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

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

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

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

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

FUJITSU MICROELECTRONICS
CONTROLLER MANUAL

CM26-10110-5E

F^2MC-8FX

8-BIT MICROCONTROLLER

MB95150/M Series

HARDWARE MANUAL

FUJITSU

F^2MC-8FX

8-BIT MICROCONTROLLER

MB95150/M 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 MICROELECTRONICS LIMITED

PREFACE

■ The Purpose and Intended Readership of This Manual

Thank you very much for your continued special support for Fujitsu semiconductor products.

The MB95150/M series is a line of products developed as general-purpose products in the F^2MC-8FX family of proprietary 8-bit single-chip microcontrollers applicable as application-specific integrated circuits (ASICs). The MB95150/M series can be used for a wide range of applications from consumer products including portable devices to industrial equipment.

Intended for engineers who actually develop products using the MB95150/M series of microcontrollers, this manual describes its functions, features, and operations. You should read through the manual.

For details on individual instructions, refer to the "F^2MC-8FX Programming Manual".

Note: F^2MC is the abbreviation of FUJITSU Flexible Microcontroller.

■ Trademark

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

■ Sample Programs

Fujitsu provides sample programs free of charge to operate the peripheral resources of the F^2MC-8FX family of microcontrollers. Feel free to use such sample programs to check the operational specifications and usages of Fujitsu microcontrollers.

Microcontroller support information:

http://www.fujitsu.com/global/services/microelectronics/product/micom/support/index.html

Note that sample programs are subject to change without notice. As these pieces of software are offered to show standard operations and usages, evaluate them sufficiently before use with your system. Fujitsu assumes no liability for any damages whatsoever arising out of the use of sample programs.

CONTENTS

Main changes in this edition

Page	Changes (For details, refer to main body.)	
-	-	Deleted description of the product with step-up circuit and related pins (C0,C1 pin). Deleted MB95FV100D-102.
-	CHAPTER 23 8/10-BIT A/D CONVERTER	Deleted description of Avcc pin and AVss pin.
20	CHAPTER 2 HANDLING DEVICES 2.1 Device Handling Precautions ■ Device Handling Precautions	Added "● Serial Communication"
91	CHAPTER 7 RESET 7.2 Reset Source Register (RSRR) ■ Configuration of Reset Source Register (RSRR) Table 7.2-1	• Corrected description of bit5 This bit varies only with the models equipped with the clock supervisor. → Read or write access (0 or 1) to this bit sets it to "0".
		• Corrected the following descriptions of bit4 to bit1 Read/write access to this bit sets it to "0". → Read or write access (0 or 1) to this bit sets it to "0".
		• Corrected description of bit0 Read/write access to this bit or power-on reset sets it to "0". → Read or write access (0 or 1) to this bit or a power-on reset sets it to "0".
		• Deleted the following descriptions of bit4 to bit0 "• The bit is read only. Writing has no effect on operation."
97	CHAPTER 8 INTERRUPTS 8.1 Interrupts ■ Interrupt Requests from Peripheral Resources Table 8.1-1	Corrected the upper cell of (Vector table address) of (Mode data) $FFFC_H$ → -
112	9.2.2 Operations of Port 0	Changed "● Operation of the pull-up control register"
118	9.3.2 Operations of Port 1	Changed "● Operation of the pull-up control register"
125	9.5 Port 9	• Changed Figure 9.5-1 • Added Figure 9.5-2
142	9.8.2 Operations of Port G	Changed "● Operation of the pull-up control register"
246	15.13 Operating Description of Input Capture Function	Added the explanation.
250	15.16 Notes on Using 8/16-bit Compound Timer	Added the explanation.

Page	Changes (For details, refer to main body.)	
253	16.2 Configuration of 8/16-bit PPG ■ Block Diagram of 8/16-bit PPG	Changed Figure 16.2-1. (MCLK, PCK0 to PCK6 → n/MCLK, $2^7/F_{CH}$, $2^8/F_{CH}$)
268	16.7 Operating Description of 8/16-bit PPG	Changed "● Initial setup". (DDR2 → DDRB)
271	16.7.2 8-bit Prescaler + 8-bit PPG Mode	Changed "■ Operation of 8-bit Prescaler + 8-bit PPG Mode". (the PPG timer 00 (ch.0) → the PPG timer 01 (ch.0))
334	CHAPTER 20 UART/SIO 20.5 Registers of UART/SIO ■ Registers Related to UART/SIO Figure 20.5-1	Corrected bit attribute of bit5 in SMC20 R/W → R1/W Added explanation R1/W : Readable/writable (Read value is always "1")
338	CHAPTER 20 UART/SIO 20.5.2 UART/SIO Serial Mode Control Register 2 (SMC20) ■ UART/SIO Serial Mode Control Register 2 (SMC20) Table 20.5-2	Corrected bit description of bit5 Setting the bit to "1" clears the reception error flag. → Setting the bit to "1": has no effect on operation.
382	CHAPTER 22 LIN-UART 22.4.1 LIN-UART Serial Control Register (SCR) ■ LIN-UART Serial Control Register (SCR) Table 22.4-1	• Deleted Note in Function cell of bit5 • Changed Note in Function cell of bit2
395	22.5 Interrupt of LIN-UART ■ Reception Interrupt ● Reception interrupt	Changed Note:
411	22.7 Operations and Setting Procedure Example of LIN-UART ■ Setup Procedure Example ● Initial setting	Corrected explanation 1) Set the port input (DDR1). → 1) Set the port for input (DDR6).
433 to 436	22.8 Notes on Using LIN-UART ■ Notes on Using LIN-UART	Added ● Handling framing errors Added Figure22.8-1 to Figure 22.8-3
437	22.9 Sample Programs of LIN-UART ■ Setting Methods not Covered by Sample Programs ● How to control the SCK, SIN, and SOT pins	Corrected the LIN-UART setting for "To set the SCK pin as input" DDR6:P05 = 0 → DDR6:P65 = 0 Corrected the LIN-UART setting for "To use the SIN pin" DDR6:P07 = 0 → DDR6:P67 = 0

Page	Changes (For details, refer to main body.)	
458	CHAPTER 23 8/10-BIT A/D CONVERTER 23.6 Operations of 8/10-bit A/D Converter and Its Setup Procedure Examples ■ Setup Procedure Example ● Initial setting	Corrected explanation 1) Set the port for input (DDR1) → 1) Set the port for input (DDR0)
465	24.3 Pins of LCD Controller	Changed Figure 24.3-1.
512	CHAPTER 27 256-KBIT FLASH MEMORY 27.1 Overview of 256-Kbit Flash Memory	Changed summary
	■ Overview of 256-Kbit Flash Memory	Changed descriptions
516	27.3.1 Flash Memory Status Register (FSR) ■ Flash Memory Status Register (FSR) Table 27.3-1	Deleted the following description of bit1 • To program data into the flash memory, set FSR:WRE to "1" to write-enable the flash memory and set the flash memory sector write control register (SWRE0/SWRE1). When FSR:WRE disables programming (contains "0"), write access to flash memory does not take place even though it is enabled by the flash memory write control register (SWRE0/SWRE1)."
517	27.4 Starting the Flash Memory Automatic Algorithm ■ Command Sequence Table Table 27.4-1	Changed explanation of "U" U : Upper 4 bits same as RA, PA, and SA → U : Upper 4 bits same as RA and PA
518	27.5 Checking the Automatic Algorithm Execution Status ■ Hardware Sequence Flag ● Overview of hardware sequence flag	Changed description consists of the following 5-bit outputs → consists of the following 3-bit outputs Deleted "Toggle bit 2 flag (DQ2)" Deleted "Note, however, that hardware sequence flags are output only for the bank on a command-issued side."
	Table 27.5-1	Changed bit 2 DQ2 → -
	Table 27.5-2	Deleted column of "DQ2" Deleted Note(*) at the bottom of the table
-	27.5.4 Toggle Bit 2 Flag (DQ2)	Deleted whole section of 27.5.4
542	APPENDIX A I/O Map	Changed the column of 0073_H,0074_H,0075_H to "Disabled".
545	APPENDIX B Table of Interrupt Causes	Changed Table B-1. ($FFFC_H$ → -)

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

CHAPTER 1
DESCRIPTION

This chapter explains a feature and a basic specification of the MB95150/M series.

1.1 Feature of MB95150/M Series

In addition to a compact instruction set, the MB95150/M series is a general-purpose single-chip microcontroller built-in abundant peripheral functions.

■ Feature of MB95150/M Series

● F²MC-8FX CPU core

Instruction system optimized for controllers

- Multiplication and division instructions
- 16-bit operation
- Bit test branch instruction
- Bit operation instructions etc.

● Clock

- Main clock
- Main PLL clock
- Sub clock
- Sub PLL clock

● Timer

- 8/16-bit compound timer × 2 channels
  Can be used as interval timer, PWC timer, PWM timer and input capture.
- 8/16-bit PPG × 2 channels
- 16-bit PPG × 1 channel
- Time-base timer × 1 channel
- Watch prescaler × 1 channel

● LIN-UART × 1 channel

- LIN function, an asynchronous clock or a synchronous serial data transfer can be used
- With full-duplex double buffer

● UART/SIO × 1 channel

- An asynchronous clock or a synchronous serial data transfer can be used
- With full-duplex double buffer

● External interrupt × 8 channels

- Interrupt by the edge detection (Select from rising edge, falling edge, or both edges)
- Can be used to recover from low-power consumption (standby) mode

● 8/10-bit A/D converter × 8 channels

8-bit or 10-bit resolutions can be selected

- LCD controller (LCDC)
  - 16 SEG × 4 COM (Max 64 pixels)
  - With blinking function

- Low-power consumption (standby) mode
  - Stop mode
  - Sleep mode
  - Watch mode
  - Time-base timer mode

- I/O port: Max 39

  General-purpose I/O ports (CMOS): 39

- Programmable input voltage levels of port

  Automotive input level / CMOS input level / hysteresis input level

- Flash memory security function

  Protects the content of Flash memory (Flash memory device only)

1.2 Product Lineup of MB95150/M Series

MB95150/M series is available in three types. Table 1.2-1 lists the product lineup and Table 1.2-2 lists the CPUs and peripheral functions.

■ Product Lineup of MB95150/M Series

Table 1.2-1 Product Lineup of MB95150/M Series

Classification *2		Product	ROM/RAM	Voltage	Option			Reset output	LCDC
					Clock system	LVD	CSV		
Evaluation products*1		MB95FV100D-101	60KB/3.75KB	3V	Single clock product	None	None	None	Built-in internal split resistor
					Dual clock product				
		MB95FV100D-103	60KB/3.75KB	5V	Single clock product	None	None	Yes	
						Yes	None		
						Yes	Yes	None	
					Dual clock product	None	None	Yes	
						Yes	None		
						Yes	Yes	None	
5V products	Flash memory products	MB95F156M	32KB/1KB	5V	Dual clock product	None	None	Yes	
		MB95F156N				Yes	None		
		MB95F156J				Yes	Yes	None	
5V products	Mask ROM products *3	MB95156M	32KB/1KB	5V	Dual clock product	None	None	Yes	
						Yes	None		
						Yes	Yes	None	

LVD: Low-voltage detection reset

CSV: Clock supervisor

*1: In evaluation products, please switch Yes/None of single clock/dual clock, LVD and CSV with the switch on the MCU board. (Specification with LVD none and CSV cannot be done.)

*2: 3V products is now planning.

*3: In mask ROM products, please specify Yes/None of LVD and CSV at the product order. (Specification with LVD none and CSV cannot be done.)

Table 1.2-2 CPU and Peripheral Function of MB95150/M Series

Item		Specification
CPU function		Number of basic instructions: 136 instructions Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, and 16 bits Minimum instruction execution time: 61.5 ns (at machine clock 10 MHz) Interrupt processing time: 0.6 µs (at machine clock 16.25 MHz)
Peripheral function	Port	General-purpose I/O ports (CMOS): 39 (Max)
	Time-base timer	1ch. Interrupt cycle: 0.5 ms, 2.1 ms, 8.2 ms, 32.8 ms (at external 4 MHz)
	Watchdog timer	Reset generation cycle Main clock at 10 MHz : 105 ms (Min) Sub clock at 32.768 kHz: 250 ms (Min)
	Wild registers	ROM data for three bytes can be replaced
	UART/SIO	1ch. Data transfer is enabled at UART/SIO Built-in full-duplex double buffer, Changeable data length (5/6/7/8-bit), Built-in baud rate generator Transfer rate: 2400bps to 125000bps NRZ method transfer format, Error detected function LSB-first or MSB-first can be selected Serial data transfer is available for clock synchronous (SIO) and clock asynchronous (UART)
	LIN-UART	1ch. A wide-range communication speed can be set with the dedicated reload timer Full-duplex double buffer Serial data transfer is available for clock synchronous and clock asynchronous LIN function can be used as a LIN master and LIN slave
	8/10-bit A/D converter	8ch. 8-bit or 10-bit resolution can be selected
	8/16-bit compound timer	2ch. Can be configured as a 2ch × 8-bit timer or 1ch × 16-bit timer per each timer channel Built-in timer function, PWC function, PWM function and capture function Count clock: available from internal clocks (7 types) or external clocks With square wave output
	16-bit PPG	1ch. PWM mode or one-shot mode can be selected Counter operation clock: available from eight selectable clock sources Support for external trigger activation
	8/16-bit PPG	2ch. Can be configured as a 2ch × 8-bit PPG or 1ch × 16-bit PPG per each PPG channel Counter operation clock: available from eight selectable clock sources
	Watch counter	Count clock: available from four selectable clock sources (125 ms, 250 ms, 500 ms, or 1 s) Counter value can be set within the range of 0 to 63 (When one second is selected as for the clock source and the counter value is set to 60, it is possible to count for one minute.)
	Watch prescaler	1ch. Available from four selectable interval times (125 ms, 250 ms, 500 ms, 1 s)
	LCD controller (LCDC)	Max. 16 SEG × 4 COM Can be operated in LCD standby mode Blinking function
	External interrupt	8ch. Interrupt by edge detection (Possible to select from rising edge, falling edge or both edges) Can be used to recover from standby modes
	Flash memory	Supports automatic programming, Embedded Algorithm™ Write/Erase/Erase-Suspend/Resume commands A flag indicating completion of the algorithm Number of write/erase cycles (Minimum): 10000 times Data retention time: 20 years Erase can be performed on each block Block protection with external programming voltage Flash Security Feature for protecting the content of the Flash
Standby Mode		Sleep, stop, watch (Only for dual clock product), and time-base timer

1.3 Difference Points among Products and Notes on Selecting the Product

The following describes differences among MB95150/M series products and notes when selecting the product.

■ Difference Points among Products and Notes on Selecting a Product

● Notes on using evaluation products

The evaluation products are intended to support software development for a number of different F^2MC-8FX family series and products, and it therefore includes additional functions that may not be included in MB95150/M series. Accordingly, access to I/O address of peripheral functions that are not used in MB95150/M series are prohibited.

Reading or writing to these prohibited addresses may cause these unused peripheral functions to operate and lead to unexpected hardware or software problems.

Take particular care not to use word access to read or write odd numbered bytes in the prohibited areas (It causes unexpected read/write operation). Also, as the read values of prohibited addresses on the evaluation product are different to the values on the flash memory and mask ROM products, do not use these values in the program.

The functions corresponding to certain bits in single-byte registers may not be supported on some mask ROM and flash memory products. However, reading or writing to these bits will not cause malfunction of the hardware. Also, as the evaluation, flash memory, and mask ROM products are designed to have identical software operation, no particular precautions are required.

● Difference of memory space

If the memory size on the evaluation product is different to the flash memory or mask ROM product, please ensure you understand these differences when developing software.

● 8/10-bit A/D converter

The analog characteristics of the 8/10-bit A/D converter analog input pins on the evaluation product are slightly different to the ones on the flash memory and mask ROM products. Please refer to the data sheet for the differences in analog input impedance enough before designing the external circuit.

● Current consumption

The current consumption of flash memory products is greater than for mask ROM products.

For the details of current consumption, refer to "■ Package and its Corresponding Product" and "Electric characteristics" in data sheet.

● Package

For detailed information on each package, refer to "1.6 Package Dimension".

- Operating voltage

The operating voltage may be different depending on the products. For the details, see the "electrical characteristics" of "data sheet".

- Difference of $\overline{\text{RST}}$/MOD pins

For the mask ROM products, the $\overline{\text{RST}}$ and MOD pins are hysteresis inputs (However, on 5 V products, these pins are hysteresis inputs for both mask ROM products and flash memory products). And, a pull-down resistor is provided for the MOD pin.

■ Package and Its Corresponding Product

Product / Package	MB95156M	MB95F156M MB95F156N MB95F156J	MB95FV100D-101 MB95FV100D-103
FPT-48P-M26	❍	❍	×
FPT-52P-M01	❍	❍	×
BGA-224P-M08	×	×	❍

❍ : usable

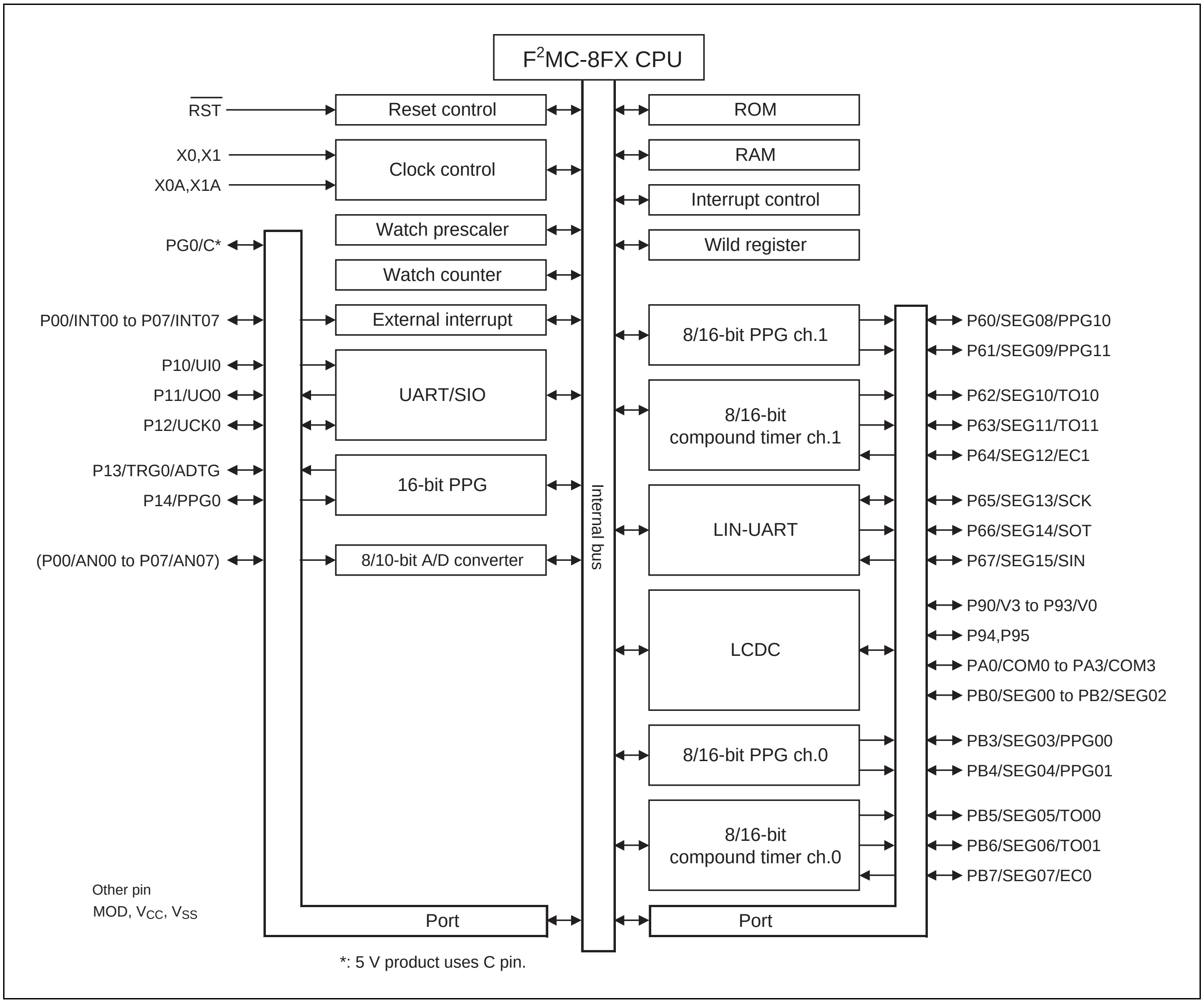
× : unusable

1.4 Block Diagram of MB95150/M Series

Figure 1.4-1 shows the block diagram of all MB95150/M series.

