, @Ri	C	90	1+2a	----	(Ri) | = u4	❍	Low-order 4 bits are manipulated.
BORH #u4, @Ri	C	91	1+2a	----	(Ri) | = (u4<<4)	❍	High-order 4 bits are manipulated.
*BOR #u8, @Ri *2				----	(Ri) | = u8	-	
BEORL #u4, @Ri	C	98	1+2a	----	(Ri) ^ = u4	❍	Low-order 4 bits are manipulated.
BEORH #u4, @Ri	C	99	1+2a	----	(Ri) ^ = (u4<<4)	❍	High-order 4 bits are manipulated.
*BEOR #u8, @Ri *3				----	(Ri) ^ = u8	-	
BTSTL #u4, @Ri	C	88	2+a	0C--	(Ri) & u4	-	Low-order 4 bits are tested.
BTSTH #u4, @Ri	C	89	2+a	CC--	(Ri) & (u4<<4)	-	High-order 4 bits are tested.

*1: The assembler generates BANDL if the bit is set at u8&0x0F, and BANDH if the bit is set at u8&0xF0. In some cases, both BANDL and BANDH may be generated.

*2: The assembler generates BORL if the bit is set at u8&0x0F, and BORH if the bit is set at u8&0xF0. In some cases, both BORL and BORH are generated.

*3: The assembler generates BEORL if the bit is set at u8&0x0F, and BEORH if the bit is set at u8&0xF0. In some cases, both BEORL and BEORH are generated.

Appendix Table D-5 Multiplication and Division

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
MUL Rj,Ri	A	AF	5	CCC-	Ri * Rj -> MDH, MDL	32bit*32bit=64bit
MULU Rj,Ri	A	AB	5	CCC-	Ri * Rj -> MDH, MDL	No sign
MULH Rj,Ri	A	BF	3	CC--	Ri * Rj -> MDL	16bit*16bit=32bit
MULUH Rj,Ri	A	BB	3	CC--	Ri * Rj -> MDL	No sign
DIV0S Ri	E	97-4	1	----		Step operation
DIV0U Ri	E	97-5	1	----		32bit/32bit=32bit
DIV1 Ri	E	97-6	d	-C-C		
DIV2 Ri	E	97-7	1	-C-C		
DIV3	E	9F-6	1	----		
DIV4S	E	9F-7	1	----		
*DIV Ri *1			36	-C-C	MDL / Ri -> MDL , MDL % Ri -> MDH	
*DIVU Ri *2				-C-C	MDL / Ri -> MDL , MDL % Ri -> MDH	

Appendix Table D-6 Shift

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	Ri << Rj -> Ri	Logical shift
*LSL #u5, Ri(u5:0 to 31)	C’	B4	1	CC-C	Ri << u5 -> Ri	
LSL #u4, Ri	C	B4	1	CC-C	Ri << u4 -> Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	Ri <<(u4+16) -> Ri	
LSR Rj, Ri	A	B2	1	CC-C	Ri >> Rj -> Ri	Logical shift
*LSR #u5, Ri(u5:0 to 31)	C’	B0	1	CC-C	Ri >> u5 -> Ri	
LSR #u4, Ri	C	B0	1	CC-C	Ri >> u4 -> Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	Ri >>(u4+16) -> Ri	
ASR Rj, Ri	A	BA	1	CC-C	Ri >> Rj -> Ri	Arithmetic shift
*ASR #u5, Ri (u5:0 to 31)	C’	B8	1	CC-C	Ri >> u5 -> Ri	
ASR #u4, Ri	C	B8	1	CC-C	Ri >> u4 -> Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	Ri >>(u4+16) -> Ri	

Appendix Table D-7 Immediate Value Set/16-Bit/32-Bit Immediate Value Transfer Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LDI:32 #i32, Ri	E	9F-8	3	----	i32 -> Ri	High-order 12 bits are zero-extended.
LDI:20 #i20, Ri	C	9B	2	----	i20 -> Ri	High-order 24 bits are zero-extended.
LDI:8 #i8, Ri	B	C0	1	----	i8 -> Ri	
*LDI # {i8|i20|i32} ,Ri *3					{i8 | i20 | i32} -> Ri	

*1: DIV0S, DIV1 x 32, DIV2, DIV3, and DIV4S are generated. The instruction code length becomes 72 bytes.