■ Block Diagram of All MB95150/M Series

Figure 1.4-1 Block Diagram of All MB95150/M Series



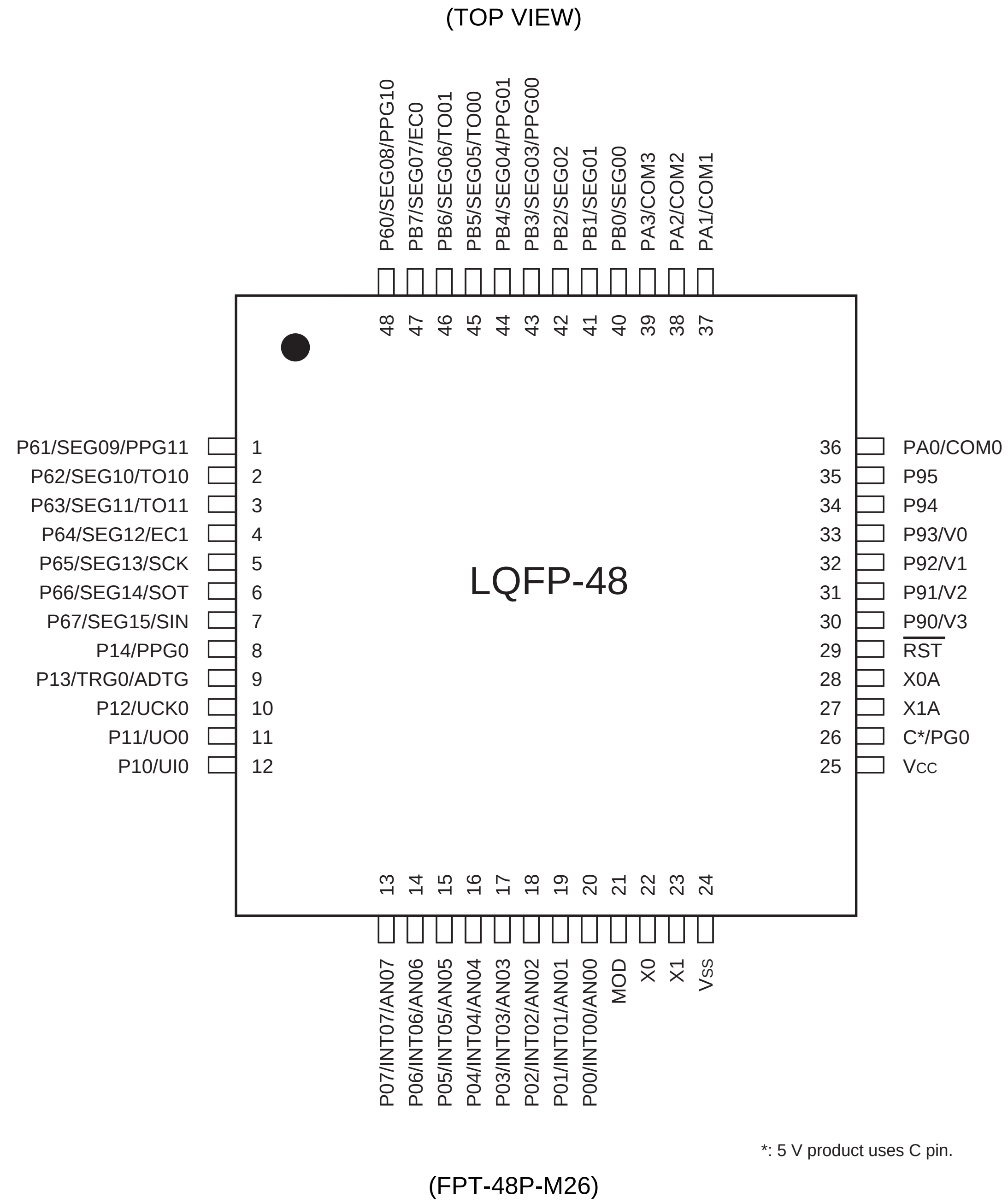
MB95150/M Series

1.5 Pin Assignment

Figure 1.5-1 shows the pin assignment of the MB95150/M series.

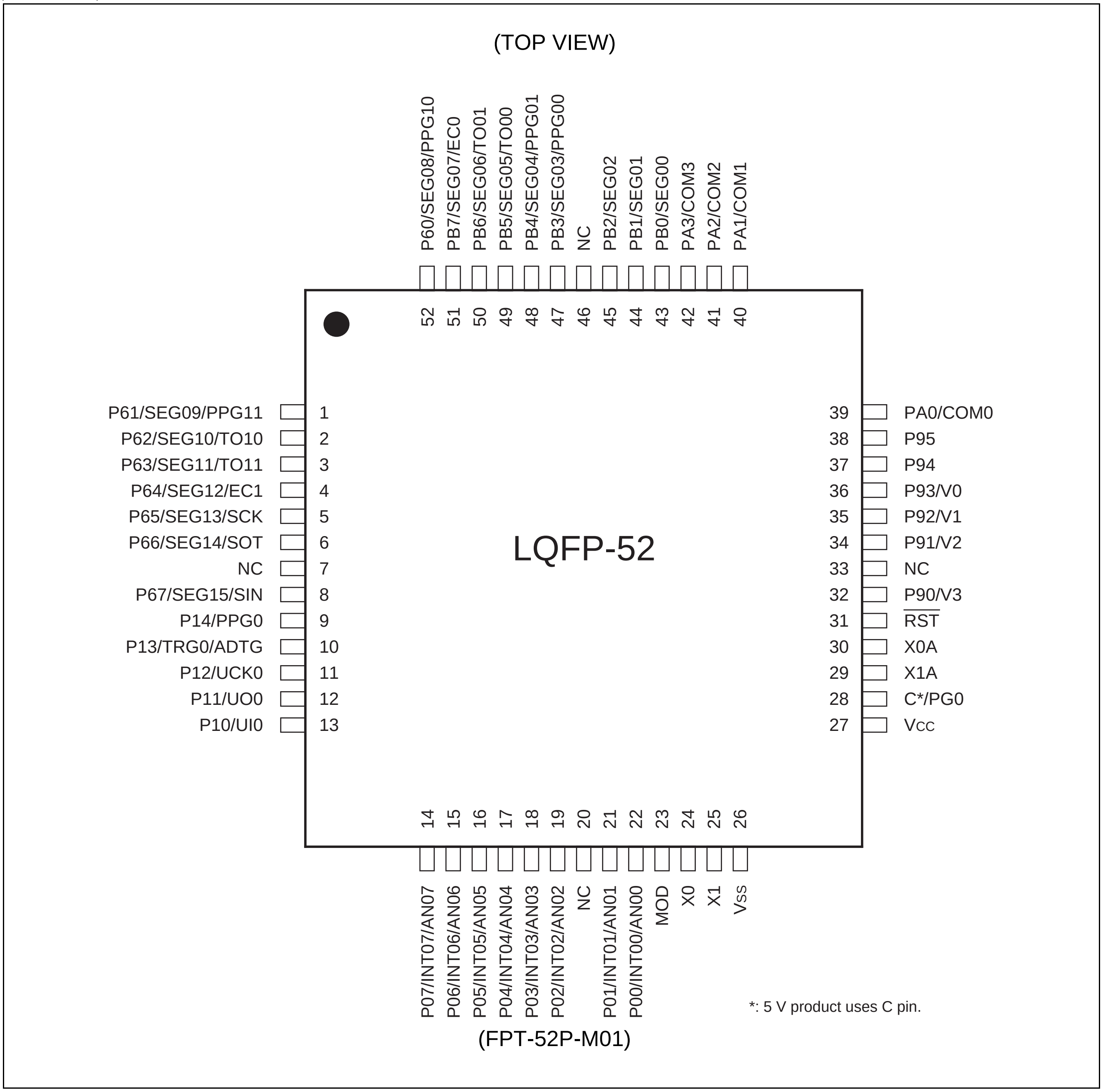
■ Pin Assignment of MB95150/M Series

Figure 1.5-1 Pin Assignment of MB95150/M Series



(Continued)

(Continued)


(TOP VIEW)
P60/SEG08/PPG10
PB7/SEG07/EC0
PB6/SEG06/TO01
PB5/SEG05/TO00
PB4/SEG04/PPG01
PB3/SEG03/PPG00
NC
PB2/SEG02
PB1/SEG01
PB0/SEG00
PA3/COM3
PA2/COM2
PA1/COM1
52 51 50 49 48 47 46 45 44 43 42 41 40
P61/SEG09/PPG11 1
P62/SEG10/TO10 2
P63/SEG11/TO11 3
P64/SEG12/EC1 4
P65/SEG13/SCK 5
P66/SEG14/SOT 6
NC 7
P67/SEG15/SIN 8
P14/PPG0 9
P13/TRG0/ADTG 10
P12/UCK0 11
P11/UO0 12
P10/UI0 13
LQFP-52
39 PA0/COM0
38 P95
37 P94
36 P93/V0
35 P92/V1
34 P91/V2
33 NC
32 P90/V3
31 RST
30 X0A
29 X1A
28 C*/PG0
27 Vcc
14 15 16 17 18 19 20 21 22 23 24 25 26
P07/INT07/AN07
P06/INT06/AN06
P05/INT05/AN05
P04/INT04/AN04
P03/INT03/AN03
P02/INT02/AN02
NC
P01/INT01/AN01
P00/INT00/AN00
MOD
X0
X1
Vss
*: 5 V product uses C pin.
(FPT-52P-M01)

MB95150/M Series

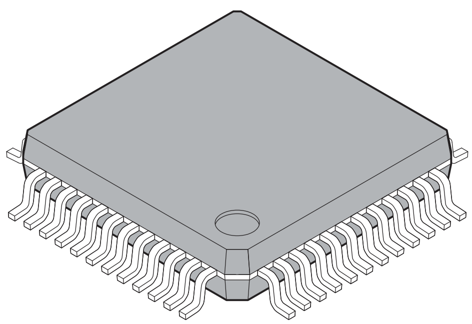
1.6 Package Dimension

MB95150/M series is available in two types of package.

■ Package Dimension of FPT-48P-M26

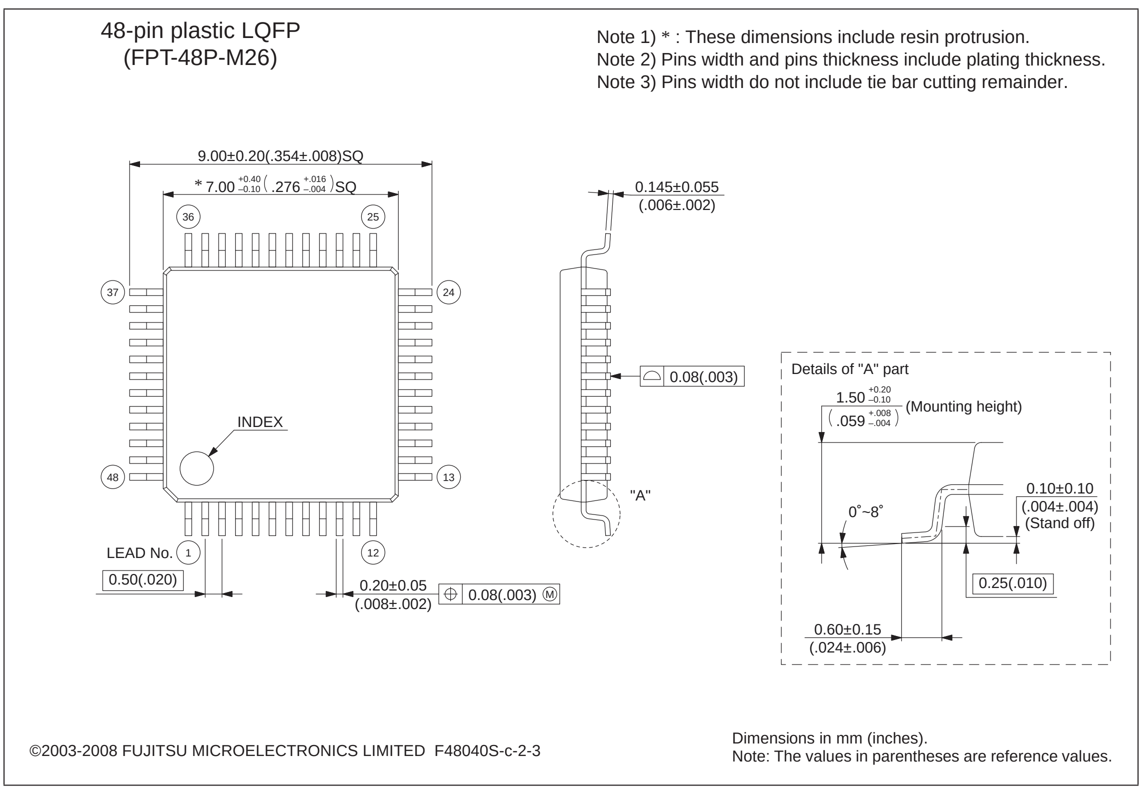
Figure 1.6-1 Package Dimension of FPT-48P-M26

48-pin plastic LQFP



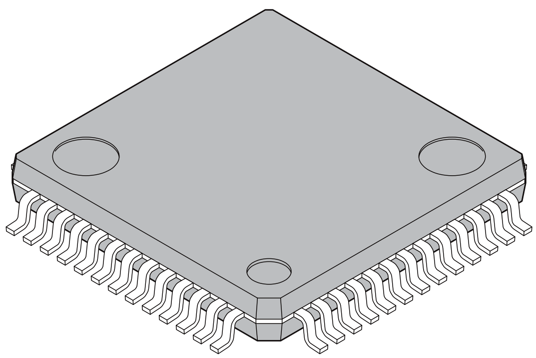
(FPT-48P-M26)

Lead pitch	0.50 mm
Package width × package length	7 × 7 mm
Lead shape	Gullwing
Sealing method	Plastic mold
Mounting height	1.70 mm MAX
Weight	0.17 g
Code (Reference)	P-LFQFP48-7×7-0.50



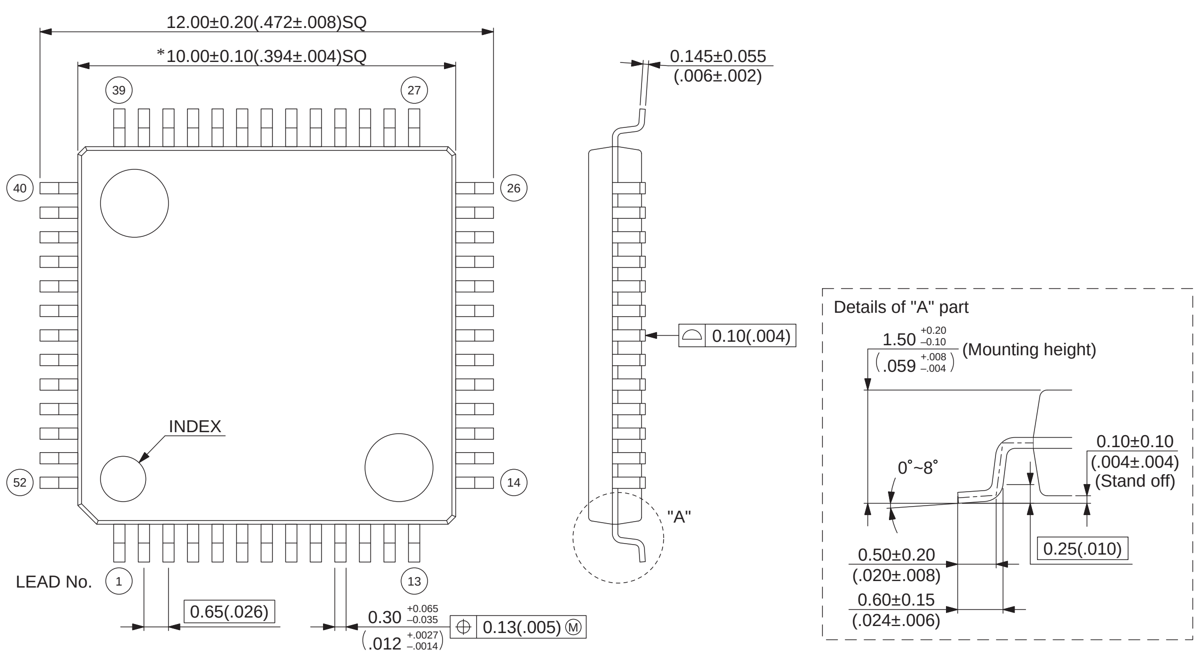
Please confirm the latest Package dimension by following URL.
http://edevice.fujitsu.com/package/en-search/

Figure 1.6-2 Package Dimension of FPT-52P-M01

52-pin plastic LQFP	Lead pitch	0.65 mm
	Package width × package length	10.0 × 10.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm Max
	Code (Reference)	P-LQFP52-10×10-0.65
(FPT-52P-M01)		

52-pin plastic LQFP
(FPT-52P-M01)

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



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

Please confirm the latest Package dimension by following URL.
http://edevice.fujitsu.com/package/en-search/

MB95150/M Series

1.7 Pin Description

Table 1.7-1 shows pin description. The alphabet in the "I/O Circuit Type" column of Table 1.7-1 corresponds to the one in the "Type" column of Table 1.8-1.

■ Pin Description

Table 1.7-1 Pin Description (1 / 3)

Pin no.		Pin name	I/O circuit type*3	Function
LQFP*1	LQFP*2			
1	1	P61/SEG09/PPG11	M	General-purpose I/O port. The pin is shared with LCDC SEG output (SEG09) and 8/16-bit PPG ch.1 output (PPG11).
2	2	P62/SEG10/TO10		General-purpose I/O port. The pins are shared with LCDC SEG output (SEG10, SEG11) and 8/16-bit compound timer ch.1 output (TO10, TO11).
3	3	P63/SEG11/TO11		
4	4	P64/SEG12/EC1		General-purpose I/O port. The pin is shared with LCDC SEG output (SEG12) and 8/16-bit compound timer ch.1 clock input (EC1).
5	5	P65/SEG13/SCK		General-purpose I/O port. The pin is shared with LCDC SEG output (SEG13) and LIN-UART clock I/O (SCK).
6	6	P66/SEG14/SOT		General-purpose I/O port. The pin is shared with LCDC SEG output (SEG14) and LIN-UART data output (SOT).
7	8	P67/SEG15/SIN	N	General-purpose I/O port. The pin is shared with LCDC SEG output (SEG15) and LIN-UART data input (SIN).
8	9	P14/PPG0	H	General-purpose I/O port. The pin is shared with 16-bit PPG ch.0 output.
9	10	P13/TRG0/ADTG		General-purpose I/O port. The pin is shared with 16-bit PPG ch.0 trigger input (TRG0) and A/D trigger input (ADTG).
10	11	P12/UCK0		General-purpose I/O port. The pin is shared with UART/SIO ch.0 clock I/O.
11	12	P11/UO0		General-purpose I/O port. The pin is shared with UART/SIO ch.0 data output.
12	13	P10/UI0	G	General-purpose I/O port. The pin is shared with UART/SIO ch.0 data input.

Table 1.7-1 Pin Description (2 / 3)

Pin no.		Pin name	I/O circuit type*3	Function
LQFP*1	LQFP*2			
13	14	P07/INT07/AN07	D	General-purpose I/O port. The pins are shared with external interrupt input (INT00 to INT07) and A/D converter analog input (AN00 to AN07).
14	15	P06/INT06/AN06		
15	16	P05/INT05/AN05		
16	17	P04/INT04/AN04		
17	18	P03/INT03/AN03		
18	19	P02/INT02/AN02		
19	21	P01/INT01/AN01		
20	22	P00/INT00/AN00		
21	23	MOD	B	The operating mode designation pin
22	24	X0	A	Main clock oscillation pin
23	25	X1		
24	26	V_{SS}	—	Power supply pin (GND)
25	27	V_{CC}	—	Power supply pin
26	28	C/PG0	H	General-purpose I/O port (at 3V). Capacitor connection pin (at 5V).
27	29	X1A	A	Sub clock oscillation pins (32 kHz)
28	30	X0A		
29	31	$\overline{RST}$	B’	Reset pin
30	32	P90/V3	R	General-purpose I/O port. The pins are shared with power supply pin for LCDC drive.
31	34	P91/V2		
32	35	P92/V1		
33	36	P93/V0		
34	37	P94	S	General-purpose I/O port.
35	38	P95		
36	39	PA0/COM0	M	General-purpose I/O port. The pins are shared with LCDC COM output.
37	40	PA1/COM1		
38	41	PA2/COM2		
39	42	PA3/COM3		

MB95150/M Series

Table 1.7-1 Pin Description (3 / 3)

Pin no.		Pin name	I/O circuit type*3	Function
LQFP*1	LQFP*2			
40	43	PB0/SEG00	M	General-purpose I/O port. The pins are shared with LCDC SEG output.
41	44	PB1/SEG01		
42	45	PB2/SEG02		
43	47	PB3/SEG03/PPG00		General-purpose I/O port. The pins are shared with LCDC SEG output (SEG03, SEG04) and 8/16-bit PPG ch.0 output (PPG00, PPG01).
44	48	PB4/SEG04/PPG01		
45	49	PB5/SEG05/TO00		General-purpose I/O port. The pins are shared with LCDC SEG output (SEG05, SEG06) and 8/16-bit compound timer ch.0 output (TO00, TO01).
46	50	PB6/SEG06/TO01		
47	51	PB7/SEG07/EC0		General-purpose I/O port. The pin is shared with LCDC SEG output (SEG07) and 8/16-bit compound timer ch.0 clock input (EC0).
48	52	P60/SEG08/PPG10	M	General-purpose I/O port. The pin is shared with LCDC SEG output (SEG08) and 8/16-bit PPG ch.1 output (PPG10).
—	7, 20, 33, 46	NC	—	Internal connect pin. Be sure this pin is left open.

*1 : FPT-48P-M26

*2 : FPT-52P-M01

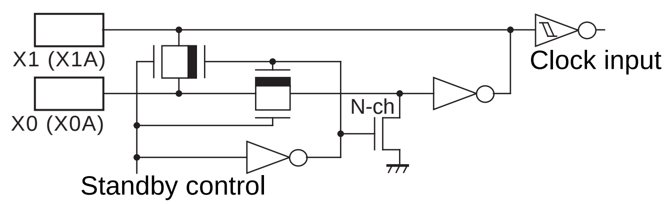
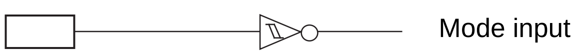
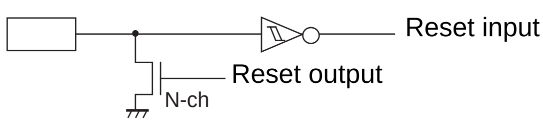
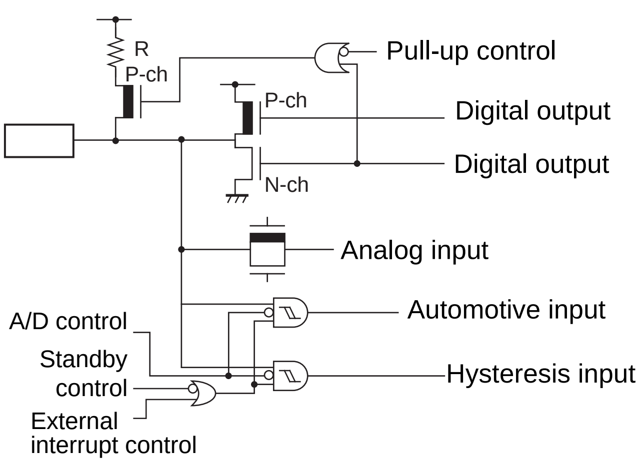
*3 : For the I/O circuit type, refer to "1.8 I/O Circuit Type".

1.8 I/O Circuit Type

Table 1.8-1 lists the I/O circuit types. Also, the alphabet in the "Type" column of Table 1.8-1 corresponds to the one in the "I/O circuit type" column of Table 1.7-1.

■ I/O Circuit Type

Table 1.8-1 I/O Circuit Type (1 / 3)

Type	Circuit	Remarks
A		• Oscillation circuit • High-speed side Feedback resistance : approx. 1 MΩ • Low-speed side Feedback resistance : approx. 10 MΩ
B		• Only for input • Hysteresis input
B’		• Hysteresis input • Reset output
D		• CMOS output • Hysteresis input • Analog input • With pull-up control • Automotive input

MB95150/M Series

Table 1.8-1 I/O Circuit Type (2 / 3)

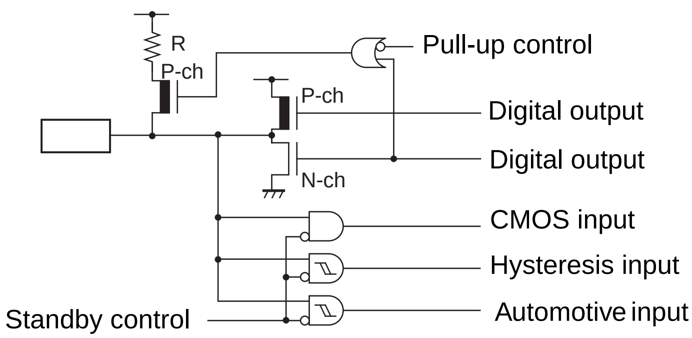
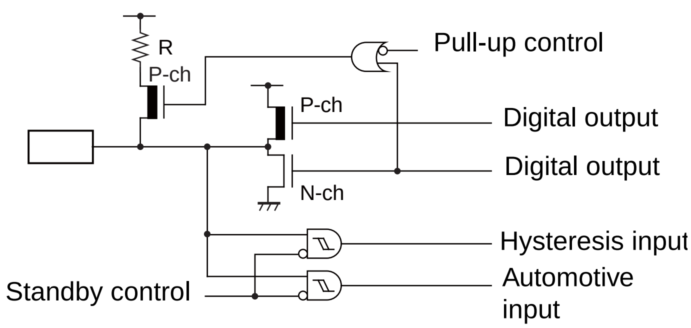
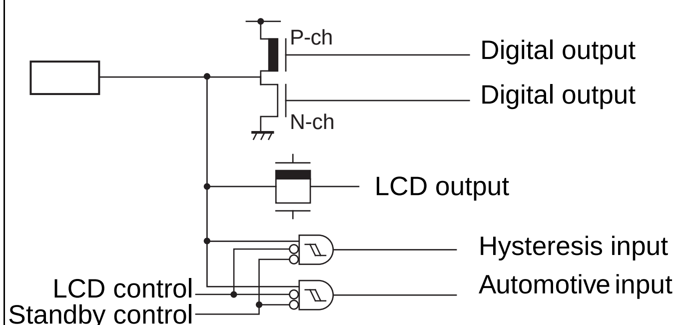
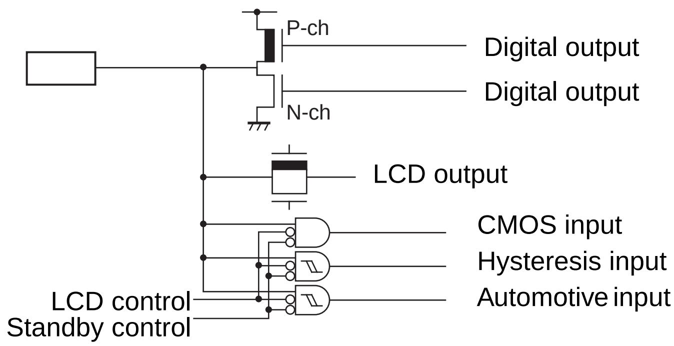
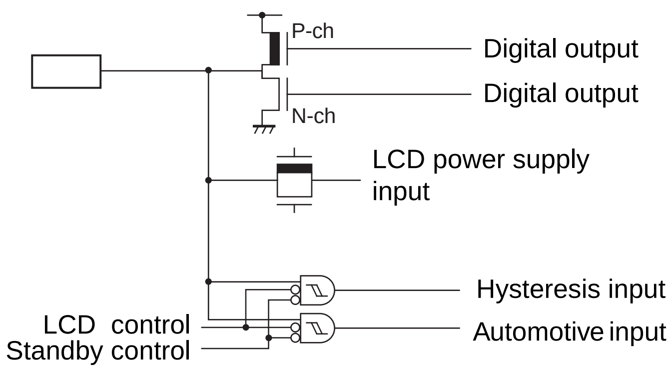
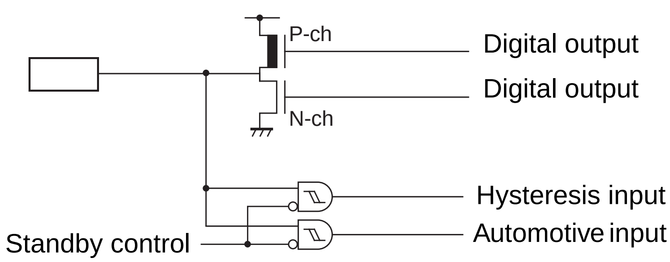
Type	Circuit	Remarks
G		• CMOS output • CMOS input • Hysteresis input • With pull-up control • Automotive input
H		• CMOS output • Hysteresis input • With pull-up control • Automotive input
M		• CMOS output • LCD output • Hysteresis input • Automotive input

Table 1.8-1 I/O Circuit Type (3 / 3)

<table>
<tr><th>Type</th><th>Circuit</th><th>Remarks</th></tr>
<tr><td>N</td><td>
</td><td>• CMOS output
• LCD output
• CMOS input
• Hysteresis input
• Automotive input</td></tr>
<tr><td>R</td><td>
</td><td>• CMOS output
• Hysteresis input
• Automotive input
• LCD power supply</td></tr>
<tr><td>S</td><td>
</td><td>• CMOS output
• Hysteresis input
• Automotive input</td></tr>
</table>

CHAPTER 2
HANDLING DEVICES

This chapter gives notes on using this series.

Code: CM26-00101-3E
Page: 23, 24

2.1 Device Handling Precautions

This section describes the precautions common to all devices including the device's power supply voltage and pin treatment.
Note that available functions differ depending on the series.

■ Device Handling Precautions

● Preventing Latch-up

Care must be taken to ensure that maximum voltage ratings are not exceeded when they are used.

Latch-up may occur on CMOS ICs if voltage higher than Vcc or lower than Vss is applied to input and output pins other than medium- and high-withstand voltage pins or if higher than the rating voltage is applied between Vcc pin and Vss pin.

When latch-up occurs, power supply current increases rapidly and might thermally damage elements.

Also, take care to prevent the analog power supply voltage (AVcc) and analog input voltage from exceeding the digital power supply voltage (Vcc) when the analog system power supply is turned on or off.

● Stable Supply Voltage

Supply voltage should be stabilized.

A sudden change in power-supply voltage may cause a malfunction even within the guaranteed operating range of the Vcc power-supply voltage.

For stabilization, in principle, keep the variation in Vcc ripple (p-p value) in a commercial frequency range (50 Hz/60 Hz) not to exceed 10% of the standard Vcc value and suppress the voltage variation so that the transient variation rate does not exceed 0.1 V/ms during a momentary change such as when the power supply is switched.

● Precautions for Use of External Clock

Even when an external clock is used, oscillation stabilization wait time is required for power-on reset, wake-up from sub clock mode or stop mode.

● Serial Communication

There is a possibility to receive wrong data due to noise or other causes on the serial communication. Therefore, design a printed circuit board so as to avoid noise. Consider receiving of wrong data, for example, apply a checksum of data at the end to detect an error. If an error is detected, retransmit the data.

■ Precautions for Debug

When using an evaluation device (mounted on an MCU board) for software development, there may be some differences between the operation of the evaluation device and the device you will actually use. The following lists some points to note during development.

● SYCC Register Settings

During debugging, the values of the DIV1 and DIV0 bits in the SYCC register may differ from the user settings. This is because, when a break occurs, the CPU adjusts the communications speed between the evaluation device and the BGM adapter to use the optimum speed.

To prevent this from occurring, you need to set response speed optimization to disabled.

For this information, refer also to "2.3.1 Setting Operating Environment" in "F^2MC-8L/8FX Family SOFTUNE Workbench USER'S MANUAL".

● Flash Memory Types and Sizes

Each evaluation device can be used for debugging of a number of different production models (series). When developing your program, please take note of the actual ROM and RAM sizes on the device you intend to use.

Further, evaluation devices use dual-operation flash memory. However, some production models have flash memory containing only one sector. Please take note of any differences between the flash memory configurations of the production and evaluation devices, particularly if writing a program that performs self-updating of flash memory.

● Differences in Flash Memory Content

The debugger for the F^2MC-8FX family uses the software break instruction to implement break points. When continuous or step execution is performed after setting a break point, the software break instruction is written to the break address in the flash memory on the evaluation device.

Accordingly, the contents of flash memory after a software break has been inserted by the debugger will be different to the program data image generated by the compiler. Before performing a checksum, you must remember to clear all break points and "synchronize flash memory".

● Restrictions Relating to the Flash Memory on the Evaluation Device

The following restrictions apply to the evaluation device for the F^2MC-8FX family.

(1) Writing or erasing the lower bank (addresses 1000_H to $3FFF_H$) is not possible.
When debugging, please do this on the production flash memory model.

(2) Do not use the chip erase command for the flash memory on the evaluation device. When debugging, please do this on the production flash memory model.

● Operation of Peripheral Functions During a Break

When a CPU break occurs, the debugger for the F^2MC-8FX family halts CPU operation (instruction code fetch, decoding, instruction execution, updating the PC, etc.) but the peripheral functions (PPG timer, UART, A/D converter, etc.) continue to operate.

The following are some example implications:

(1) If the overflow flag for a timer/counter is set during a CPU break and the interrupt is enabled, the interrupt routine will run immediately when execution restarts after the break.

(2) Clearing the overflow flag for a timer/counter via the memory window or similar during a CPU break will not appear to work as the flag will quickly be reset again.

● Prohibited Access to Undefined I/O Addresses

The debugger for the F^2MC-8FX family uses the same evaluation device for debugging all models. This evaluation device includes all peripheral functions that may be used during debugging. Accessing a register that does not exist on your target production device may invoke a peripheral function that should not exist and may result in abnormal operation. Accordingly, please do not access undefined address areas.

MB95150/M Series

■ Pin Connection

● Treatment of Unused Pin

Leaving unused input pins unconnected can cause abnormal operation or latch-up, leaving to permanent damage. Unused input pins should always be pulled up or down through resistance of at least 2 kΩ.

Any unused input/output pins may be set to output mode and left open, or set to input mode and treated the same as unused input pins. If there is unused output pin, make it open.

● Power Supply Pins

In products with multiple V_{CC} or V_{SS} pins, the pins of the same potential are internally connected in the device to avoid abnormal operations including latch-up. However, you must connect the pins to external power supply and a ground line to lower the electro-magnetic emission level, to prevent abnormal operation of strobe signals caused by the rise in the ground level, and to conform to the total output current rating. Moreover, connect the current supply source with the V_{CC} and V_{SS} pins of this device at the low impedance.

It is also advisable to connect a ceramic bypass capacitor of approximately 0.1 µF between V_{CC} and V_{SS} pins near this device.

● Mode Pin (MOD)

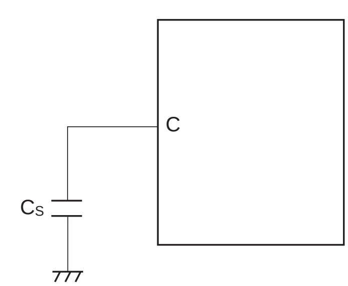
Connect the mode pin directly to V_{CC} or V_{SS}.

To prevent the device unintentionally entering test mode due to noise, lay out the printed circuit board so as to minimize the distance from the mode pins to V_{CC} or V_{SS} and to provide a low-impedance connection.

● C pin

Use a ceramic capacitor or a capacitor with equivalent frequency characteristics. A bypass capacitor of V_{CC} pin must have a capacitance value higher than C_S. For connection of smoothing capacitor C_S, see Figure 2.1-1.

Figure 2.1-1 C pin connection diagram



● NC Pins

Any pins marked "NC" must be left open.

CHAPTER 3
MEMORY SPACE

This chapter describes memory space.

3.1 Memory Space

The memory space on the F²MC-8FX family is 64 K bytes, divided into I/O, extended I/O, data, and program areas. The memory space includes special-purpose areas such as the general-purpose registers and vector table.

■ Configuration of Memory Space

● I/O area (addresses: 0000_H to $007F_H$)

- This area contains the control registers and data registers for on-chip peripheral resources.
- As the I/O area is allocated as part of memory space, it can be accessed in the same way as for memory. It can also be accessed at higher speed by using direct addressing instructions.

● Extended I/O area (addresses: $0F80_H$ to $0FFF_H$)

- This area contains the control registers and data registers for on-chip peripheral resources.
- As the extended I/O area is allocated as part of memory space, it can be accessed in the same way as for memory.

● Data area

- Static RAM is incorporated as the internal data area.
- The internal RAM capacity is different depending on the product.
- The area from 0080_H to $047F_H$ is an extended direct addressing area. It can be accessed at higher speed by direct addressing instructions with the direct bank pointer set (initial value: 0080_H - $00FF_H$).
- Addresses 0100_H to $01FF_H$ can be used as a general-purpose register area.

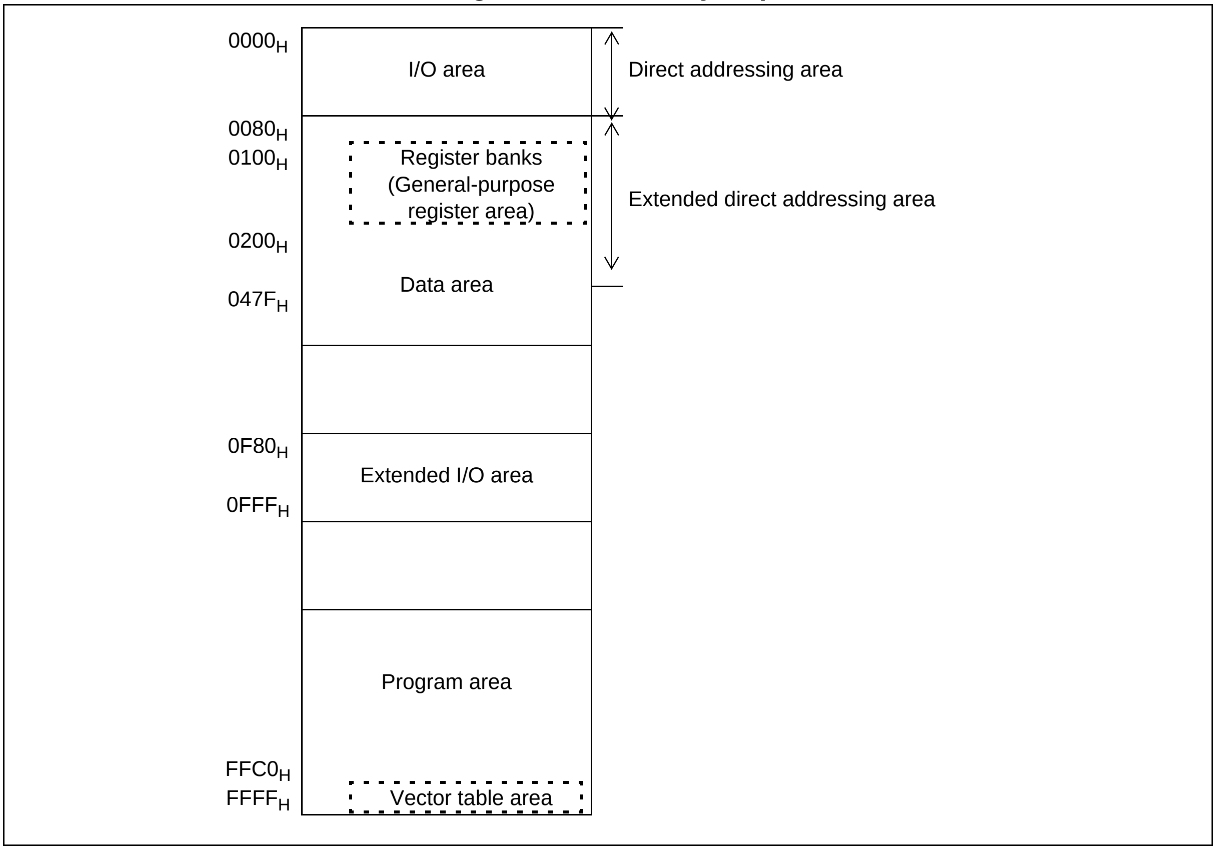
● Program area

- ROM is incorporated as the internal program area.
- The internal ROM capacity is different depending on the model.
- Addresses $FFC0_H$ to $FFFF_H$ are used as the vector table.

MB95150/M Series

■ Memory Map

Figure 3.1-1 Memory Map



3.1.1 Areas for Specific Applications

The general-purpose register area and vector table area are used for the specific applications.

■ General-purpose Register Area (Addresses: 0100_H to $01FF_H$)

- This area contains the auxiliary registers used for 8-bit arithmetic or transfer operations.
- As the area is allocated as part of the RAM area, it can also be used as ordinary RAM.
- When the area is used as general-purpose registers, general-purpose register addressing enables higher-speed access using short instructions.

For details, see Section "5.1.1 Register Bank Pointer (RP)" and Section "5.2 General-purpose Registers".

■ Vector Table Area (Addresses: $FFC0_H$ to $FFFF_H$)

- This area is used as the vector table for vector call instructions (CALLV), interrupts, and resets.
- The vector table area is allocated at the top of the ROM area. At the individual addresses in the vector table, the start addresses of their respective service routines are set as data.

Table 8.1-1 lists the vector table addresses to be referenced for vector call instructions, interrupts, and for resets.

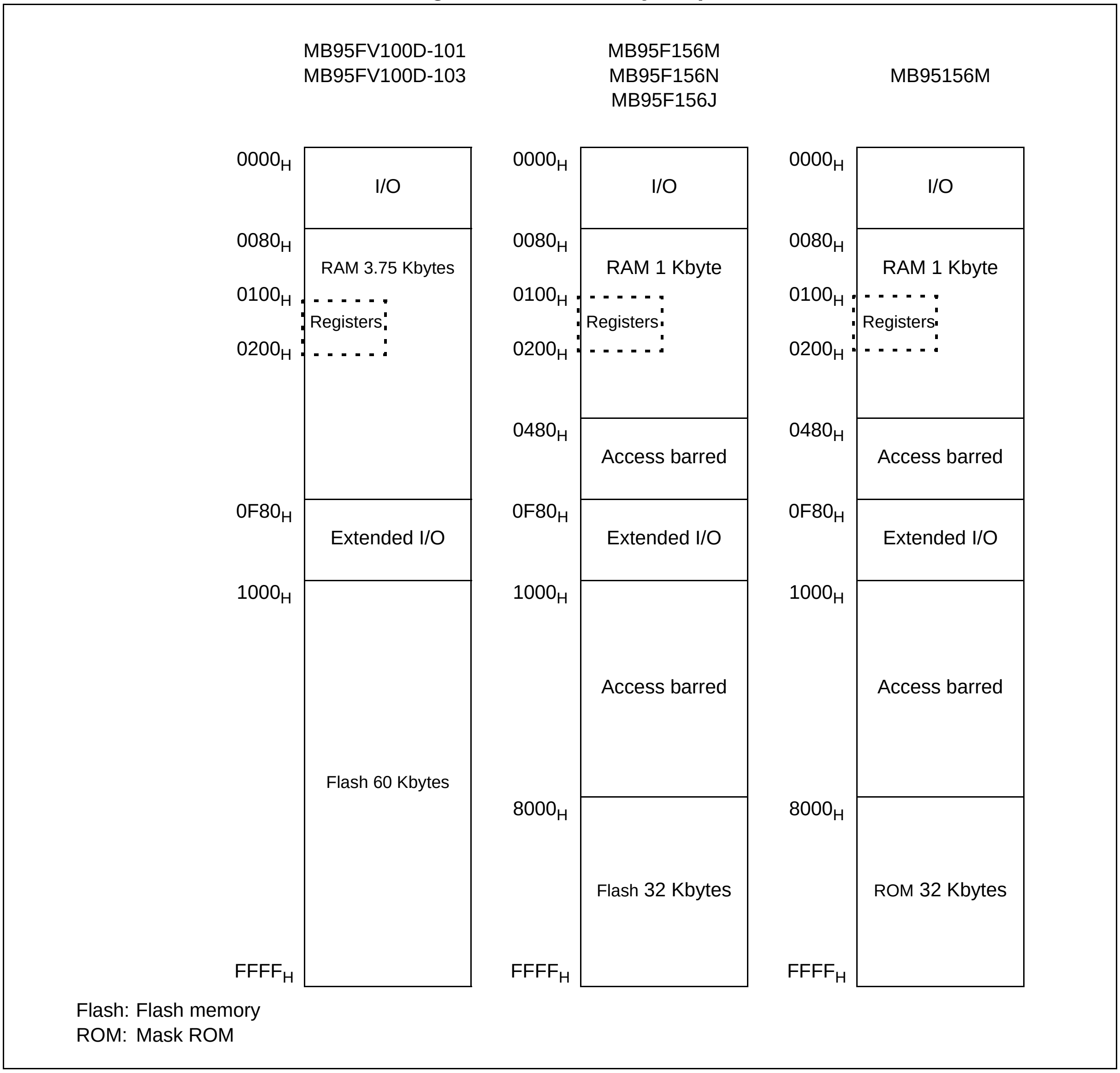
For details, see "CHAPTER 8 INTERRUPTS", "CHAPTER 7 RESET", and "■Special Instruction ●CALLV #vct" in Appendix "E.2 Special Instruction".