*2: DIV0U and DIV1 x 32 are generated. The instruction code length becomes 66 bytes.

*3: If the immediate value is an absolute value, i8, i20, or i32 is selected automatically by the assembler.
If immediate value contains a relative value or an external reference symbol, i32 is selected.

Appendix Table D-8 Memory Load

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LD @Rj, Ri	A	04	b	----	(Rj)->Ri	Rs: Special register
LD @(R13,Rj), Ri	A	00	b	----	(R13+Rj)->Ri	*1
LD @(R14,disp10),Ri	B	2	b	----	(R14+disp10)->Ri	
LD @(R15,udisp6),Ri	C	03	b	----	(R15+udisp6)->Ri	
LD @R15+, Ri	E	07-0	b	----	(R15)->Ri,R15+=4	
LD @R15+, Rs	E	07-8	b	----	(R15)->Rs,R15+=4	
LD @R15+, PS	E	07-9	1+a+b	CCCC	(R15)->PS, R15+=4	
LDUH @Rj, Ri	A	05	b	----	(Rj)->Ri	Zero extension
LDUH @(R13,Rj), Ri	A	01	b	----	(R13+Rj)->Ri	Zero extension
LDUH @(R14,disp9), Ri	B	4	b	----	(R14+disp9)->Ri	Zero extension
LDUB @Rj, Ri	A	06	b	----	(Rj)->Ri	Zero extension
LDUB @(R13,Rj), Ri	A	02	b	----	(R13+Rj)->Ri	Zero extension
LDUB @(R14,disp8), Ri	B	6	b	----	(R14+disp8)->Ri	Zero extension

*1: In the o8 and o4 fields of the hardware specifications, the assembler calculates values and sets them as shown below: disp10/4->o8, disp9/2->o8, disp8->o8, disp10, disp9, and disp8 have a sign, udisp6/4->o4 udisp6 has no sign.

Appendix Table D-9 Memory Store

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
ST Ri,@Rj	A	14	a	----	Ri->(Rj)	Word
ST Ri,@(R13,Rj)	A	10	a	----	Ri->(R13+Rj)	Word
ST Ri,@(R14,disp10)	B	3	a	----	Ri->(R14+disp10)	Word
ST Ri,@(R15,udisp6)	C	13	a	----	Ri->(R15+udisp6)	
ST Ri,@-R15	E	17-0	a	----	R15-=4,Ri->(R15)	
ST Rs,@-R15	E	17-8	a	----	R15-=4, Rs->(R15)	Rs Special register
ST PS,@-R15	E	17-9	a	----	R15-=4, PS->(R15)	*1
STH Ri,@Rj	A	15	a	----	Ri->(Rj)	Halfword
STH Ri,@(R13,Rj)	A	11	a	----	Ri->(R13+Rj)	Halfword
STH Ri,@(R14,disp9)	B	5	a	----	Ri->(R14+disp9)	Halfword
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	7	a	----	Ri->(R14+disp8)	Byte

*1: In the o8 and o4 fields of the hardware specifications, the assembler calculates values and sets them as shown below: disp10/4->o8, disp9/2->o8, disp8->o8, disp10, disp9, and disp8 have a sign, udisp6/4->o4 udisp6 has no sign.

Appendix Table D-10 Register-to-Register Transfer

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	Rj -> Ri	Transfer between general-
MOV Rs, Ri	A	B7	1	----	Rs -> Ri	purpose registers
MOV Ri, Rs	E	B3	1	----	Ri -> Rs	Rs: Special register
MOV PS, Ri	E	17-1	1	----	PS -> Ri	Rs: Special register
MOV Ri, PS	E	07-1	c	CCCC	Ri -> PS	*1

*1: Special register Rs: TBR, RP, USP, SSP, MDH, and MDL

Appendix Table D-11 Normal Branch (No Delay)