3.2 Memory Map

This section gives a memory map of this series.

■ Memory Map

Figure 3.2-1 Memory map



CHAPTER 4 MEMORY ACCESS MODE

This chapter describes the memory access mode.

Code: CM26-00102-1E

4.1 Memory Access Mode

The memory access mode supported by this series is only single-chip mode.

■ Single-chip Mode

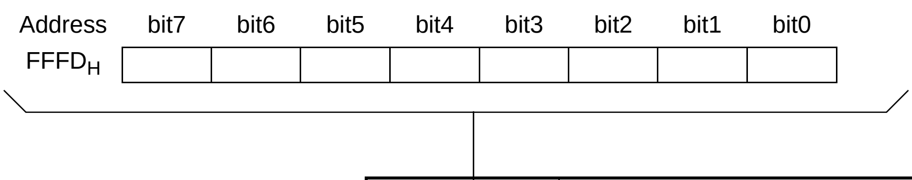
Single-chip mode uses only internal RAM and ROM. External bus access is not used.

● Mode data

Mode data is used to determine the memory access mode of the CPU.

The mode data address is fixed as $FFFD_H$ (the value of $FFFC_H$ can be any value). Be sure to set the mode data of internal ROM to "00_H" to select single-chip mode.

Figure 4.1-1 Mode Data Settings



Data	Operation
00_H	Select single-chip mode.
Other than 00_H	Reserved. Do not make any setting.

After a reset, the CPU fetches mode data first.

The CPU then fetches the reset vector after the mode data. The instruction is performed from the address set by reset vector.

● Mode pin (MOD)

Be sure to set the mode pin (MOD) to V_{SS}.

CHAPTER 5
CPU

This chapter describes functions and operations of the CPU.

5.1 Dedicated Registers

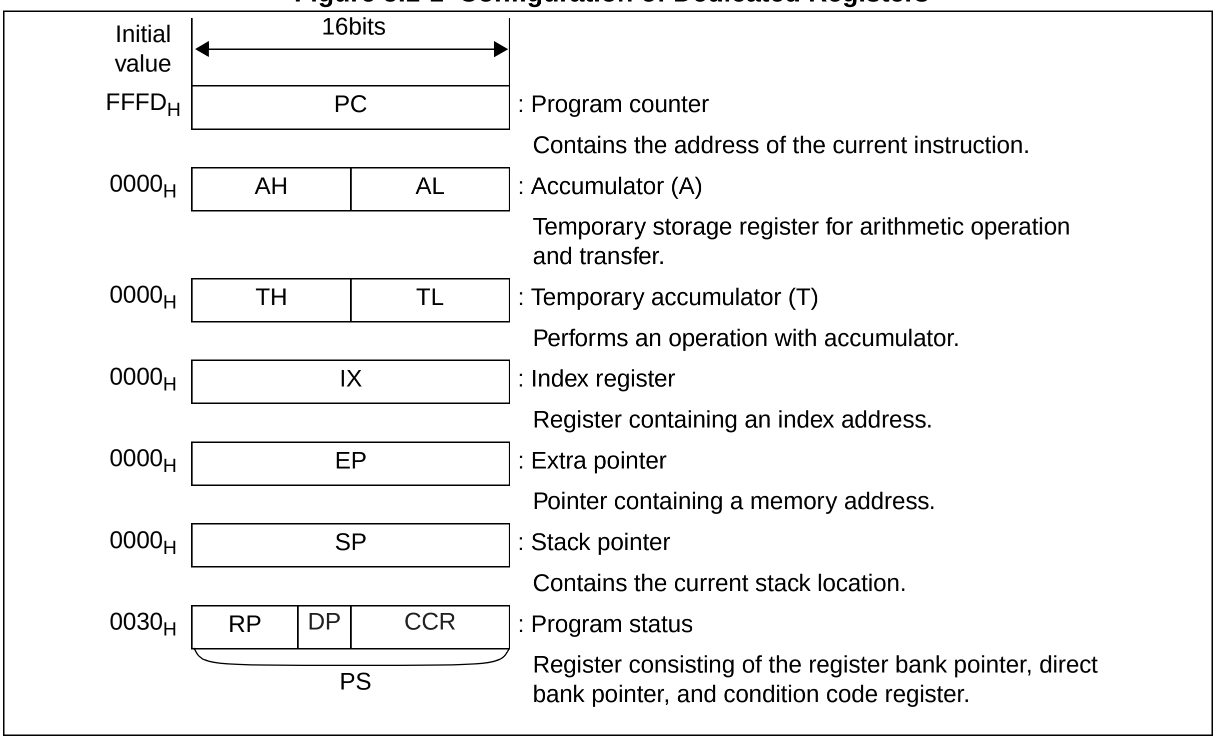
The CPU has its dedicated registers:the program counter (PC), two arithmetic registers (A and T), three address pointers (IX, EP, and SP), and the program status (PS) register. Each of the registers is 16 bits long.The PS register consists of the register bank pointer (RP), direct pointer (DP), and condition code register (CCR).

■ Configuration of Dedicated Registers

The dedicated registers in the CPU are seven 16-bit registers. Accumulator (A) and temporary accumulator (T) can also be used with only their lower eight bits in service.

Figure 5.1-1 shows the configuration of the dedicated registers.

Figure 5.1-1 Configuration of Dedicated Registers



■ Functions of Dedicated Registers

● Program counter (PC)

The program counter is a 16-bit counter which contains the memory address of the instruction currently executed by the CPU. The program counter is updated whenever an instruction is executed or an interrupt or reset occurs. The initial value set immediately after a reset is the mode data read address ($FFFD_H$).

● Accumulator (A)

The accumulator is a 16-bit register for arithmetic operation. It is used for a variety of arithmetic and transfer operations of data in memory or data in other registers such as the temporary accumulator (T). The data in the accumulator can be handled either as word (16-bit) data or byte (8-bit) data. For byte-length arithmetic and transfer operations, only the lower eight bits (AL) of the accumulator are used with the upper eight bits (AH) left unchanged. The initial value after a reset is "0000_H".

- Temporary accumulator (T)

The temporary accumulator is an auxiliary 16-bit register for arithmetic operation. It is used to perform arithmetic operations with the data in the accumulator (A). The data in the temporary accumulator is handled as word data for word-length (16-bit) operations with the accumulator (A) and as byte data for byte-length (8-bit) operations. For byte-length operations, only the lower eight bits (TL) of the temporary accumulator are used and the upper eight bits (TH) are not used.

When a MOV instruction is used to transfer data to the accumulator (A), the previous contents of the accumulator are automatically transferred to the temporary accumulator. When transferring byte-length data, the upper eight bits (TH) of the temporary accumulator remain unchanged. The initial value after a reset is "0000_H".

- Index register (IX)

The index register is a 16-bit register used to hold the index address. The index register is used with a single-byte offset (-128 to +127). The offset value is added to the index address to generate the memory address for data access. The initial value after a reset is "0000_H".

- Extra pointer (EP)

The extra pointer is a 16-bit register which contains the value indicating the memory address for data access. The initial value after a reset is "0000_H".

- Stack pointer (SP)

The stack pointer is a 16-bit register which holds the address referenced when an interrupt or subroutine call occurs and by the stack push and pop instructions. During program execution, the value of the stack pointer indicates the address of the most recent data pushed onto the stack. The initial value after a reset is "0000_H".

- Program status (PS)

The program status is a 16-bit control register. The upper eight bits make up the register bank pointer (RP) and direct bank pointer (DP); the lower eight bits make up the condition code register (CCR).

In the upper eight bits, the upper five bits make up the register bank pointer used to contain the address of the general-purpose register bank. The lower three bits make up the direct bank pointer which locates the area to be accessed at high speed by direct addressing.

The lower eight bits make up the condition code register (CCR) which consists of flags that represent the state of the CPU.

The instructions that can access the program status are MOVW A,PS or MOVW PS,A. The register bank pointer (RP) and direct bank pointer (DP) in the program status register can also be read from or written to by accessing the mirror address (0078_H).

Note that the condition code register (CCR) is part of the program status register and cannot be accessed independently.

Refer to the "F^2MC-8FX Programming Manual" for details on using the dedicated registers.

5.1.1 Register Bank Pointer (RP)

The register bank pointer (RP) in bits 15 to 11 of the program status (PS) register contains the address of the general-purpose register bank that is currently in use and is translated into a real address when general-purpose register addressing is used.

■ Configuration of Register Bank Pointer (RP)

Figure 5.1-2 shows the configuration of the register bank pointer.

Figure 5.1-2 Configuration of Register Bank Pointer

	RP					DP			CCR								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	RP Initial value
PS	R4	R3	R2	R1	R0	DP2	DP1	DP0	H	I	IL1	IL0	N	Z	V	C	00000_B

The register bank pointer contains the address of the register bank currently being used. The content of the register bank pointer is translated into a real address according to the rule shown in Figure 5.1-3.

Figure 5.1-3 Rule for Translation into Real Addresses in General-purpose Register Area

	Fixed value								RP: Upper					Op-code: Lower		
	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"1"	R4	R3	R2	R1	R0	b2	b1	b0
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Generated address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

The register bank pointer specifies the register bank used as general-purpose registers in the RAM area. There are a total of 32 register banks. The current register bank is specified by setting a value between 0 and 31 in the upper five bits of the register bank pointer. Each register bank has eight 8-bit general-purpose registers which are selected by the lower three bits of the op-code.

The register bank pointer allows the space from "0100_H" to up to "$01FF_H$" to be used as a general-purpose register area. Note, however, that the available area is limited depending on the product. The initial value after a reset is "0000_H".

■ Mirror Address for Register Bank and Direct Bank Pointers

The register bank pointer (RP) and direct bank pointer (DP) can be written to and read from by accessing the program status (PS) register using the "MOVW A,PS" and "MOVW PS,A" instructions, respectively. They can also be written to and read from directly by accessing mirror address "0078_H" of the register bank pointer.

5.1.2 Direct Bank Pointer (DP)

The direct bank pointer (DP) in bits 10 to 8 of the program status (PS) register specifies the area to be accessed by direct addressing.

■ Configuration of Direct Bank Pointer (DP)

Figure 5.1-4 shows the configuration of the direct bank pointer.

Figure 5.1-4 Configuration of Direct Bank Pointer

	RP					DP			CCR								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	DP Initial value
PS	R4	R3	R2	R1	R0	DP2	DP1	DP0	H	I	IL1	IL0	N	Z	V	C	000_B

The areas from 0000_H to $007F_H$ and 0080_H to $047F_H$ can be accessed by direct addressing. Access to 0000_H to $007F_H$ is specified with an operand regardless of the value in the direct bank pointer. Access to 0080_H to $047F_H$ is specified with the value in the value of the direct bank pointer and the operand.

Table 5.1-1 shows the relationship between direct bank pointer (DP) and access area; Table 5.1-2 lists the direct addressing instructions.

Table 5.1-1 Direct Access Pointer and Access Area

Direct bank pointer (DP) [2:0]	Operand-specified dir	Access area
XXX_B (It does not affect the mapping.)	0000_H to $007F_H$	0000_H to $007F_H$
000_B (Initial value)	0080_H to $00FF_H$	0080_H to $00FF_H$
001_B		0100_H to $017F_H$
010_B		0180_H to $01FF_H$
011_B		0200_H to $027F_H$
100_B		0280_H to $02FF_H$
101_B		0300_H to $037F_H$
110_B		0380_H to $03FF_H$
111_B		0400_H to $047F_H$

Table 5.1-2 Direct Address Instruction List

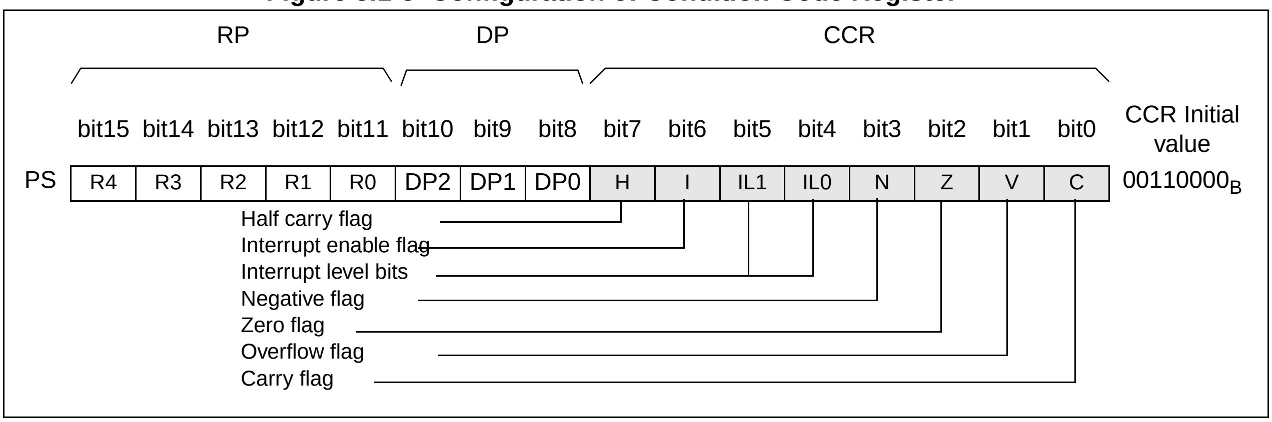
Applicable Instruction
CLRB dir:bit
SETB dir:bit
BBC dir:bit,rel
BBS dir:bit,rel
MOV A,dir
CMP A,dir
ADDC A,dir
SUBC A,dir
MOV dir,A
XOR A,dir
AND A,dir
OR A,dir
MOV dir,#imm
CMP dir,#imm
MOVW A,dir
MOVW dir,A

5.1.3 Condition Code Register (CCR)

The condition code register (CCR) in the lower eight bits of the program status (PS) register consists of the bits (H, N, Z, V, and C) containing information about the arithmetic result or transfer data and the bits (I, IL1, and IL0) used to control the acceptance of interrupt requests.

■ Configuration of Condition Code Register (CCR)

Figure 5.1-5 Configuration of Condition Code Register



The condition code register is a part of the program status (PS) register and therefore cannot be accessed independently.

■ Bits Result Information Bits

● Half carry flag (H)

This flag is set to "1" when a carry from bit3 to bit4 or a borrow from bit4 to bit3 occurs as the result of an operation. Otherwise, the flag is set to "0". Do not use this flag for any operation other than addition and subtraction as the flag is intended for decimal-adjusted instructions.

● Negative flag (N)

This flag is set to "1" when the value of the most significant bit is "1" as the result of an operation and set to "0" if the value is "0".

● Zero flag (Z)

This flag is set to "1" when the result of an operation is "0" and set to "0" otherwise.

● Overflow flag (V)

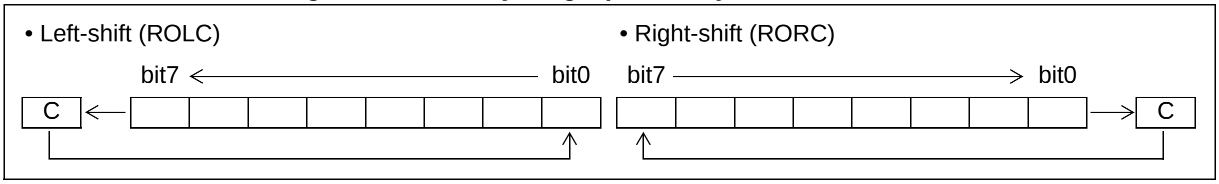
This flag indicates whether an operation has resulted in an overflow, assuming the operand used for the operation as an integer represented by a two's complement. The flag is set to "1" when an overflow occurs and set to "0" otherwise.

● Carry flag (C)

This flag is set to "1" when a carry from bit7 or a borrow to bit7 occurs as the result of an operation. Otherwise, the flag is set to "0". When a shift instruction is executed, the flag is set to the shift-out value.

Figure 5.1-6 shows how the carry flag is updated by a shift instruction.

Figure 5.1-6 Carry Flag Updated by Shift Instruction



■ Interrupt Acceptance Control Bits

● Interrupt enable flag (I)

When this flag is set to "1", interrupts are enabled and accepted by the CPU. When this flag is set to "0", interrupts are disabled and rejected by the CPU.

The initial value after a reset is "0".

The SETI and CLRI instructions set and clear the flag to "1" and "0", respectively.

● Interrupt level bits (IL1, IL0)

These bits indicate the level of the interrupt currently accepted by the CPU.

The interrupt level is compared with the value of the interrupt level setting register (ILR0 to ILR5) that corresponds to the interrupt request (IRQ0 to IRQ23) of each peripheral resource.

The CPU services an interrupt request only when its interrupt level is smaller than the value of these bits with the interrupt enable flag set (CCR: I = 1). Table 5.1-3 lists interrupt level priorities. The initial value after a reset is "11_B".

Table 5.1-3 Interrupt Levels

IL1	IL0	Interrupt Level	Priority
0	0	0	High
0	1	1	▲
1	0	2	▼
1	1	3	Low (No interrupt)

The interrupt level bits (IL1, IL0) are usually "11_B" with the CPU not servicing an interrupt (with the main program running).

For details on interrupts, see Section "8.1 Interrupts".

MB95150/M Series

5.2 General-purpose Registers

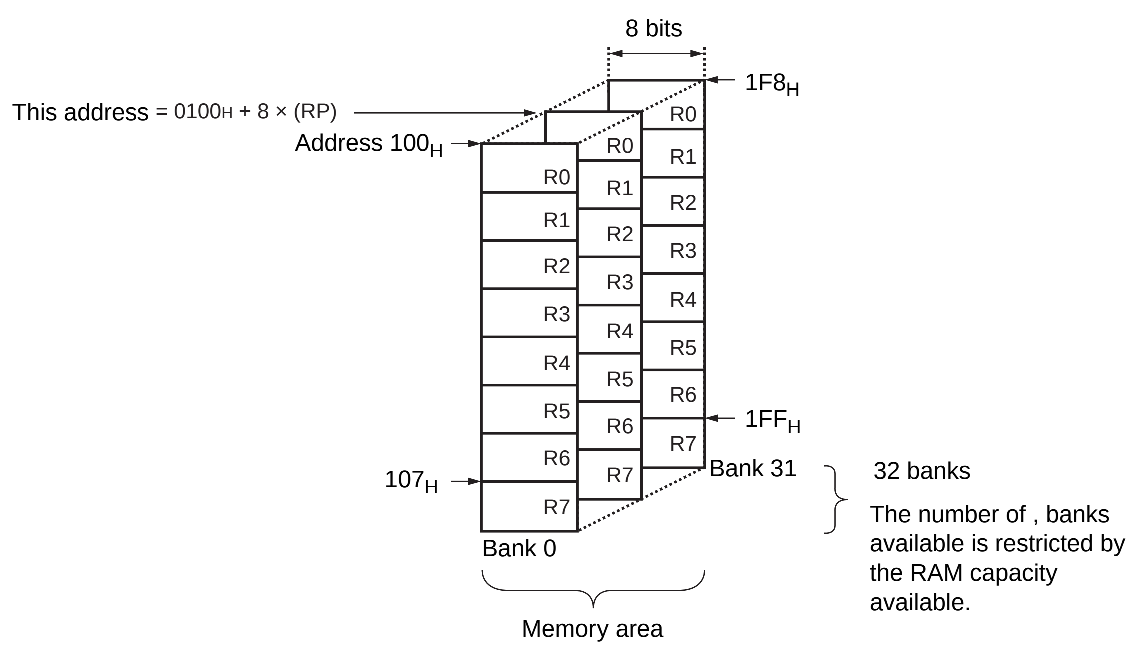
The general-purpose registers are memory blocks consisting of eight 8-bit registers per bank. A total of up to 32 register banks can be used. The register bank pointer (RP) is used to specify the register bank.
Register banks are useful for interrupt handling, vector call processing, and subroutine calls.

■ Configuration of General-purpose Registers

- The general-purpose registers are 8-bit registers and are located in register banks in the general-purpose register area (in RAM).
- Up to 32 banks can be used, where each bank consists of eight registers (R0 to R7).
- The register bank pointer (RP) specifies the register bank currently being used and the lower three bits of the op-code specify general-purpose register 0 (R0) to 7 (R7).

Figure 5.2-1 shows the configuration of the register banks.

Figure 5.2-1 Configuration of Register Banks



For information on the general-purpose register area available on each model, see Section "3.1.1 Areas for Specific Applications".

■ Features of General-purpose Registers

There are the following features in the general-purpose registers:

- High-speed access to RAM using short instructions (general-purpose register addressing).
- Blocks of register banks facilitating data backup and division by function unit.

General-purpose register banks can be allocated exclusively for specific interrupt service routines or vector call (CALLV #0 to #7) processing routines. An example is always using the fourth register bank for the second interrupt.

Only specifying a dedicated register bank at the beginning of an interrupt service routine automatically saves the general-purpose registers before the interrupt. This eliminates the need for pushing general-purpose register data onto the stack, allowing the CPU to accept interrupts at high speed.

Notes:

When coding an interrupt service routine, be careful not to change the value of the interrupt level bits (CCR: IL1, IL0) in the condition code register when specifying the register bank by updating the register bank pointer (RP) in that routine. Perform the programming by using either of them.

- Read the interrupt level bits and save their value before writing to the RP.
- Directly write to the RP mirror address "0078_H" to update the RP.

5.3 Placement of 16-bit Data in Memory

This section describes how 16-bit data is stored in memory.

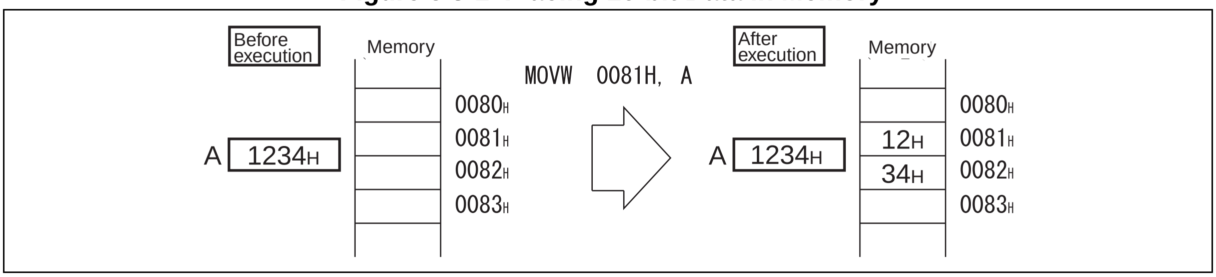
■ Placement of 16-bit Data in Memory

● State of 16-bit data stored in RAM

When you write 16-bit data to memory, the upper byte of the data is stored at a smaller address and the lower byte is stored at the next address. When you read 16-bit data, it is handled in the same way.

Figure 5.3-1 shows how 16-bit data is placed in memory.

Figure 5.3-1 Placing 16-bit Data in Memory



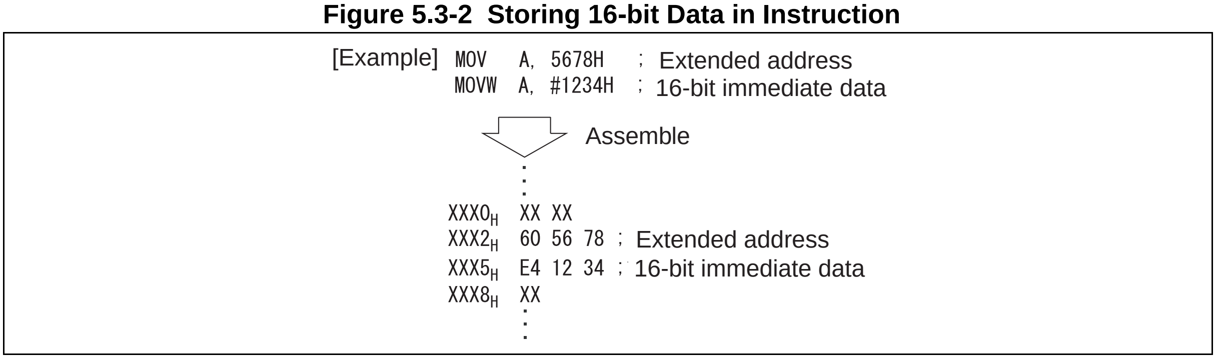
● State of operand-specified 16-bit data

In the same way, even when the operands in an instruction specifies 16-bit data, the upper byte is stored at the address closer to the op-code (instruction) and the lower byte is stored at the next address.

That is true whether the operands are either memory addresses or 16-bit immediate data.

Figure 5.3-2 shows how 16-bit data in an instruction is placed.

Figure 5.3-2 Storing 16-bit Data in Instruction



● State of 16-bit data in the stack

When 16-bit register data is pushed onto the stack upon an interrupt, the upper byte is stored at a lower address in the same way.

CHAPTER 6
CLOCK CONTROLLER

This chapter describes the functions and operations of the clock controller.

Code: CM26-00123-1E

Page: 78, 81, 82, 84

6.1 Overview of Clock Controller

The F^{2}MC-8FX family has a built-in clock controller that optimizes its power consumption. It includes dual clock product supporting both of the main clock and sub clock and single clock product supporting only the main clock.
The clock controller enables/disables clock oscillation, enables/disables clock supply to the internal circuitry, selects the clock source, and controls the PLL and frequency divider circuits.

■ Overview of Clock Controller

The clock controller enables/disables clock oscillation, enables/disables clock supply to the internal circuitry, selects the clock source, and controls the PLL and frequency divider circuits.

The clock controller controls the internal clock according to the clock mode, standby mode settings and the reset operation. The current clock mode selects the internal operating clock and the standby mode selects whether to enable or disable clock oscillation and signal supply.

The clock controller selects the optimum power consumption and features depending on the combination of clock mode and standby mode.

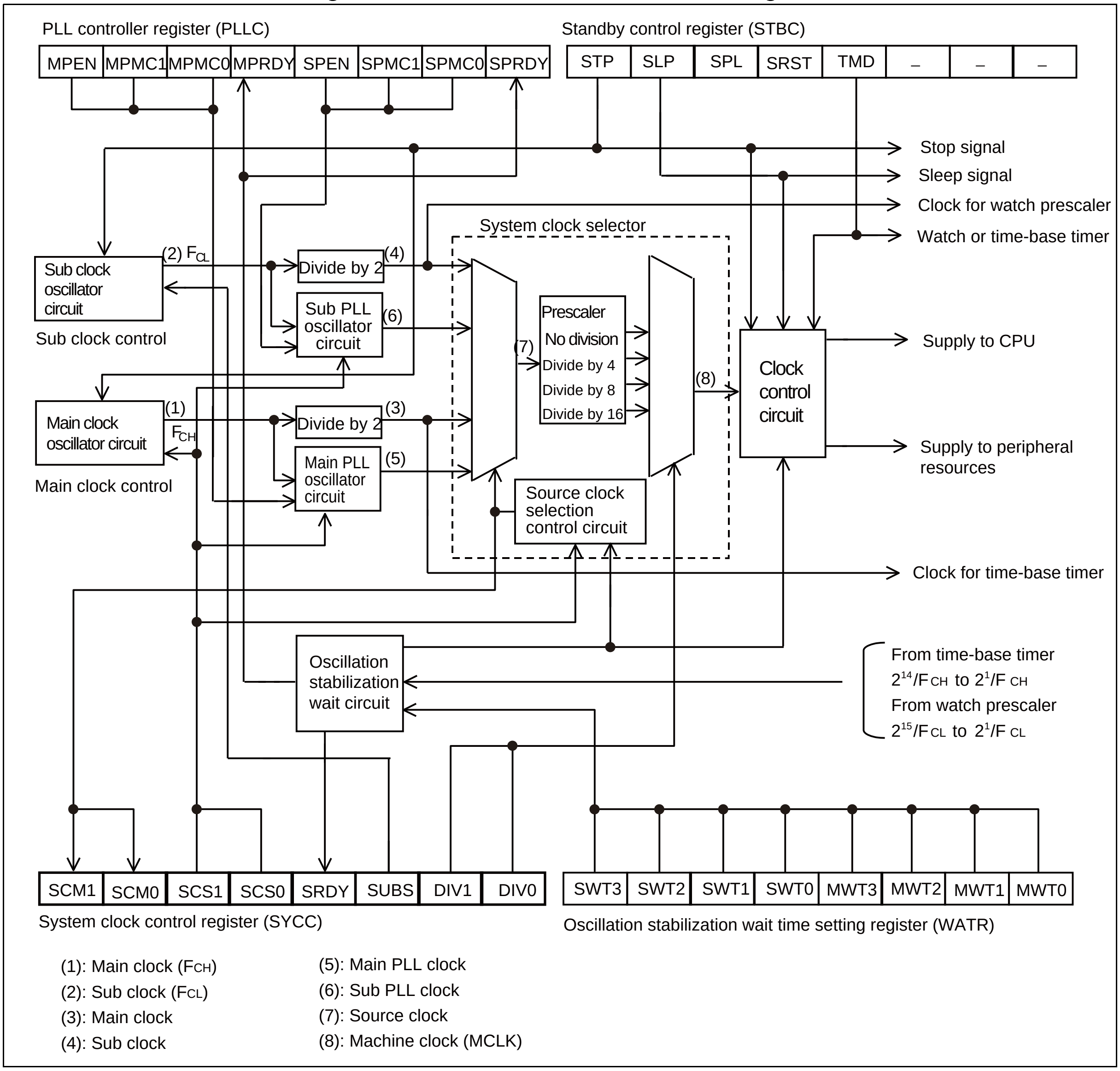
Dual clock product have four different source clocks: a main clock, which is the main oscillation clock divided by two, a sub clock, which is the sub oscillation clock divided by two, a main PLL clock, which is the main oscillation clock multiplied by the PLL multiplier and a sub PLL clock, which is the sub oscillation clock multiplied by the PLL multiplier.

Single clock product have two different source clocks: a main clock, which is the main oscillation clock divided by two; and a main PLL clock, which is the main oscillation clock multiplied by the PLL multiplier.

■ Block Diagram of the Clock Controller

Figure 6.1-1 shows the block diagram of the clock controller.

Figure 6.1-1 Clock Controller Block Diagram



The clock controller consists of the following blocks:

- Main clock oscillator circuit

This block is the oscillator circuit for the main clock.

- Sub clock oscillator circuit (Dual clock product)

This block is the oscillator circuit for the sub clock.

- Main PLL oscillator circuit

This block is the oscillator circuit for the main PLL.

- Sub PLL oscillator circuit (Dual clock product)

This block is the oscillator circuit for the sub PLL clock.

- System clock selector

This block selects one of the four different source clocks for main clock, sub clock, main PLL clock, and sub PLL clock depending on the clock mode. The prescaler frequency-divides the selected source clock into the machine clock. It is supplied to the clock control circuit.

- Clock control circuit

This block controls the supply of the machine clock to the CPU and each peripheral resource according to the standby mode or oscillation stabilization wait time.

- Oscillation stabilization wait circuit

This block outputs the oscillation stabilization wait time signal for each clock from 14 types of main clock oscillation stabilization signals created by the time-base timer and 15 types of sub clock oscillation stabilization signals created by the watch prescaler.

- System clock control register (SYCC)

This register is used to control current clock mode display, clock mode selection, machine clock divide ratio selection, and sub clock oscillation in main clock mode and main PLL clock mode.

- Standby control register (STBC)

This register is used to control the transition from RUN state to standby mode, the setting of pin states in stop mode, time-base timer mode, or watch mode, and the generation of software resets.

- PLL control register (PLLC)

This register is used to enable/disable the oscillation of the main PLL and sub PLL clocks, set the multiplier, and to indicate the stability of PLL oscillation.

- Oscillation stabilization wait time setting register (WATR)

This register is used to set the oscillation stabilization wait time for the main clock and sub clock.

■ Clock Mode

There are four clock modes available: main clock mode, main PLL clock mode, sub clock mode, and sub PLL clock mode.

Table 6.1-1 shows the relationships between the clock modes and the machine clock (operating clock for the CPU and peripheral resources).

Table 6.1-1 Clock Modes and Machine Clock Selection

Clock Mode	Machine Clock
Main clock mode	The machine clock is generated from the main clock (main clock divided by 2).
Main PLL clock mode	The machine clock is generated from the main PLL clock (main clock multiplied by the PLL multiplier).
Sub clock mode (Dual clock product only)	The machine clock is generated from the sub clock (sub clock divided by 2).
Sub PLL clock mode (Dual clock product only)	The machine clock is generated from the sub PLL clock (sub clock multiplied by the PLL multiplier).

In any of the clock modes, the selected clock can also be frequency-divided. Additionally, in modes using a PLL clock, a multiplier for the clock frequency can also be set.

■ Peripheral Resources Not Affected by Clock Mode

Note that the peripheral resources listed in the table below are not affected by the clock mode, division, and PLL multiplier settings. Table 6.1-2 lists the peripheral resources not affected by the clock mode.

Table 6.1-2 Peripheral Resources Not Affected by Clock Mode

Peripheral Function	Operating Clock
Time-base timer	Main clock ($2^1/F_{CH}$: main clock divided by 2)
Watchdog timer	Main clock (with time-base timer output selected) Sub clock (with watch prescaler output selected) (dual clock product only)
Watch prescaler (Dual clock product only)	Sub clock ($2^1/F_{CL}$: sub clock divided by 2)
Watch counter (Dual clock product only)	Sub clock (watch prescaler output)

For some peripheral resources other than those listed above, it may be possible to select the time-base timer or watch prescaler output as a count clock. Check the description of each peripheral resource for details.

■ Standby Modes

The clock controller selects whether to enable or disable clock oscillation and clock supply to internal circuitry depending on each standby mode. With the exception of time-base timer mode and watch mode, the standby mode can be set independently of the clock mode.

Table 6.1-3 shows the relationships between standby modes and clock supply states.

Table 6.1-3 Standby Modes and Clock Supply States

Standby Mode	Clock Supply States
Sleep mode	Stops clock supply to the CPU and watchdog timer. As a result, the CPU stops operation, but other peripheral resources continue operating.
Time-base timer mode	Supplies clock signals only to the time-base timer, watch prescaler, and watch counter while stopping clock supply to other circuits. As a result, all the functions other than the time-base timer, watch prescaler, watch counter, external interrupt, and low-voltage detection reset (option) are stopped. Time-base timer mode is only the standby mode for main clock mode or main PLL clock mode.
Watch mode (Dual clock product only)	Stops main clock oscillation, but supplies clock signals only to the watch prescaler and watch counter while stopping clock supply to other circuits. As a result, all the functions other than the watch prescaler, watch counter, external interrupt, and low-voltage detection reset (option) are stopped. Watch mode is only the standby mode for sub clock mode or sub PLL clock mode.
Stop mode	Stops main clock oscillation and sub clock oscillation and stops the supply of all clock signals. As a result, all the functions other than external interrupt and low-voltage detection reset (option) are stopped.

■ Combinations of Clock Mode and Standby Mode

Table 6.1-4 lists the combinations of clock mode and standby mode and their respective operating states of internal circuits.

Table 6.1-4 Combinations of Standby Mode and Clock Mode and Internal Operating States

Function	RUN				Sleep				Time-base TIMER		Watch (Dual clock product)		Stop	
	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product)	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product)	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product)	Main PLL clock mode	Sub PLL clock mode (Dual clock product)
Main clock	Operating		Stopped		Operating		Stopped		Operating		Stopped		Stopped	Stopped
Main PLL clock	Stopped *1	Operat-ing	Stopped		Stopped *1	Operat-ing	Stopped		Stopped*1		Stopped		Stopped	Stopped
Sub clock	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operat-ing*2	Stopped
Sub PLL clock	Stopped*3		Stopped *3	Operat-ing	Stopped*3		Stopped *3	Operat-ing	Stopped*3		Stopped *3	Operat-ing	Stopped*3	Stopped
CPU	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	Stopped
ROM RAM	Operating		Operating		Value held		Value held		Value held		Value held		Value held	Value held
I/O port	Operating		Operating		Output held		Output held		Output held/ Hi-Z		Output held/ Hi-Z		Output held/ Hi-Z	Output held/ Hi-Z
Time-base TIMER	Operating		Stopped		Operating		Stopped		Operating		Stopped		Stopped	Stopped
Watch Prescaler	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operat-ing*2	Stopped
Watch counter	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operat-ing*4	Stopped
External interrupt	Operating		Operating		Operating		Operating		Operating		Operating		Operat-ing	Operat-ing
Watchdog timer	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	Stopped
Low-voltage detection reset	Operating		Operating		Operating		Operating		Operating		Operating		Operat-ing	Operat-ing
Other peripheral functions	Operating		Operating		Operating		Operating		Stopped		Stopped		Stopped	Stopped

*1: Operates when the main PLL clock oscillation enable bit in the PLL control register (PLLC:MPEN) is set to "1".

*2: Stops when the sub clock oscillation stop bit in the system clock control register (SYCC:SUBS) is set to "1".

*3: Operates when the sub PLL clock oscillation enable bit in the PLL control register (PLLC:SPEN) is set to "1".

*4: Watch counter keeps counting and no interrupts occur. When the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) is set to "1", watch counter stops.

6.2 Oscillation Stabilization Wait Time

The oscillation stabilization wait time is the time after the oscillator circuit stops oscillation until the oscillator resumes its stable oscillation at its natural frequency. The clock controller obtains the oscillation stabilization wait time by counting a set number of oscillation clock cycles to prevent clock supply to internal circuits.

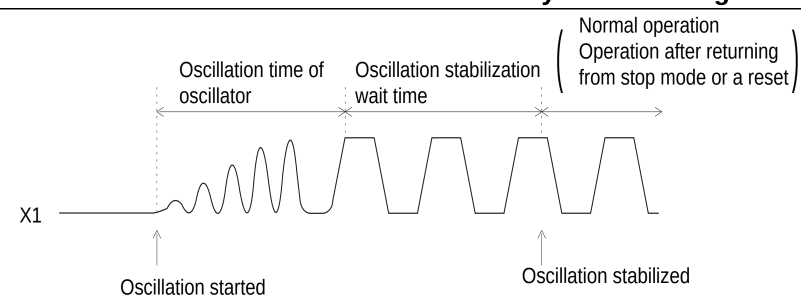
■ Oscillation Stabilization Wait Time

The clock controller obtains the oscillation stabilization wait time followed by the initiation of oscillation by counting a set number of oscillation clock cycles to prevent clock supply to internal circuits.

When a state transition request for starting oscillation when the power is turned on or for restarting halted oscillation at a clock mode change by a reset, an interrupt in standby mode, or by software, the clock controller automatically waits until the oscillation stabilization wait time for the main clock or sub clock has passed and then causes transition to the next state.

Figure 6.2-1 shows oscillation immediately after being started.

Figure 6.2-1 Behavior of Oscillator Immediately after Starting Oscillation



The main clock oscillation stabilization wait time is counted by using the time-base timer. The sub clock oscillation stabilization wait time is counted by using the watch prescaler. The count can be set in the oscillation stabilization wait time setting register (WATR). Set it in keeping with the oscillator characteristics.

When a power-on reset occurs, the oscillation stabilization wait time is fixed to the initial value. For masked ROM products, however, you can specify the initial value of the oscillation stabilization wait time when ordering masked ROM.

Table 6.2-1 shows the length of oscillation stabilization wait time.

Table 6.2-1 Oscillation Stabilization Wait Time

Clock	Factor	Oscillation stabilization wait time
Main clock	Power-on reset	Initial value: $(2^{14}-2)/F_{CH}$, where F_{CH} is the main clock frequency. (Specified when ROM is ordered for mask ROM products)
	Other than power-on reset	Register setting value (WATR:MWT3, MWT2, MWT1, MWT0)
Sub clock (Dual clock product)	Power-on reset	Initial value: $(2^{15}-2)/F_{CL}$, where F_{CL} is the sub clock frequency.
	Other than power-on reset	Register setting value (WATR:SWT3, SWT2, SWT1, SWT0)

After the oscillation stabilization wait time of the main clock ends, the oscillation stabilization wait time of sub clock measurement is begun.

■ PLL Clock Oscillation Stabilization Wait Time

As with the oscillation stabilization wait time of the oscillator, the clock controller automatically waits for the PLL oscillation stabilization wait time to elapse after a request for state transition from PLL oscillation stopped state to oscillation start is generated via an interrupt in standby mode or a change of clock mode by software.

Note that the PLL clock oscillation stabilization wait time changes according to the PLL startup timing.

Table 6.2-2 shows the PLL oscillation stabilization wait time.

Table 6.2-2 PLL Oscillation Stabilization Wait Time

	PLL Oscillation Stabilization Wait Time		Remarks
	Minimum time	Maximum time	
Main PLL clock	$2^{11}/F_{CH} \times 2$	$2^{11}/F_{CH} \times 3$	• Oscillation stabilization wait time is taken while $2^{11}/F_{CH}$ is counted twice (minimum) or three times (maximum). • F_{CH} represents the main clock frequency.
Sub PLL clock (Dual clock product)	$2^{8}/F_{CL} \times 2$	$2^{8}/F_{CL} \times 3$	• Oscillation stabilization wait time is taken while $2^{8}/F_{CL}$ is counted twice (minimum) or three times (maximum). • F_{CL} represents the sub clock frequency.

■ Oscillation Stabilization Wait Time and Clock Mode/Standby Mode Transition

The clock controller automatically waits for the oscillation stabilization wait time to elapse as needed when the operating state causes a transition. Depending on the state transition, however, the clock controller does not always wait for the oscillation stabilization wait time.

For details on state transitions, see Sections "6.7 Clock Mode" and "6.8 Operations in Low-power Consumption Modes (Standby Modes)".

6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) is used to indicate and switch the current clock mode, select the machine clock divide ratio, and control sub clock oscillation in main clock mode and main PLL clock mode.