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
JMP @Ri	E	97-0	2	----	Ri -> PC	
CALL label12	E	D0	2	----	PC+2->RP , PC+2+(label12-PC-2)->PC	
CALL @Ri	F	97-1	2	----	PC+2->RP ,Ri->PC	
RET	E	97-2	2	----	RP -> PC	Return
INT #u8	D	1F	3+3a	----	SSP-=4, PS->(SSP), SSP-=4, PC+2->(SSP), 0->I Flag, 0->S Flag, (TBR+0x3FC-u8 × 4)->PC	
INTE	E	9F-3	3+3a	----	SSP-=4, PS->(SSP), SSP-=4, PC+2->(SSP), 0->S Flag,(TBR+0x3D8)->PC	For emulator
RETI	E	97-3	2+2A	CCCC	(R15)->PC,R15-=4,(R15)->PS,R15-=4	
BRA label9	D	E0	2	----	PC+2+(label9-PC-2)->PC	
BNO label9	D	E1	1	----	No branch	
BEQ label9	D	E2	2/1	----	if(Z==1) then PC+2+(label9-PC-2)->PC	
BNE label9	D	E3	2/1	----	↑ s/Z==0	
BC label9	D	E4	2/1	----	↑ s/C==1	
BNC label9	D	E5	2/1	----	↑ s/C==0	
BN label9	D	E6	2/1	----	↑ s/N==1	
BP label9	D	E7	2/1	----	↑ s/N==0	
BV label9	D	E8	2/1	----	↑ s/V==1	
BNV label9	D	E9	2/1	----	↑ s/V==0	
BLT label9	D	EA	2/1	----	↑ s/V xor N==1	
BGE label9	D	EB	2/1	----	↑ s/V xor N==0	
BLE label9	D	EC	2/1	----	↑ s/(V xor N) or Z==1	
BGT label9	D	ED	2/1	----	↑ s/(V xor N) or Z==0	
BLS label9	D	EE	2/1	----	↑ s/C or Z==1	
BHI label9	D	EF	2/1	----	↑ s/C or Z==0	

Notes:

- "2/1" under CYCLE indicates "2" cycles when branching occurs and "1" cycle when branching does not occur.
- In the rel11 and rel8 fields of the hardware specifications, the assembler calculates values and sets them as shown below: (label12-PC-2)/2->rel11, (label9-PC-2)/2->rel8, label12 and label9 have a sign.
- To execute the RETI instruction, the S flag must be set to "0".

Appendix Table D-12 Delayed Branch

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
JMP:D @Ri	E	9F-0	1	----	Ri -> PC	
CALL:D label12	F	D8	1	----	PC+4->RP , PC+2+(label12-PC-2)->PC	
CALL:D @Ri	E	9F-1	1	----	PC+4->RP ,Ri->PC	
RET:D	E	9F-2	1	----	RP -> PC	Return
BRA:D label9	D	F0	1	----	PC+2+(label9-PC-2)->PC	
BNO:D label9	D	F1	1	----	No branch	
BEQ:D label9	D	F2	1	----	if(Z==1) then PC+2+(label9-PC-2)->PC	
BNE:D label9	D	F3	1	----	↑ s/Z==0	
BC:D label9	D	F4	1	----	↑ s/C==1	
BNC:D label9	D	F5	1	----	↑ s/C==0	
BN:D label9	D	F6	1	----	↑ s/N==1	
BP:D label9	D	F7	1	----	↑ s/N==0	
BV:D label9	D	F8	1	----	↑ s/V==1	
BNV:D label9	D	F9	1	----	↑ s/V==0	
BLT:D label9	D	FA	1	----	↑ s/V xor N==1	
BGE:D label9	D	FB	1	----	↑ s/V xor N==0	
BLE:D label9	D	FC	1	----	↑ s/(V xor N) or Z==1	
BGT:D label9	D	FD	1	----	↑ s/(V xor N) or Z==0	
BLS:D label9	D	FE	1	----	↑ s/C or Z==1	
BHI:D label9	D	FF	1	----	↑ s/C or Z==0	

Notes:

- In the rel11 and rel8 fields of the hardware specifications, the assembler calculates values and sets them as shown below: (label12-PC-2)/2->rel11, (label9-PC-2)/2->rel8, label12 and label9 have a sign.
- A delayed branch always occurs after the next instruction (delay slot) is executed.
- Instructions that can be placed in the delay slot are all 1-cycle, a-, b-, c-, and d-cycle instructions. Multi-cycle instructions cannot be placed in the delay slot.

Appendix Table D-13 Other Instructions

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
NOP	E	9F-A	1	----	No change	-	
ANDCCR #u8	D	83	c	cccc	CCR and u8 -> CCR	-	
ORCCR #u8	D	93	c	cccc	CCR or u8 -> CCR	-	
STILM #u8	D	87	1	----	i8 -> ILM	-	ILM Immediate set
ADDSP #s10 *1	D	A3	1	----	R15 += s10	-	ADD SP instruction
EXTSB Ri	E	97-8	1	----	Sign extension 8->32bit	-	
EXTUB Ri	E	97-9	1	----	Zero extension 8->32bit	-	
EXTSH Ri	E	97-A	1	----	Sign extension 16->32bit	-	
EXTUH Ri	E	97-B	1	----	Zero extension 16->32bit	-	
LDM0 (reglist)	D	8C		----	(R15)->reglist, R15 increment	-	Load multi R0-R7
LDM1 (reglist)	D	8D		----	(R15)->reglist, R15 increment	-	Load multi R8-R15
*LDM (reglist) *2				----	(R15)->reglist, R15 increment	-	Load multi R0-R15
STM0 (reglist)	D	8E		----	R15 decrement reglist->(R15)	-	Store multi R0-R7
STM1 (reglist)	D	8F		----	R15 decrement reglist->(R15)	-	Store multi R8-R15
*STM (reglist) *3				----	R15 decrement reglist->(R15)	-	Store multi R0-R15
ENTER #u10 *4	D	0F	1+a	----	R14 -> (R15 - 4), R15 - 4 -> R14, R15 - u10 -> R15	-	Entry processing of a function
LEAVE	E	9F-9	b	----	R14 + 4 -> R15, (R15 - 4) -> R14	-	Exit processing of a function
XCHB @Rj, Ri	A	8A	2a	----	Ri -> TEMP (Rj) -> Ri TEMP -> (Rj)	❍	For semaphore management Byte data

*1: For s10, the assembler calculates s10/4 and then changes to s8 to set a value. s10 has a sign.

*2: If any of R0 to R7 is specified in reglist, LDM0 is generated, and if any of R8 to R15 is specified, LDM1 is generated. In some cases, both LDM0 and LDM1 are generated.

*3: If any of R0 to R7 is specified in reglist, STM0 is generated, and if any of R8 to R15 is specified, STM1 is generated. In some cases, both STM0 and STM1 are generated.

*4: For u10, the assembler calculates u10/4 and then changes to u8 to set a value. u10 has no sign.

Notes:

- The number of execution cycles of LDM0(reglist) and LDM1(reglist) can be calculated as a × (n-1)+b+1 cycles if the number of specified registers is n.
- The number of execution cycles of STM0(reglist) and STM1(reglist) can be calculated as a × n+1 cycles if the number of specified registers is n.

Appendix Table D-14 20-Bit Normal Branch Macro Instruction

Mnemonic	Operation	Remarks
*CALL20 label20,Ri	Address of the next instruction ->RP, label20->PC	Ri: Temporary register (See Reference 1)
*BRA20 label20,Ri	label20->PC	Ri: Temporary register (See Reference 2)
*BEQ20 label20,Ri	if(Z==1) then label20->PC	Ri: Temporary register (See Reference 3)
*BNE20 label20,Ri	↑ s/Z==0	↑
*BC20 label20,Ri	↑ s/C==1	↑
*BNC20 label20,Ri	↑ s/C==0	↑
*BN20 label20,Ri	↑ s/N==1	↑
*BP20 label20,Ri	↑ s/N==0	↑
*BV20 label20,Ri	↑ s/V==1	↑
*BNV20 label20,Ri	↑ s/V==0	↑
*BLT20 label20,Ri	↑ s/V xor N==1	↑
*BGE20 label20,Ri	↑ s/V xor N==0	↑
*BLE20 label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20 label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20 label20,Ri	↑ s/C or Z==1	↑
*BHI20 label20,Ri	↑ s/C or Z==0	↑

[Reference 1] CALL20

(1) If label20-PC-2 is between -0x800 and +0x7fe, the following instruction will be generated:

```
CALL label12
```

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:20 #label20,Ri
CALL  @Ri
```

[Reference 2] BRA20

(1) If label20-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
BRA  label9
```

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:20 #label20,Ri
JMP  @Ri
```

[Reference 3] Bcc20

(1) If label20-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
Bcc  label9
```