■ Configuration of System Clock Control Register (SYCC)

Figure 6.3-1 Configuration of System Clock Control Register (SYCC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0007_H	SCM1	SCM0	SCS1	SCS0	SRDY	SUBS	DIV1	DIV0	$1010x011_B$
	R/WX	R/WX	R/W	R/W	R/WX	R/W	R/W	R/W	

DIV1	DIV0	Machine clock divide ratio selection bits
0	0	Source clock
0	1	Source clock / 4
1	0	Source clock / 8
1	1	Source clock / 16

SUBS	Sub clock oscillation stop bit
0	Starts sub clock oscillation
1	Stops sub clock oscillation

SRDY	Sub clock oscillation stability bit
0	Indicates the sub clock oscillation stabilization wait state or sub clock oscillation being stopped
1	Indicates sub clock oscillation being stable

SCS1	SCS0	Cock mode selection bits
0	0	Sub clock mode
0	1	Sub PLL clock mode
1	0	Main clock mode
1	1	Main PLL clock mode

SCM1	SCM0	Clock mode monitor bits
0	0	Sub clock mode
0	1	Sub PLL clock mode
1	0	Main clock mode
1	1	Main PLL clock mode

R/W : Readable/writable (Read value is the same as write value)
R/WX : Read only (Readable, writing has no effect on operation)
X : Indeterminate
(shaded) : Initial value

Table 6.3-1 Functions of Bits in System Clock Control Register (SYCC)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7,
bit6</td><td>SCM1, SCM0:
clock mode
monitor bit</td><td>Indicate the current clock mode.
When set to "00_B": the bits indicate sub clock mode.
When set to "01_B": the bits indicate sub PLL clock mode.
When set to "10_B": the bits indicate main clock mode.
When set to "11_B": the bits indicate main PLL clock mode.
These bits are read-only. Writing has no effect on operation.</td></tr>
<tr><td>bit5,
bit4</td><td>SCS1, SCS0:
clock mode
select bits</td><td>Specify the clock mode.
When set to "00_B": the bits specify transition to sub clock mode.
(Dual clock product only)
When set to "01_B": the bits specify transition to sub PLL clock mode.
(Dual clock product only)
When set to "10_B": the bits specify transition to main clock mode.
When set to "11_B": the bits specify transition to main PLL clock mode.
Once a clock mode has been selected in the SCS1 and SCS0 bits, any attempt to write to them is ignored until the transition to that clock mode is completed.
On single clock product, an attempt to write "00_B" or "01_B" to these bits is ignored, leaving their value unchanged.</td></tr>
<tr><td>bit3</td><td>SRDY:
Sub clock oscillation stability bit
(Dual clock product only)</td><td>Indicates whether sub clock oscillation has become stable.
• When set to "1", the SRDY bit indicates that the oscillation stabilization wait time for the sub clock has passed.
• When set to "0", the SRDY bit indicates that the clock controller is in the sub clock oscillation stabilization wait state or that sub clock oscillation has been stopped.
This bit is read-only. Writing has no effect on operation.
On single clock product, the value of these bits is meaningless.</td></tr>
<tr><td>bit2</td><td>SUBS:
Sub clock oscillation stop bit
(Dual clock product only)</td><td>Stops sub clock oscillation in main clock mode or main PLL clock mode.
When set to "0": the bit enables sub clock oscillation.
When set to "1": the bit stops sub clock oscillation.
Note:
• In sub clock mode or sub PLL clock mode, the sub clock oscillates regardless of the value of this bit, except in stop mode.
• In main clock mode or main PLL clock mode as well, the sub clock oscillates regardless of the value of this bit when sub PLL clock oscillation has been enabled by the PLL clock oscillation enable bit in the PLL control register (PLLC:SPEN).
• Do not update the SYCC: SCS1 bit and this bit at the same time.
• On single clock product, the value of the bit has no effect on the operation.</td></tr>
<tr><td>bit1,
bit0</td><td>DIV1, DIV0:
Machine Clock divide ratio select bits</td><td>• These bits select the machine clock divide ratio to the source clock.
• The machine clock is generated from the source clock according to the divide ratio set by the bits.
<table>
<tr><th>DIV1</th><th>DIV0</th><th>Machine Clock Divide Ratio Selection Bits</th><th>SCM1, SCM0 = 10_B</th></tr>
<tr><td>0</td><td>0</td><td>Source clock (No division)</td><td>Main clock divided by 2</td></tr>
<tr><td>0</td><td>1</td><td>Source clock/4</td><td>Main clock divided by 8</td></tr>
<tr><td>1</td><td>0</td><td>Source clock/8</td><td>Main clock divided by 16</td></tr>
<tr><td>1</td><td>1</td><td>Source clock/16</td><td>Main clock divided by 32</td></tr>
</table></td></tr>
</table>

6.4 PLL Control Register (PLLC)

The PLL control register (PLLC) controls the main PLL clock and sub PLL clock.

■ Configuration of PLL Control Register (PLLC)

Figure 6.4-1 Configuration of PLL Control Register (PLLC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0006_H	MPEN	MPMC1	MPMC0	MPRDY	SPEN	SPMC1	SPMC0	SPRDY	00000000_B
	R/W	R/W	R/W	R/WX	R/W	R/W	R/W	R/WX	

SPRDY	Sub PLL clock oscillation stability bit
0	Indicates the sub PLL clock oscillation stabilization wait state or sub PLL clock oscillation being stopped
1	Indicates sub PLL clock oscillation being stable

SPMC1	SPMC0	Sub PLL clock multiplier setting bits
0	0	Setting prohibited
0	1	Sub clock × 2
1	0	Sub clock × 3
1	1	Sub clock × 4

SPEN	Sub PLL clock oscillation enable bit
0	Disables sub PLL clock oscillation
1	Enables sub PLL clock oscillation

MPRDY	Main PLL clock oscillation stability bit
0	Indicates the main PLL clock oscillation stabilization wait state or main PLL clock oscillation being stopped
1	Indicates main PLL clock oscillation being stable

MPMC1	MPMC0	Main PLL clock multiplier setting bits
0	0	Main clock × 1
0	1	Main clock × 2
1	0	Main clock × 2.5
1	1	Main clock × 4

MPEN	Main PLL clock oscillation enable bit
0	Disables main PLL clock oscillation
1	Enables main PLL clock oscillation

R/W : Readable/writable (Read value is the same as write value)
R/WX : Read-only (Read-only. Writing does not affect the operation.)
(shaded) : Initial value

Table 6.4-1 Functions of Bits in PLL Control Register (PLLC) (1 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>MPEN:
Main PLL clock oscillation enable bit</td><td>Enables or disables the oscillation of the main PLL clock in main clock mode or time-base timer mode.
When set to "0": the bit disables main PLL clock oscillation.
When set to "1": the bit enables main PLL clock oscillation.
In main PLL clock mode, the main PLL clock oscillates regardless of the value of this bit either in the RUN state or in sleep mode.</td></tr>
<tr><td>bit6,
bit5</td><td>MPMC1, MPMC0:
Main PLL clock multiplier setting bits</td><td>Set the multiplier for the main PLL clock.
<table>
<tr><th>MPMC1</th><th>MPMC0</th><th>Main PLL clock multiplier setting bits</th></tr>
<tr><td>0</td><td>0</td><td>Main clock × 1</td></tr>
<tr><td>0</td><td>1</td><td>Main clock × 2</td></tr>
<tr><td>1</td><td>0</td><td>Main clock × 2.5</td></tr>
<tr><td>1</td><td>1</td><td>Main clock × 4</td></tr>
</table>
Note: The value of these bits can be changed only when the main PLL clock is stopped. Consequently, you should not update the bits either with the main PLL clock oscillation enable bit (MPEN) set to "1" or with the clock mode selection bits in the system clock control register (SYCC:SCS1, SCS0) set to "11_B". (It is however possible to set these bits at the same time as setting SPEN to "1".)</td></tr>
<tr><td>bit4</td><td>MPRDY:
Main PLL clock oscillation stability bit</td><td>Indicates whether main PLL clock oscillation has become stable.
• When set to "1", the MPRDY bit indicates that the oscillation stabilization wait time for the main PLL clock has passed.
• When set to "0", the MPRDY bit indicates that the clock controller is in the main PLL clock oscillation stabilization wait state or that main PLL clock oscillation has been stopped.
This bit is read-only. Any value attempted to be written is meaningless and has no effect on operation.</td></tr>
<tr><td>bit3</td><td>SPEN:
Sub PLL clock oscillation enable bit (Dual clock product only)</td><td>Enables or disables the oscillation of the sub PLL clock in main clock mode, main PLL clock mode, sub clock mode, or in watch mode.
When set to "0": the bit disables sub PLL clock oscillation.
When set to "1": the bit enables sub PLL clock oscillation.
In sub PLL clock mode, the sub PLL clock oscillates regardless of the value of this bit except in watch mode.
Even in sub PLL clock mode, the sub PLL clock stops oscillation in stop mode regardless of the value of this bit.
On single clock product, the value of the bit has no effect on the operation.</td></tr>
</table>

Table 6.4-1 Functions of Bits in PLL Control Register (PLLC) (2 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit2, bit1</td><td>SPMC1, SPMC0:
Sub PLL clock multiplier setting bits
(Dual clock product only)</td><td>Set the multiplier for the Sub PLL clock.
<table>
<tr><th>SPMC1</th><th>SPMC0</th><th>Sub PLL Clock Multiplier Setting Bits</th></tr>
<tr><td>0</td><td>0</td><td>Setting prohibited.Be sure to write any other value before using the PLL.</td></tr>
<tr><td>0</td><td>1</td><td>Sub clock × 2</td></tr>
<tr><td>1</td><td>0</td><td>Sub clock × 3</td></tr>
<tr><td>1</td><td>1</td><td>Sub clock × 4</td></tr>
</table>
On single clock product, the value of the bit has no effect on the operation.
Notes:
• Although the initial value of these bits is "00_B", the PLL does not operate normally with this setting. Be sure to set the bits to any value other than "00_B" either before setting the sub PLL clock oscillation enable bit (SPEN) to "1" or before setting the clock mode selection bits in the system clock control register (SYCC:SCS1, SCS0) to "01_B".
• These bits can be updated only when the sub PLL clock is stopped. Consequently, you should not update the bits either with the sub PLL clock oscillation enable bit (SPEN) set to "1" or with the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) set to "01_B". (It is however possible to set these bits at the same time as setting SPEN to "1".)</td></tr>
<tr><td>bit0</td><td>SPRDY:
Sub PLL clock oscillation stability bit
(Dual clock product only)</td><td>Indicates whether sub PLL clock oscillation has become stable.
• When set to "1", the SPRDY bit indicates that the oscillation stabilization wait time for the sub PLL clock has passed.
• When set to "0", the SPRDY bit indicates that the clock controller is in the sub PLL clock oscillation stabilization wait state or that sub PLL clock oscillation has been stopped.
This bit is read-only. Any value written is meaningless.
On single clock product, the value of the bit is meaningless.</td></tr>
</table>

6.5 Oscillation Stabilization Wait Time Setting Register (WATR)

This register is used to set the oscillation stabilization wait time.

■ Configuration of Oscillation Stabilization Wait Time Setting Register (WATR)

Figure 6.5-1 Configuration of Oscillation Stabilization Wait Time Setting Register (WATR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0005_H	SWT3	SWT2	SWT1	SWT0	MWT3	MWT2	MWT1	MWT0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

MWT3	MWT2	MWT1	MWT0	Number of Cycles	Main Oscillation Clock F_{CH} = 4 MHz
1	1	1	1	$2^{14}-2$	$(2^{14}-2)/F_{CH}$ About 4.10 ms
1	1	1	0	$2^{13}-2$	$(2^{13}-2)/F_{CH}$ About 2.05ms
1	1	0	1	$2^{12}-2$	$(2^{12}-2)/F_{CH}$ About 1.02ms
1	1	0	0	$2^{11}-2$	$(2^{11}-2)/F_{CH}$ 511.5 µs
1	0	1	1	$2^{10}-2$	$(2^{10}-2)/F_{CH}$ 255.5 µs
1	0	1	0	$2^{9}-2$	$(2^{9}-2)/F_{CH}$ 127.5 µs
1	0	0	1	$2^{8}-2$	$(2^{8}-2)/F_{CH}$ 63.5 µs
1	0	0	0	$2^{7}-2$	$(2^{7}-2)/F_{CH}$ 31.5 µs
0	1	1	1	$2^{6}-2$	$(2^{6}-2)/F_{CH}$ 15.5 µs
0	1	1	0	$2^{5}-2$	$(2^{5}-2)/F_{CH}$ 7.5 µs
0	1	0	1	$2^{4}-2$	$(2^{4}-2)/F_{CH}$ 3.5 µs
0	1	0	0	$2^{3}-2$	$(2^{3}-2)/F_{CH}$ 1.5 µs
0	0	1	1	$2^{2}-2$	$(2^{2}-2)/F_{CH}$ 0.5 µs
0	0	1	0	$2^{1}-2$	$(2^{1}-2)/F_{CH}$ 0.0 µs
0	0	0	1	$2^{1}-2$	$(2^{1}-2)/F_{CH}$ 0.0 µs
0	0	0	0	$2^{1}-2$	$(2^{1}-2)/F_{CH}$ 0.0 µs

SWT3	SWT2	SWT1	SWT0	Number of Cycles	Sub Oscillation Clock F_{CL} = 32.768 kHz
1	1	1	1	$2^{15}-2$	$(2^{15}-2)/F_{CL}$ About 1.00s
1	1	1	0	$2^{14}-2$	$(2^{14}-2)/F_{CL}$ About 0.5s
1	1	0	1	$2^{13}-2$	$(2^{13}-2)/F_{CL}$ About 0.25s
1	1	0	0	$2^{12}-2$	$(2^{12}-2)/F_{CL}$ About 0.125s
1	0	1	1	$2^{11}-2$	$(2^{11}-2)/F_{CL}$ About 62.44ms
1	0	1	0	$2^{10}-2$	$(2^{10}-2)/F_{CL}$ About 31.19ms
1	0	0	1	$2^{9}-2$	$(2^{9}-2)/F_{CL}$ About 15.56ms
1	0	0	0	$2^{8}-2$	$(2^{8}-2)/F_{CL}$ About 7.75ms
0	1	1	1	$2^{7}-2$	$(2^{7}-2)/F_{CL}$ About 3.85ms
0	1	1	0	$2^{6}-2$	$(2^{6}-2)/F_{CL}$ About 1.89ms
0	1	0	1	$2^{5}-2$	$(2^{5}-2)/F_{CL}$ About 915.5 µs
0	1	0	0	$2^{4}-2$	$(2^{4}-2)/F_{CL}$ About 427.2 µs
0	0	1	1	$2^{3}-2$	$(2^{3}-2)/F_{CL}$ About 183.1 µs
0	0	1	0	$2^{2}-2$	$(2^{2}-2)/F_{CL}$ About 61.0 µs
0	0	0	1	$2^{1}-2$	$(2^{1}-2)/F_{CL}$ 0.0 µs
0	0	0	0	$2^{1}-2$	$(2^{1}-2)/F_{CL}$ 0.0 µs

R/W : Readable/writable (Read value is the same as write value)

(shaded) : Initial value (For mask ROM products, initial oscillation stabilization time depends on the option setting when ordering mask ROM.)

Table 6.5-1 Functions of Bits in Oscillation Stabilization Wait Time Setting Register (WATR) (1 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7
to
bit4</td><td>SWT3, SWT2, SWT1, SWT0: Sub Clock Oscillation stabilization wait time select bits</td><td>Set the sub clock oscillation stabilization wait time.
<table>
<tr><th>SWT3 SWT2 SWT1 SWT0</th><th>Number of Cycles</th><th colspan="2">Sub Clock F_{CL} = 32.768kHz</th></tr>
<tr><td>1111_B</td><td>2^{15}-2</td><td>$(2^{15}-2) /F_{CL}$</td><td>About 1.0 s</td></tr>
<tr><td>1110_B</td><td>2^{14}-2</td><td>$(2^{14}-2) /F_{CL}$</td><td>About 0.5 s</td></tr>
<tr><td>1101_B</td><td>2^{13}-2</td><td>$(2^{13}-2) /F_{CL}$</td><td>About 0.25 s</td></tr>
<tr><td>1100_B</td><td>2^{12}-2</td><td>$(2^{12}-2) /F_{CL}$</td><td>About 0.125 s</td></tr>
<tr><td>1011_B</td><td>2^{11}-2</td><td>$(2^{11}-2) /F_{CL}$</td><td>About 62.44 ms</td></tr>
<tr><td>1010_B</td><td>2^{10}-2</td><td>$(2^{10}-2) /F_{CL}$</td><td>About 31.19 ms</td></tr>
<tr><td>1001_B</td><td>2^{9}-2</td><td>$(2^{9}-2) /F_{CL}$</td><td>About 15.56 ms</td></tr>
<tr><td>1000_B</td><td>2^{8}-2</td><td>$(2^{8}-2) /F_{CL}$</td><td>About 7.75 ms</td></tr>
<tr><td>0111_B</td><td>2^{7}-2</td><td>$(2^{7}-2) /F_{CL}$</td><td>About 3.85 ms</td></tr>
<tr><td>0110_B</td><td>2^{6}-2</td><td>$(2^{6}-2) /F_{CL}$</td><td>About 1.89 ms</td></tr>
<tr><td>0101_B</td><td>2^{5}-2</td><td>$(2^{5}-2) /F_{CL}$</td><td>About 9 15.5µs</td></tr>
<tr><td>0100_B</td><td>2^{4}-2</td><td>$(2^{4}-2) /F_{CL}$</td><td>About 4 27.2µs</td></tr>
<tr><td>0011_B</td><td>2^{3}-2</td><td>$(2^{3}-2) /F_{CL}$</td><td>About 183.1 µs</td></tr>
<tr><td>0010_B</td><td>2^{2}-2</td><td>$(2^{2}-2) /F_{CL}$</td><td>About 61.0 µs</td></tr>
<tr><td>0001_B</td><td>2^{1}-2</td><td>$(2^{1}-2) /F_{CL}$</td><td>0.0 µs</td></tr>
<tr><td>0000_B</td><td>2^{1}-2</td><td>$(2^{1}-2) /F_{CL}$</td><td>0.0 µs</td></tr>
</table>
On single clock product, the value of these bits is meaningless.
Number of cycles in the above table is for a minimum value. Add $1/F_{CL}$ to the number of cycle in the above table for a maximum value.
Note: Do not update these bits during sub clock oscillation stabilization wait time.You should update them either with the sub clock oscillation stability bit in the system clock control register (SYCC:SRDY) set to "1" or in sub clock mode or sub PLL clock mode.You can also update them while the sub clock is stopped with the sub clock oscillation stop bit in the system clock control register (SYCC:SUBS) set to "1" in main clock mode or main PLL clock mode.</td></tr>
</table>

Table 6.5-1 Functions of Bits in Oscillation Stabilization Wait Time Setting Register (WATR) (2 / 2)

Bit name		Function
bit3 to bit0	MWT3, MWT2, MWT1, MWT0: Main clock Oscillation stabilization wait time select bits	Set the main clock oscillation stabilization wait time. (see the sub-table below) Number of cycles is for a minimum value. Add $+1/F_{CH}$ to the minimum value for a maximum value. Note: Do not update these bits during main clock oscillation stabilization wait time. You should update them in main clock mode or main PLL clock mode. You can also update them in sub clock mode.

MWT3 MWT2 MWT1 MWT0	Number of Cycles	Main Clock F_{CH} = 4 MHz	
1111_B	2^{14}-2	$(2^{14}-2)/F_{CH}$	About 4.10 ms
1110_B	2^{13}-2	$(2^{13}-2)/F_{CH}$	About 2.05 ms
1101_B	2^{12}-2	$(2^{12}-2)/F_{CH}$	About 1.02 ms
1100_B	2^{11}-2	$(2^{11}-2)/F_{CH}$	511.5 µs
1011_B	2^{10}-2	$(2^{10}-2)/F_{CH}$	255.5 µs
1010_B	2^{9}-2	$(2^{9}-2)/F_{CH}$	127.5 µs
1001_B	2^{8}-2	$(2^{8}-2)/F_{CH}$	63.5 µs
1000_B	2^{7}-2	$(2^{7}-2)/F_{CH}$	31.5 µs
0111_B	2^{6}-2	$(2^{6}-2)/F_{CH}$	15.5 µs
0110_B	2^{5}-2	$(2^{5}-2)/F_{CH}$	7.5 µs
0101_B	2^{4}-2	$(2^{4}-2)/F_{CH}$	3.5 µs
0100_B	2^{3}-2	$(2^{3}-2)/F_{CH}$	1.5 µs
0011_B	2^{2}-2	$(2^{2}-2)/F_{CH}$	0.5 µs
0010_B	2^{1}-2	$(2^{1}-2)/F_{CH}$	0.0 µs
0001_B	2^{1}-2	$(2^{1}-2)/F_{CH}$	0.0 µs
0000_B	2^{1}-2	$(2^{1}-2)/F_{CH}$	0.0 µs

6.6 Standby Control Register (STBC)

The standby control register (STBC) is used to control transition from the RUN state to sleep mode, stop mode, time-base timer mode, or watch mode, set the pin state in stop mode, time-base timer mode, and watch mode, and to control the generation of software resets.

■ Standby Control Register (STBC)

Figure 6.6-1 Standby Control Register (STBC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0008_H	STP	SLP	SPL	SRST	TMD	–	–	–	00000000_B
	R0,W	R0,W	R/W	R0,W	R0,W	R0/WX	R0/WX	R0/WX	

TMD	Watch bit		
	Read	Write	
0	Always reads "0".	Has no effect on the operation.	
1	–	Main clock mode Main PLL clock mode	Sub clock mode Sub PLL clock mode
		Causes transition to time-base time r mode	Causes transition to watch mode

SRST	Software reset bit	
	Read	Write
0	Always reads "0".	Has no effect on the operation
1	–	Generates a 3 machine clock reset signal

SPL	Pin state setting bit
0	Holds external pins in their immediately preceding state in stop mode, time-base timer mode, or watch mode
1	Places external pins in a high impedance state in stop mode, time-base timer mode, or watch mode.

SLP	Sleep bit	
	Read	Write
0	Always reads "0".	Has no effect on the operation
1	–	Causes transition to sleep mode

STP	Stop bit	
	Read	Write
0	Always reads "0".	Has no effect on the operation
1	–	Causes transition to stop mode

R/W : Readable/writable (Read value is the same as write value)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined
(shaded) : Initial value

Table 6.6-1 Functions of Bits in Standby Control Register (STBC)

Bit name		Function
bit7	STP: Stop bit	Sets transition to stop mode. **When set to "0":** has no effect on operation. **When set to "1":** the bit causes transition to stop mode. When read, the bit always returns "0". Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see Section "6.8.1 Notes on Using Standby Mode".
bit6	SLP: Sleep bit	Sets transition to sleep mode. **When set to "0":** has no effect on operation. **When set to "1":** the bit causes transition to sleep mode. When read, the bit always returns "0". Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see Section "6.8.1 Notes on Using Standby Mode".
bit5	SPL: Pin state setting bit	Sets the states of external pins in stop mode, time-base timer mode, and watch mode. **When set to "0":** the bit holds the states (levels) of external pins in stop mode, time-base timer mode, and watch mode. **When set to "1":** the bit places external pins in a high impedance state in stop mode, time-base timer mode, and watch mode. (Those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register.)
bit4	SRST: Software reset bit	Sets a software reset. **When set to "0":** has no effect on operation. **When set to "1":** the bit generates a 3 machine clock reset signal. When read, the bit always returns "0".
bit3	TMD: Watch bit	On dual clock product, this bit sets transition to time-base timer mode or watch mode. On single clock product, the bit sets transition to time-base timer mode. • Writing "1" to the bit in main clock mode or main PLL clock mode causes transition to time-base timer mode. • Writing "1" to the bit in sub clock mode or sub PLL clock mode causes transition to watch mode. • Writing "0" to this bit has no effect on operation. • When read, the bit always returns "0". Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see Section "6.8.1 Notes on Using Standby Mode".
bit2 to bit0	Undefined bits	The read value is always "0". These bits are undefined. This bit is read-only. Writing has no effect on operation.