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
Bxcc  false        xcc is the opposite condition of cc.
LDI:20 #label20,Ri
JMP  @Ri
false:
```

Appendix Table D-15 20-Bit Delayed Branch Macro Instruction

Mnemonic	Operation	Remarks
*CALL20:D label20,Ri	Address of the next instruction +2->RP, label20->PC	RRi: Temporary register (See Reference 1)
*BRA20:D label20,Ri	label20->PC	Ri: Temporary register (See Reference 2)
*BEQ20:D label20,Ri	if(Z==1) then label20->PC	Ri: Temporary register (See Reference 3)
*BNE20:D label20,Ri	↑ s/Z==0	↑
*BC20:D label20,Ri	↑ s/C==1	↑
*BNC20:D label20,Ri	↑ s/C==0	↑
*BN20:D label20,Ri	↑ s/N==1	↑
*BP20:D label20,Ri	↑ s/N==0	↑
*BV20:D label20,Ri	↑ s/V==1	↑
*BNV20:D label20,Ri	↑ s/V==0	↑
*BLT20:D label20,Ri	↑ s/V xor N==1	↑
*BGE20:D label20,Ri	↑ s/V xor N==0	↑
*BLE20:D label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20:D label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20:D label20,Ri	↑ s/C or Z==1	↑
*BHI20:D label20,Ri	↑ s/C or Z==0	↑

[Reference 1] CALL20:D

(1) If label20-PC-2 is between -0x800 and +0x7fe, the following instruction will be generated:

CALL:D label12

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

LDI:20 #label20,Ri

CALL:D @Ri

[Reference 2] BRA20:D

(1) If label20-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

BRA:D label9

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

LDI:20 #label20,Ri

JMP:D @Ri

[Reference 3] Bcc20:D

(1) If label20-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

Bcc:D label9

(2) If label20-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

Bxcc false xcc is the opposite condition of cc.

LDI:20 #label20,Ri

JMP:D @Ri

false:

Appendix Table D-16 32-Bit Normal Branch Macro Instruction

Mnemonic	Operation	Remarks
*CALL32 label32,Ri	Address of the next instruction ->RP, label32->PC	Ri: Temporary register (See Reference 1)
*BRA32 label32,Ri	label32->PC	Ri: Temporary register (See Reference 2)
*BEQ32 label32,Ri	if(Z==1) then label32->PC	Ri: Temporary register (See Reference 3)
*BNE32 label32,Ri	↑ s/Z==0	↑
*BC32 label32,Ri	↑ s/C==1	↑
*BNC32 label32,Ri	↑ s/C==0	↑
*BN32 label32,Ri	↑ s/N==1	↑
*BP32 label32,Ri	↑ s/N==0	↑
*BV32 label32,Ri	↑ s/V==1	↑
*BNV32 label32,Ri	↑ s/V==0	↑
*BLT32 label32,Ri	↑ s/V xor N==1	↑
*BGE32 label32,Ri	↑ s/V xor N==0	↑
*BLE32 label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32 label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32 label32,Ri	↑ s/C or Z==1	↑
*BHI32 label32,Ri	↑ s/C or Z==0	↑

[Reference 1] CALL32

(1) If label32-PC-2 is between -0x800 and +0x7fe, the following instruction will be generated:

```
CALL label12
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:32 #label32,Ri
CALL @Ri
```

[Reference 2] BRA32

(1) If label32-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
BRA label9
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:32 #label32,Ri
JMP @Ri
```

[Reference 3] Bcc32

(1) If label32-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
Bcc label9
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
Bxcc false      xcc is the opposite condition of cc.
LDI:32 #label32,Ri
JMP @Ri
false:
```

Appendix Table D-17 32-Bit Delayed Branch Macro Instruction

Mnemonic	Operation	Remarks
*CALL32D label32,Ri	Address of the next instruction +2->RP, label32->PC	Ri: Temporary register (See Reference 1)
*BRA32:D label32,Ri	label32->PC	Ri: Temporary register (See Reference 2)
*BEQ32:D label32,Ri	if(Z==1) then label32->PC	Ri: Temporary register (See Reference 3)
*BNE32:D label32,Ri	↑ s/Z==0	↑
*BC32:D label32,Ri	↑ s/C==1	↑
*BNC32:D label32,Ri	↑ s/C==0	↑
*BN32:D label32,Ri	↑ s/N==1	↑
*BP32:D label32,Ri	↑ s/N==0	↑
*BV32:D label32,Ri	↑ s/V==1	↑
*BNV32:D label32,Ri	↑ s/V==0	↑
*BLT32:D label32,Ri	↑ s/V xor N==1	↑
*BGE32:D label32,Ri	↑ s/V xor N==0	↑
*BLE32:D label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32:D label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32:D label32,Ri	↑ s/C or Z==1	↑
*BHI32:D label32,Ri	↑ s/C or Z==0	↑

[Reference 1] CALL32:D

(1) If label32-PC-2 is between -0x800 and +0x7fe, the following instruction will be generated:

```
CALL:D label12
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:32 #label32,Ri
CALL:D @Ri
```

[Reference 2] BRA32:D

(1) If label32-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
BRA:D label9
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
LDI:32 #label32,Ri
JMP:D @Ri
```

[Reference 3] Bcc32:D

(1) If label32-PC-2 is between -0x100 and +0xfe, the following instruction will be generated:

```
Bcc:D label9
```

(2) If label32-PC-2 is outside the range of (1) or contains an external reference symbol, the following instruction will be generated:

```
Bxcc false      xcc is the opposite condition of cc.
LDI:32 #label32,Ri
JMP:D @Ri
false:
```

Appendix Table D-18 Direct Addressing

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	(dir10)-> R13	Word
DMOV R13, @dir10	D	18	a	----	R13 ->(dir10)	Word
DMOV @dir10, @R13+	D	0C	2a	----	(dir10)->(R13),R13+=4	Word
DMOV @R13+, @dir10	D	1C	2a	----	(R13)->(dir10),R13+=4	Word
DMOV @dir10, @-R15	D	0B	2a	----	R15-=4,(R15)->(dir10)	Word
DMOV @R15+, @dir10	D	1B	2a	----	(R15)->(dir10),R15+=4	Word
DMOVH @dir9, R13	D	09	b	----	(dir9)-> R13	Halfword
DMOVH R13, @dir9	D	19	a	----	R13 ->(dir9)	Halfword
DMOVH @dir9, @R13+	D	0D	2a	----	(dir9)->(R13),R13+=2	Halfword
DMOVH @R13+, @dir9	D	1D	2a	----	(R13)->(dir9),R13+=2	Halfword
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: In the dir8, dir9, and dir10 fields, the assembler calculates values and sets them as shown below:
dir8->dir, dir9/2->dir, dir10/4->dir dir8, dir9, and dir10 have no sign.

Appendix Table D-19 Resource Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LDRES @Ri+, #u4	C	BC	a	----	Resource of u4 Ri+=4	u4: Channel No.
STRES #u4, @Ri+	C	BD	a	----	Resource of u4->(Ri) Ri+=4	u4: Channel No.

Note: These instructions cannot be used in this MB91313A series since resource having channel number is not installed.

Appendix Table D-20 Coprocessor Control Instruction

{CRi|CRj} := CR0 | CR1 | CR2 | CR3 | CR4 | CR5 | CR6 | CR7 | CR8 | CR9 | CR10 | CR11 | CR12 | CR13 | CR14 | | CR15
u4: := Specify channel
u8: := Specify command

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
COPOP #u4, #u8, CRj, Cri	E	9F-C	2+a	----	Operation instruction	No error stop
COPLD #u4, #u8, Rj, Cri	E	9F-D	1+2a	----	Rj -> CRi	
COPST #u4, #u8, CRj, Ri	E	9F-E	1+2a	----	CRj -> Ri	
COPSV #u4, #u8, CRj, Ri	E	9F-F	1+2a	----	CRj -> Ri	

Note: These instructions cannot be used in this MB91313A series since coprocessor is not installed.

INDEX

The index follows on the next page.
This is listed in alphabetic order.

Index

B

C

D

E

F

G

H

I

R

S

T

CM71-10143-5E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

FR60

32-BIT MICROCONTROLLER

MB91313A Series

HARDWARE MANUAL

August 2010 the fifth edition

Published **FUJITSU SEMICONDUCTOR LIMITED**

Edited Sales Promotion Dept.