Notes:

- Set the standby mode after making sure that the transition to clock mode has been completed by comparing the values of the clock mode monitor bits (SYCC:SCM1,SCM0) and clock mode setting bits (SYCC:SCS1,SCS0) in the system clock control register.
- If you write "1" simultaneously to two or more of the stop bit (STP), sleep bit (SLP), software reset bit (SRST), and watch bit (TMD), priority is given to them in the following order:

  (1) Software reset bit (SRST)

  (2) Stop bit (STP)

  (3) Watch bit (TMD)

  (4) Sleep bit (SLP)

When released from the standby mode, the device returns to the normal operating status.

6.7 Clock Mode

The clock modes available are: main clock mode, sub clock mode, main PLL clock mode, and sub PLL clock mode. Mode switching takes place according to the settings in the system clock control register (SYCC).
Sub clock mode and sub PLL clock mode are not supported by single clock product.

■ Operations in Main Clock Mode

Main clock mode uses the main clock as the machine clock for the CPU and peripheral resources.

The time-base timer operates with the main clock.

The watch prescaler and watch counter operate with the sub clock (on dual clock product).

If you set standby mode during operation in main clock mode, the device can enter sleep mode, stop mode, or time-base timer mode.

After a reset, main clock mode is always set regardless of the clock mode used before the reset.

■ Operations in Sub Clock Mode (on Dual Clock Product)

Sub clock mode uses the sub clock as the machine clock for the CPU and peripheral resources with main clock oscillation stopped. In this mode, the time-base timer remains stopped as it requires the main clock for operation.

If you set standby mode during operation in sub clock mode, the device can enter sleep mode, stop mode, or watch mode.

■ Operations in Main PLL Clock Mode

Main PLL clock mode uses the main PLL clock as the machine clock for the CPU and peripheral resources. The time-base timer and watchdog timer operate with the main clock.

The watch prescaler and watch counter operate with the sub clock (on dual clock product).

If you set standby mode during operation in main PLL clock mode, the device can enter sleep mode, stop mode, or time-base timer mode.

■ Operations in Sub PLL Clock Mode (on Dual Clock Product)

Sub PLL clock mode uses the sub PLL clock as the machine clock for the CPU and peripheral resources with main clock oscillation stopped. In this mode, the time-base timer remains stopped as it requires the main clock for operation. The watch prescaler and watch counter operate with the sub clock.

If you set standby mode during operation in sub PLL clock mode, the device can enter sleep mode, stop mode, or watch mode.

■ Clock Mode State Transition Diagram

The clock modes available are: main clock mode, main PLL clock mode, sub clock mode, and sub PLL clock mode. The device can switch between these modes according to the settings in the system clock control register (SYCC).

Figure 6.7-1 Clock Mode State Transition Diagram (Dual Clock Product)

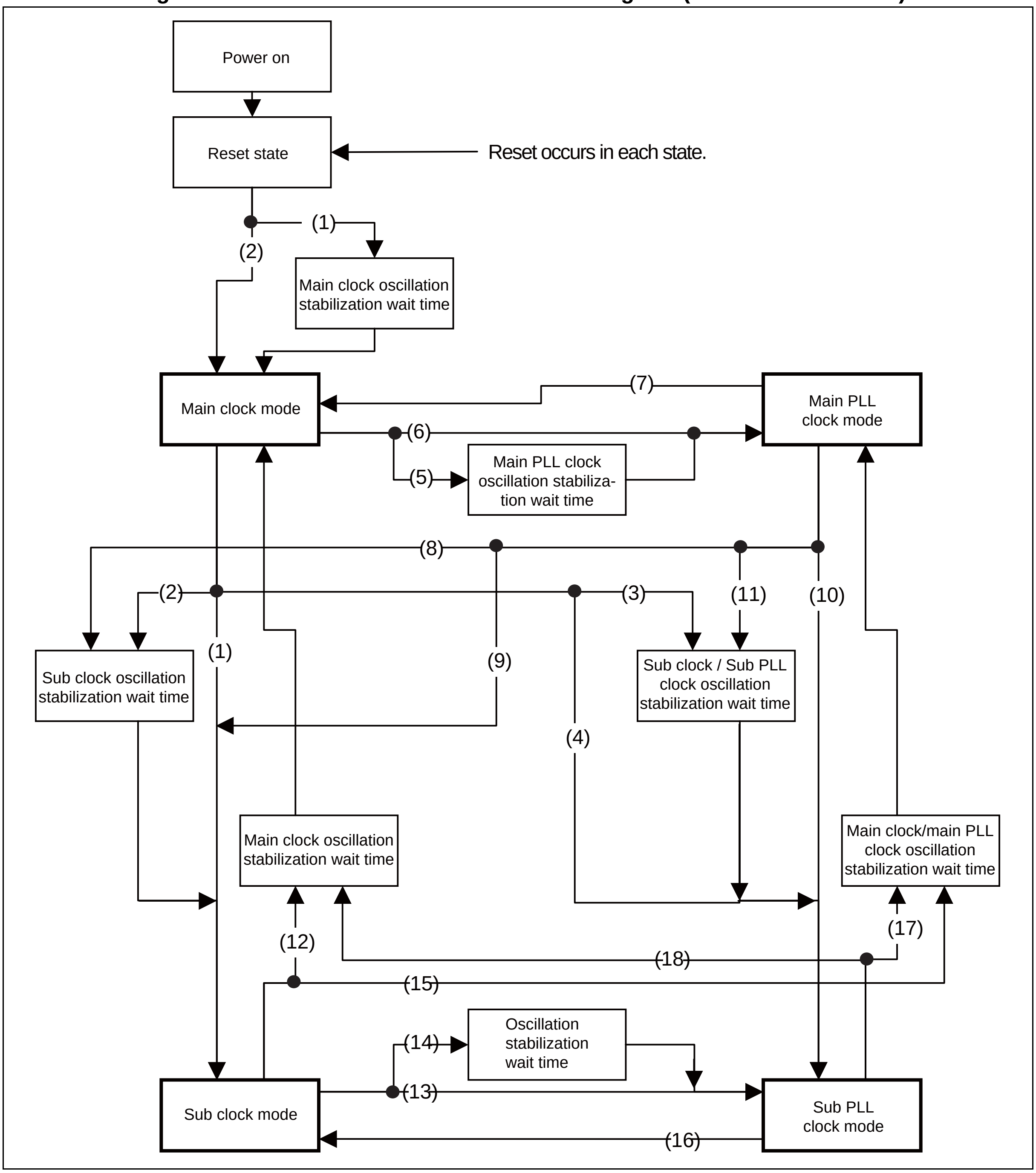


Figure 6.7-2 Clock Mode State Transition Diagram (Single Clock Product)

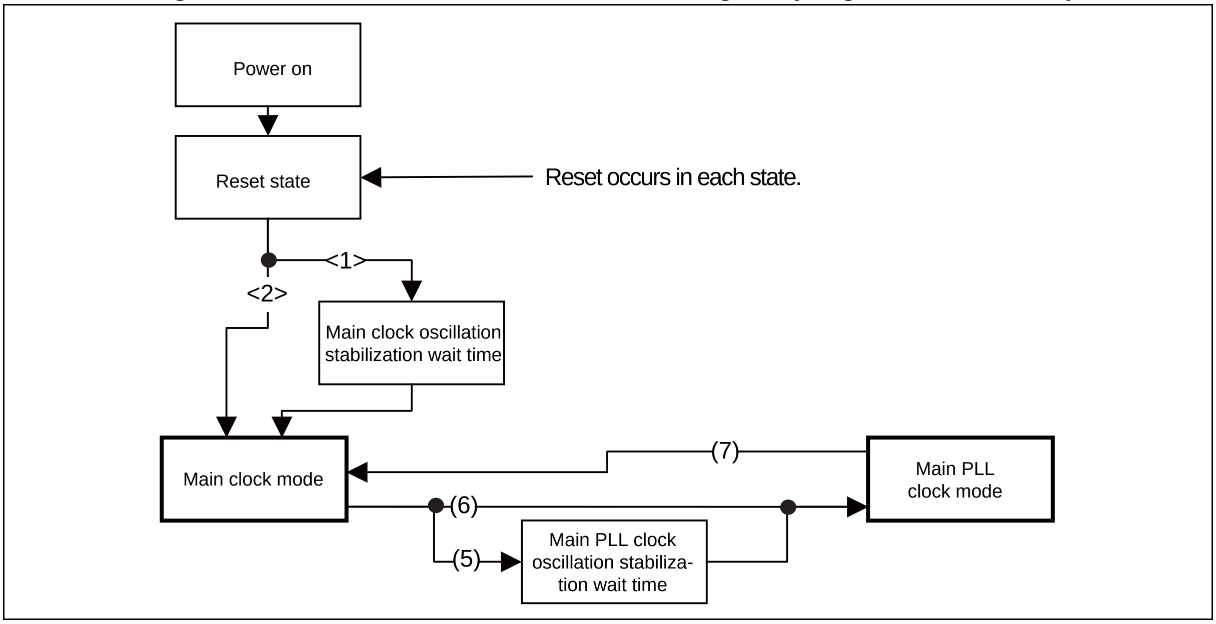


Table 6.7-1 Clock Mode State Transition Table (1 / 2)

<table>
<tr><th></th><th>Current State</th><th>Next State</th><th>Description</th></tr>
<tr><td><1></td><td rowspan="2">Reset State</td><td rowspan="2">Main Clock</td><td rowspan="2">After a reset, the device waits for the main clock oscillation stabilization wait time to elapse and enters main clock mode. If the reset is a watchdog reset, software reset, or external reset caused in main clock mode or main PLL clock mode, however, the device does not wait for the main clock oscillation stabilization wait time to elapse.</td></tr>
<tr><td><2></td></tr>
<tr><td>(1)</td><td rowspan="6">Main Clock</td><td rowspan="2">Sub Clock</td><td rowspan="2">The device enters sub clock mode when the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "00_B". Note, however, that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on.</td></tr>
<tr><td>(2)</td></tr>
<tr><td>(3)</td><td rowspan="2">Sub PLL Clock</td><td rowspan="2">When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "01_B", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in main clock mode. Note also that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub PLL clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on.
When the device waits for the sub clock oscillation stabilization wait time or sub PLL clock oscillation stabilization wait time, it waits for whichever is longer to elapse.</td></tr>
<tr><td>(4)</td></tr>
<tr><td>(5)</td><td rowspan="2">Main PLL Clock</td><td rowspan="2">When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "11_B", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time.Note, however, that the device does not wait for the main PLL clock oscillation stabilization wait time to elapse if the main PLL clock has been oscillating according to the setting of the main PLL clock oscillation enable bit in the PLL control register (PLLC: MPEN).</td></tr>
<tr><td>(6)</td></tr>
</table>

MB95150/M Series

Table 6.7-1 Clock Mode State Transition Table (2 / 2)

	Current State	Next State	Description
(7)	Main PLL Clock	Main Clock	The device enters main clock mode when the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "10_B".
(8)		Sub Clock	The device enters sub clock mode when the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "00_B". Note, however, that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main PLL clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on.
(9			
(10)		Sub PLL Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "01_B", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in main PLL clock mode. Note also that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub PLL clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main PLL clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on. When the device waits for the sub clock oscillation stabilization wait time or sub PLL clock oscillation stabilization wait time, it waits for whichever is longer to elapse.
(11)			
(12)	Sub Clock	Main Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "10_B", the device enters main clock mode after waiting for the main clock oscillation stabilization wait time.
(13)		Sub PLL Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "01_B", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time.Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in sub clock mode.
(14)			
(15)		Main PLL Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "11_B", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time or main clock oscillation stabilization wait time to elapse, whichever is longer.
(16)	Sub PLL Clock	Sub Clock	The device enters sub clock mode when the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "00_B".
(17)		Main PLL Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "11_B", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time or main clock oscillation stabilization wait time to elapse, whichever is longer.
(18)		Main Clock	When the system clock select bits in the system clock control register (SYCC:SCS1, SCS0) are set to "10_B", the device enters main clock mode after waiting for the main clock oscillation stabilization wait time.

6.8 Operations in Low-power Consumption Modes (Standby Modes)

The standby modes available are: sleep mode, stop mode, time-base timer mode, and watch mode.

■ Overview of Transitions to and from Standby Mode

The standby modes available are: sleep mode, stop mode, time-base timer mode, and watch mode. The device enters standby mode according to the settings in the standby control register (STBC).

The device is released from standby mode in response to an interrupt or reset. Before transition to normal operation, the device waits for the oscillation stabilization wait time to elapse as required.

When released from standby mode by a reset, the device returns to main clock mode. When released from standby mode by an interrupt, the device enters the clock mode in which the device was before entering the standby mode.

■ Pin States in Standby Mode

The pin state setting bit (STBC:SPL) of the standby control register can be used to set the I/O port/peripheral resource pins in the stop mode, time-base timer mode, or watch mode to hold their immediately preceding state or to be placed in a high inpedance state.

See "Pin Status" for the states of all pins in standby modes.

6.8.1 Notes on Using Standby Mode

Even if the standby control register (STBC) sets standby mode, transition to the standby mode does not take place when an interrupt request has been issued from a peripheral resource. When the device returns from standby mode to the normal operating state in response to an interrupt, the operation that follows varies depending on whether the interrupt request is accepted or not.

■ Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction.

The device requires four machine clock cycles before entering standby mode after it is set in the standby control register. During that period, the CPU executes the program. To avoid program execution during this transition to standby mode, enter at least three NOP instructions.

The device operates normally if you place instructions other than NOP instructions. In that case, however, note that the device may execute the instructions to be executed after being released from standby mode before entering the standby mode and that the device may enter the standby mode during instruction execution, which is resumed after the device is released from the standby mode (increasing the number of instruction execution cycles).

■ Check That Clock-mode Transition has been Completed before Setting Standby Mode.

Before setting standby mode, make sure that clock-mode transition has been completed by comparing the values of the clock mode monitor bit (SYCC: SCM1, SCM0) and clock mode setting bit (SYCC:SCS1, SCS0) in the system clock control register.

■ An Interrupt Request may Suppress Transition to Standby Mode.

If an attempt is made to set a standby mode while an interrupt request with an interrupt level higher than "11_B" has been issued, the device ignores the attempt to write to the standby control register and continues instruction execution without entering the standby mode. The device does not enter the standby mode even after having serviced the interrupt.

This behavior is the same as when interrupts are disabled by the interrupt enable flag (CCR:I) and interrupt level bits in the condition code register (CCR:IL1, IL0) of the CPU.

■ Standby Mode is Also Canceled when the CPU Rejects Interrupts.

When an interrupt request with an interrupt level higher than "11_B" is issued in standby mode, the device is released from the standby mode regardless of the settings of the interrupt enable flag (CCR: I) and interrupt level bits (CCR:IL1, IL0) of the condition code register of the CPU.

After being released from standby mode, the device services the interrupt when the CPU's condition code register has been set to accept interrupts. If the register has been set to reject interrupts, the device resumes processing from the instruction that follows the last instruction executed before entering the standby mode.

■ Standby Mode State Transition Diagram

Figure 6.8-1 and Figure 6.8-2 are standby mode state transition diagrams.

Figure 6.8-1 Standby Mode State Transition Diagram (Dual Clock Product)

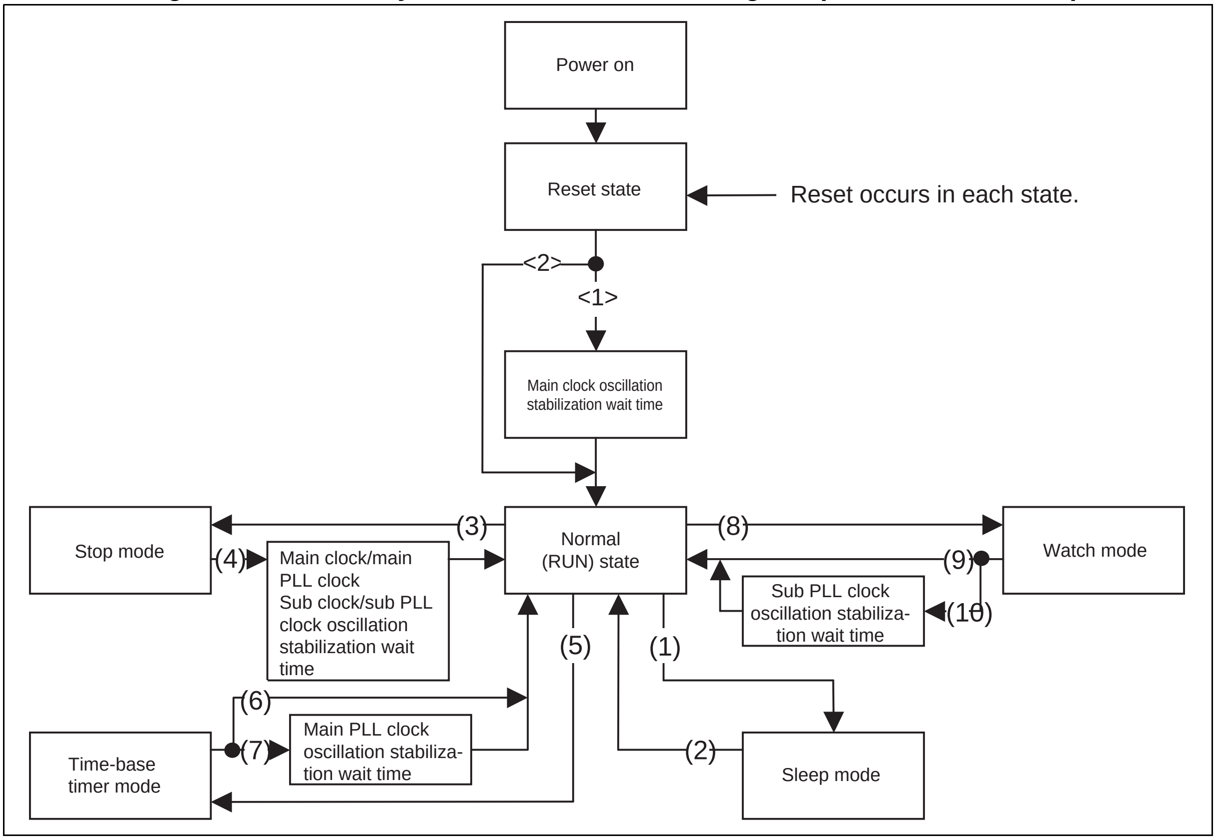


Figure 6.8-2 Standby Mode State Transition Diagram (Single Clock Product)

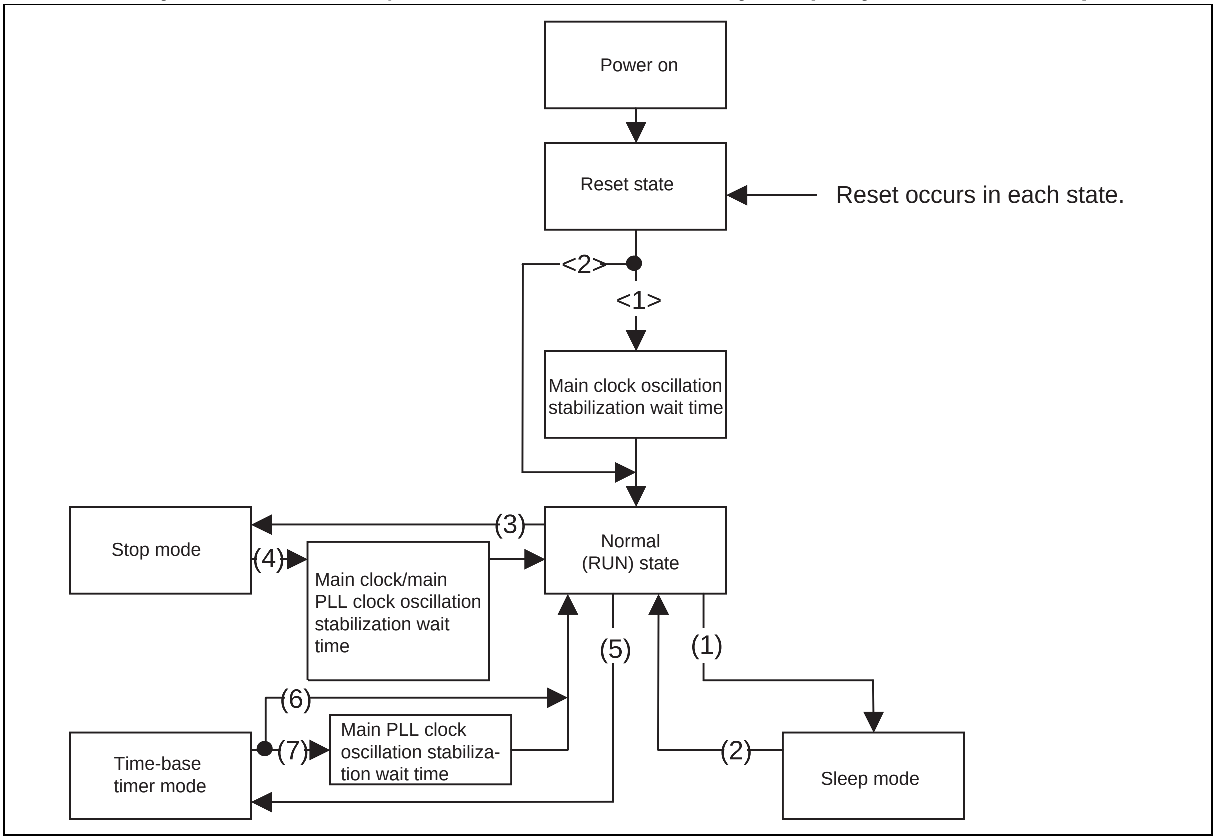


Table 6.8-1 State Transition Diagram (Transitions to and from Standby Modes)

	State Transition	Description
<1>	Normal operation from reset state	After a reset, the device enters main clock mode. If the reset is a power-on reset, the device always waits for the main clock oscillation stabilization wait time to elapse. When the clock mode before the reset is sub clock mode or sub PLL clock mode, the device waits for the main clock oscillation stabilization wait time to elapse. The device waits for it as well when the standby mode is stop mode. When the clock mode before the reset is main clock mode or main PLL clock mode and the standby mode is other than stop mode, the device does not wait for the main clock oscillation stabilization wait time to elapse even after entering a reset state in response to a watchdog reset, software reset, or external reset.
<2>		
(1)	Sleep mode	The device enters sleep mode when "1" is written to the sleep bit in the standby control register (STBC: SLP).
(2)		The device returns to the RUN state in response to an interrupt from a peripheral resource.
(3)	Stop mode	The device enters stop mode when "1" is written to the stop bit in the standby control register (STBC: STP).
(4)		In response to an external interrupt, the device returns to the RUN state after waiting for the oscillation stabilization wait time required for each clock mode. When the device waits for a PLL oscillation stabilization wait time, it waits for the relevant oscillation stabilization wait time or PLL oscillation stabilization wait time to elapse, whichever is longer.
(5)	Time-base timer mode	The device enters time-base timer mode when "1" is written to the watch bit in the standby control register (STBC: TMD) in main clock mode or main PLL clock mode.
(6)		The device returns to the RUN state in response to a time-base timer interrupt, watch prescaler/watch counter interrupt, or external interrupt. When the clock mode is main PLL clock mode, the device waits for the main PLL clock oscillation stabilization wait time to elapse.If the main PLL oscillation enable bit in the PLL control register (PLLC: MPEN) contains "1", however, the device does not wait for that time to elapse even when the clock mode is main PLL clock mode.
(7)		
(8)	Watch mode	The device enters watch mode when "1" is written to the watch bit in the standby control register (STBC: TMD) in sub clock mode or sub PLL clock mode.
(9)		The device returns to the normal operating state in response to a watch prescaler/watch counter interrupt or external interrupt. When the clock mode is sub PLL clock mode, the device waits for the sub PLL clock oscillation stabilization wait time to elapse.If the sub PLL oscillation enable bit in the PLL control register (PLLC: SPEN) contains "1", however, the device does not wait for that time to elapse even when the clock mode is sub PLL clock mode.
(10)		

6.8.2 Sleep Mode

Sleep mode stops the operations of the CPU and watchdog timer.

■ Operations in Sleep Mode

Sleep mode stops the operating clock for the CPU and watchdog timer. In this mode, the CPU stops while retaining the contents of registers and RAM that exist immediately before the transition to sleep mode, but the peripheral resources except the watchdog timer continue operating.

● Transition to sleep mode

Writing "1" to the sleep bit in the standby control register (STBC:SLP) causes the device to enter sleep mode.

● Cancellation of sleep mode

A reset or an interrupt from a peripheral resource releases the device from sleep mode.

6.8.3 Stop Mode

Stop mode stops the main clock.

■ Operations in Stop Mode

Stop mode stops the main clock. In this mode, the device stops all the functions except external interrupt and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to stop mode.

In main clock mode or main PLL clock mode, however, you can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, the watch prescaler and watch counter operate.

● Transition to stop mode

Writing "1" to the stop bit in the standby control register (STBC:STP) causes the device to enter stop mode. At this time, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

In main clock mode or main PLL clock mode, a time-base timer interrupt request may be generated while the device is waiting for main clock oscillation to stabilize after being released from stop mode by an interrupt. If the interrupt interval time of the time-base timer is shorter than the main clock oscillation stabilization wait time, you should disable interrupt requests output from the time-base timer before entering stop mode, thereby preventing unexpected interrupts from occurring.

You should also disable interrupt requests output from the watch prescaler before entering stop mode in sub clock mode or sub PLL clock mode.

● Cancellation of stop mode

The device is released from stop mode in response to a reset or an external interrupt.

In main clock mode or main PLL clock mode, you can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, you can also release the device from stop mode using an interrupt by the watch prescaler or watch counter.

Note:

When stop mode is canceled via an interrupt, peripheral resources placed into stop mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from stop mode, initialize each peripheral resource as necessary.

6.8.4 Time-base Timer Mode

Time-base timer mode allows only the main clock oscillation, sub clock oscillation, time-base timer, and watch prescaler to work. The operating clock for the CPU and peripheral resources is stopped in this mode.

■ Operations in Time-base Timer Mode

In time-base timer mode, main clock supply is stopped except for the time-base timer. The device stops all the functions except time-base timer, external interrupt and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to time-base timer mode.

You can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, the watch prescaler and watch counter operate.

● Transition to time-base timer mode

Writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to enter time-base timer mode if the system clock monitor bits in the system clock control register (SYCC: SCM1, SCM0) are "10_B" or "11_B".

The device can enter time-base timer mode only when the clock mode is main clock mode or main PLL clock mode.

Upon transition to time-base timer mode, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

● Cancellation of time-base timer mode

The device is released from time-base timer mode in response to a reset, time-base timer interrupt, or external interrupt.

You can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, you can also release the device from time-base timer mode using an interrupt by the watch prescaler or watch counter.

Note:

When time-base timer mode is canceled via an interrupt, peripheral resources placed into time-base timer mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from time-base timer mode, initialize each peripheral resource as necessary.

6.8.5 Watch Mode

In watch mode, the operating clock for the CPU and peripheral resources is stopped. The device stops all the functions except the LCD controller, watch prescaler, watch counter, external interrupt, and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to watch mode.

■ Operations in Watch Mode

In watch mode, the operating clock for the CPU and peripheral resources is stopped. The device stops all the functions except the LCD controller, watch prescaler, watch counter, external interrupt, and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to watch mode.

● Transition to watch mode

Writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to enter watch mode if the system clock monitor bits in the system clock control register (SYCC: SCM1, SCM0) are "00_B" or "01_B".

The device can enter watch mode only when the clock mode is sub clock mode or sub PLL clock mode. Upon transition to watch mode, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

● Cancellation of watch mode

The device is released from watch mode in response to a reset, watch interrupt, or external interrupt.

Note:

When watch mode is canceled via an interrupt, peripheral resources placed into watch mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from watch mode, initialize each peripheral resource as necessary.

6.9 Clock Oscillator Circuits

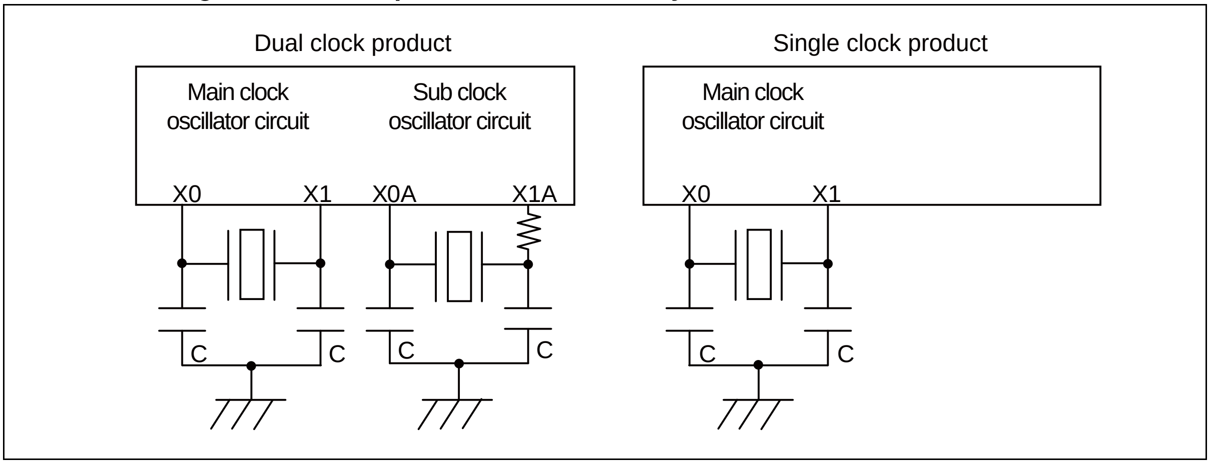
The clock oscillator circuit generates an internal clock with an oscillator connected to or a clock signal input to the clock oscillation pin.

■ Clock Oscillator Circuit

● Using crystal and ceramic oscillators

Connect crystal and ceramic oscillators as shown in Figure 6.9-1.

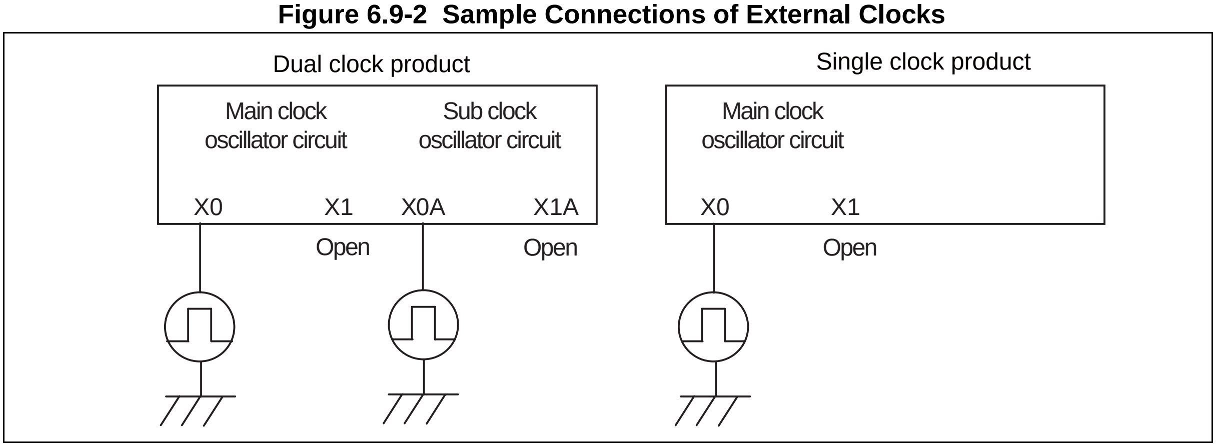
Figure 6.9-1 Sample Connections of Crystal and Ceramic Oscillators



● Using external clock

As shown inFigure 6.9-2, connect the external clock to the X0 pin while leaving the X1 pin open. To supply the sub clock from an external source, connect the external clock to the X0A pin while leaving the X1A pin open.

Figure 6.9-2 Sample Connections of External Clocks



Note:

If you use only the main clock without using sub clock oscillation on a dual clock product and it enters sub clock mode for some reason, there is no solution for recovering its operation as there is no clock supply available. If you use the main clock alone, therefore, be sure to select a single clock product.

MB95150/M Series

6.10 Overview of Prescaler

The prescaler generates the count clock source for various peripheral resources from the machine clock (MCLK) and the count clock output from the time-base timer.

■ Prescaler

The prescaler generates the count clock source for various peripheral resources from the machine clock (MCLK) that drives the CPU and the count clock ($2^7/F_{CH}$ or $2^8/ F_{CH}$) output from the time-base timer. The count clock source is a clock frequency-divided by the prescaler or a buffered clock, used by the peripheral resources listed below.

Note that the prescaler has no control register and operates continuously driven by the machine clock (MCLK) and the count clock ($2^7/ F_{CH}$ or $2^8/ F_{CH}$) of the time-base timer.

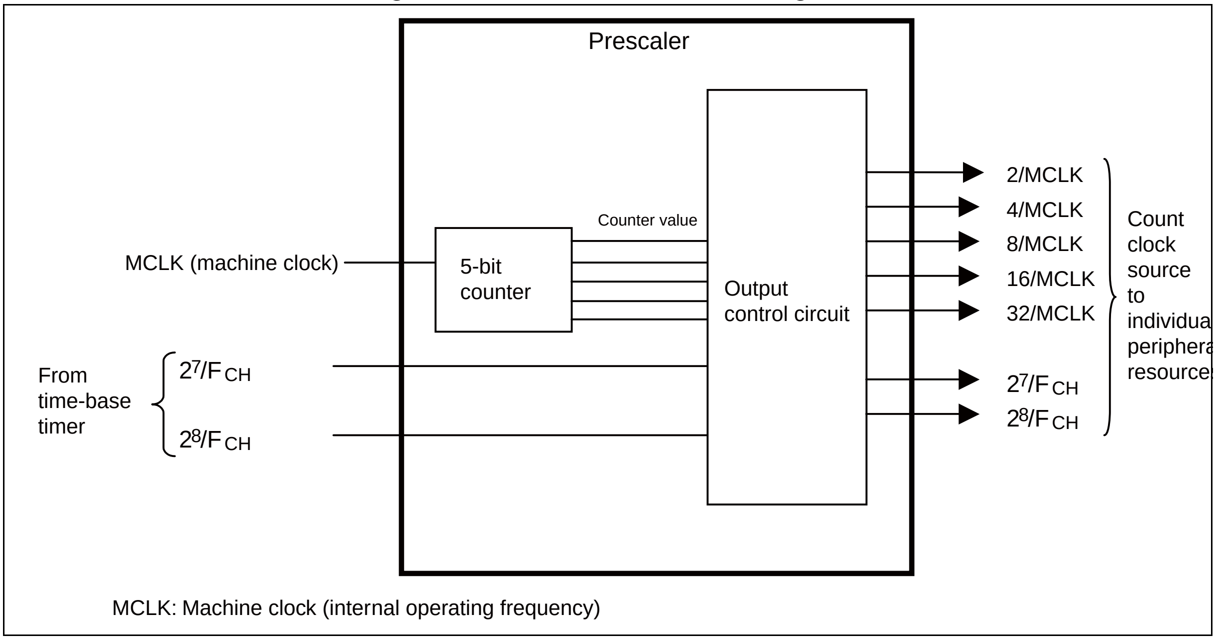
- 8/16-bit compound timer
- 16-bit reload timer
- 8/16-bit PPG Timer
- 16-bit PPG timer
- UART/SIO Dedicated Baud Rate Generator
- 8/10-bit A/D converter

6.11 Configuration of Prescaler

Figure 6.11-1 is a block diagram of the prescaler.

■ Prescaler Block Diagram

Figure 6.11-1 Prescaler Block Diagram



- 5 -bit counter

  The machine clock (MCLK) is counted by a 5-bit counter and the count value is output to the output control circuit.

- Output control circuit

  The division ratio (divided by 2/4/8/16/32) is determined by the counter value of the 5-bit counter. Clocks generated by dividing the machine clock by this value will be supplied to individual peripheral resources. The circuit also buffers the clock from the time-base timer ($2^7/F_{CH}$ and $2^8/F_{CH}$) and supplies it to the peripheral resources.

■ Input Clock

The prescaler uses the machine clock or the clock output from the time-base timer as the input clock.

■ Output Clock

The prescaler supplies clocks to the 8/10-bit compound timer, 8/16-bit PPG timer, 16-bit PPG timer, UART/SIO dedicated baud rate generator, and 8/10-bit A/D converter.

6.12 Operating Explanation of Prescaler

The prescaler generates count clock sources to individual peripheral resources.

■ Operations of Prescaler

The prescaler generates count clock sources from the frequency-divided version of the machine clock (MCLK) and buffered signals from the time-base timer ($2^7/F_{CH}$, $2^8/F_{CH}$) and supplies them to individual peripheral resources. The prescaler remains operating as long as the machine clock and time-base timer clocks are supplied.

Table 6.12-1 lists the count clock sources generated by the prescaler.

Table 6.12-1 Count Clock Sources Generated by Prescaler

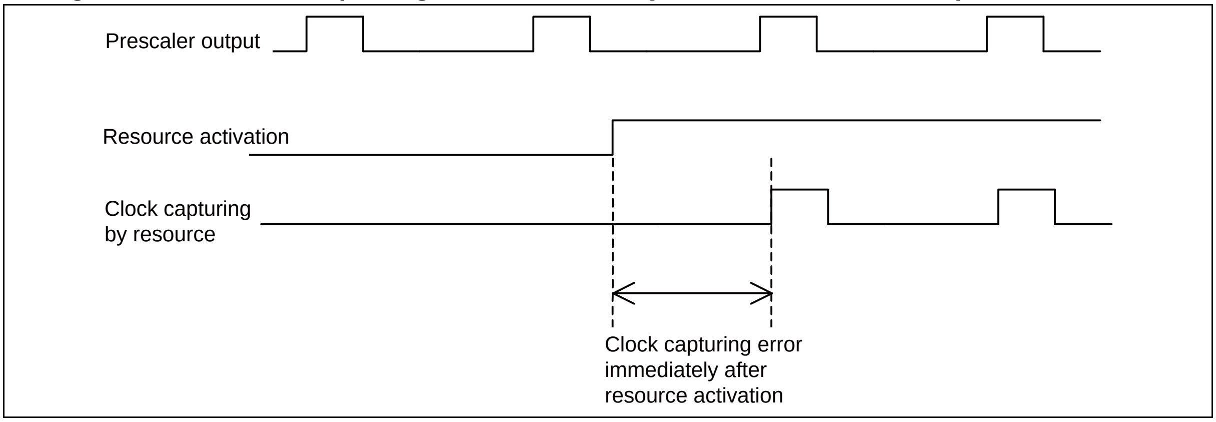
Count Clock Source Cycle	Cycle (F_{CH} =10MHz, MCLK=10MHz)		Cycle (F_{CH} =16MHz, MCLK=16MHz)		Cycle (F_{CH} =16.25MHz, MCLK=16.25MHz)	
2/MCLK	MCLK/2	(5MHz)	MCLK/2	(8MHz)	MCLK/2	(8.125MHz)
4/MCLK	MCLK/4	(2.5MHz)	MCLK/4	(4MHz)	MCLK/4	(4.0625MHz)
8/MCLK	MCLK/8	(1.25MHz)	MCLK/8	(2MHz)	MCLK/8	(2.0313MHz)
16/MCLK	MCLK/16	(0.625MHz)	MCLK/16	(1MHz)	MCLK/16	(1.0156MHz)
32/MCLK	MCLK/32	(0.3125MHz)	MCLK/32	(0.5MHz)	MCLK/32	(0.5078MHz)
$2^7/F_{CH}$	$F_{CH}/2^7$	(78kHz)	$F_{CH}/2^7$	(125kHz)	$F_{CH}/2^7$	(127kHz)
$2^8/F_{CH}$	$F_{CH}/2^8$	(39kHz)	$F_{CH}/2^8$	(62.5kHz)	$F_{CH}/2^8$	(63.5kHz)

6.13 Notes on Use of Prescaler

This section gives notes on using the prescaler.

The prescaler uses the machine clock and time-base timer clock and operates continuously while these clocks are running. Accordingly, the operations of individual peripheral resources immediately after they are activated may involve an error of up to one cycle of the clock source captured by the resource, depending on the prescaler output value.

Figure 6.13-1 Clock Capturing Error Immediately after Activation of Peripheral Resources



The prescaler count value affects the following resources:

- UART/SIO
- 8/16-bit compound timer
- 8/16-bit PPG
- 16-bit PPG
- 8/10-bit A/D converter

CHAPTER 7
RESET

This section describes the reset operation.

Code: CM26-00104-1E

7.1 Reset Operation

When a reset factor occurs, the CPU stops the current execution immediately and enters the reset release wait state. When the device is released from the reset, the CPU reads mode data and the reset vector from internal ROM (mode fetch). When the power is turned on or when the device is released from a reset in sub clock mode, sub-PLL clock mode, or stop mode, the CPU performs mode fetch after the oscillation stabilization wait time has passed.

■ Reset Factors

Resets are classified into five reset factors.

Table 7.1-1 Reset Sources

Reset Sources	Reset Condition
External reset	"L" level input to the external reset pin
Software reset	"1" is written to the software reset bit (STBC: SRST) in the standby control register.
Watchdog reset	The watchdog timer causes an overflow.
Power-on reset/ low-voltage detection reset	The power is turned on or the supply voltage falls below the detected voltage. (Option)
Clock supervisor reset	Abnormal Stop of Clock Oscillation (Option)

● External reset

An external reset is generated upon "L" level input to the external reset pin ($\overline{\text{RST}}$).

An externally input reset signal is accepted asynchronously via the internal noise filter and generates an internal reset signal in synchronization with the machine clock to initialize the internal circuit.Consequently, a clock is necessary for internal circuit initialization. Clock input is therefore necessary for operation with an external clock. Note, however, that external pins (including I/O ports and peripheral resources) are reset asynchronously. Additionally, there are standard pulse-width values for external reset input. If the value is below the standard, the reset may not be accepted.

The standard value is listed on the data sheet. Please design your external reset circuit so that this standard is met.

● Software reset

Writing "1" to the software reset bit of the standby control register (STBC:SRST) generates a software reset.

● Watchdog reset

After the watchdog timer starts, a watchdog reset is generated if the watchdog timer is not cleared within a preset amount of time.

● Power-on reset/low-voltage detection reset (Option)

A power-on reset is generated when the power is turned on.

Some 5-V products have a low-voltage detection reset circuit (option) integrated.

The low-voltage detection reset circuit generates a reset if the power supply voltage falls below a predetermined level.

The logical function of the low-voltage detection reset is completely equivalent to the power-on reset. All the text in this manual concerning power-on resets applies to low-voltage detection resets as well.

For details about low-voltage detection resets, see "CHAPTER 25 LOW-VOLTAGE DETECTION RESET CIRCUIT".

● Clock Supervisor Reset (Option)

Some 5V products have the (optional) clock supervisor.

The clock supervisor monitors the main and sub clocks and generates a reset when the oscillation stops due to not given state transition but any abnormality.After reset, a clock occurred in the built-in RC oscillation circuit is provided internally.

For details on the clock supervisor, see "CHAPTER 26 CLOCK SUPERVISOR".

■ Reset Time

In the case of a software reset or watchdog reset, the reset time consists of a total of three machine clock cycles: one machine clock cycle at the machine clock frequency selected before the reset, and two machine clock cycles at the machine clock frequency initially set after the reset (1/32 of the main clock frequency). However, the reset time may be extended in machine clock cycles of the frequency selected before the reset, via the RAM access protection function which suppresses resets during RAM access. In addition, when in main clock oscillation stabilization standby mode, the reset time is further extended for the oscillation stabilization wait time.

External resets and resets are also affected by the RAM access protection function and main clock oscillation stabilization wait time.

In the case of a power-on reset or low-voltage detection reset, the reset continues during the oscillation stabilization wait time.

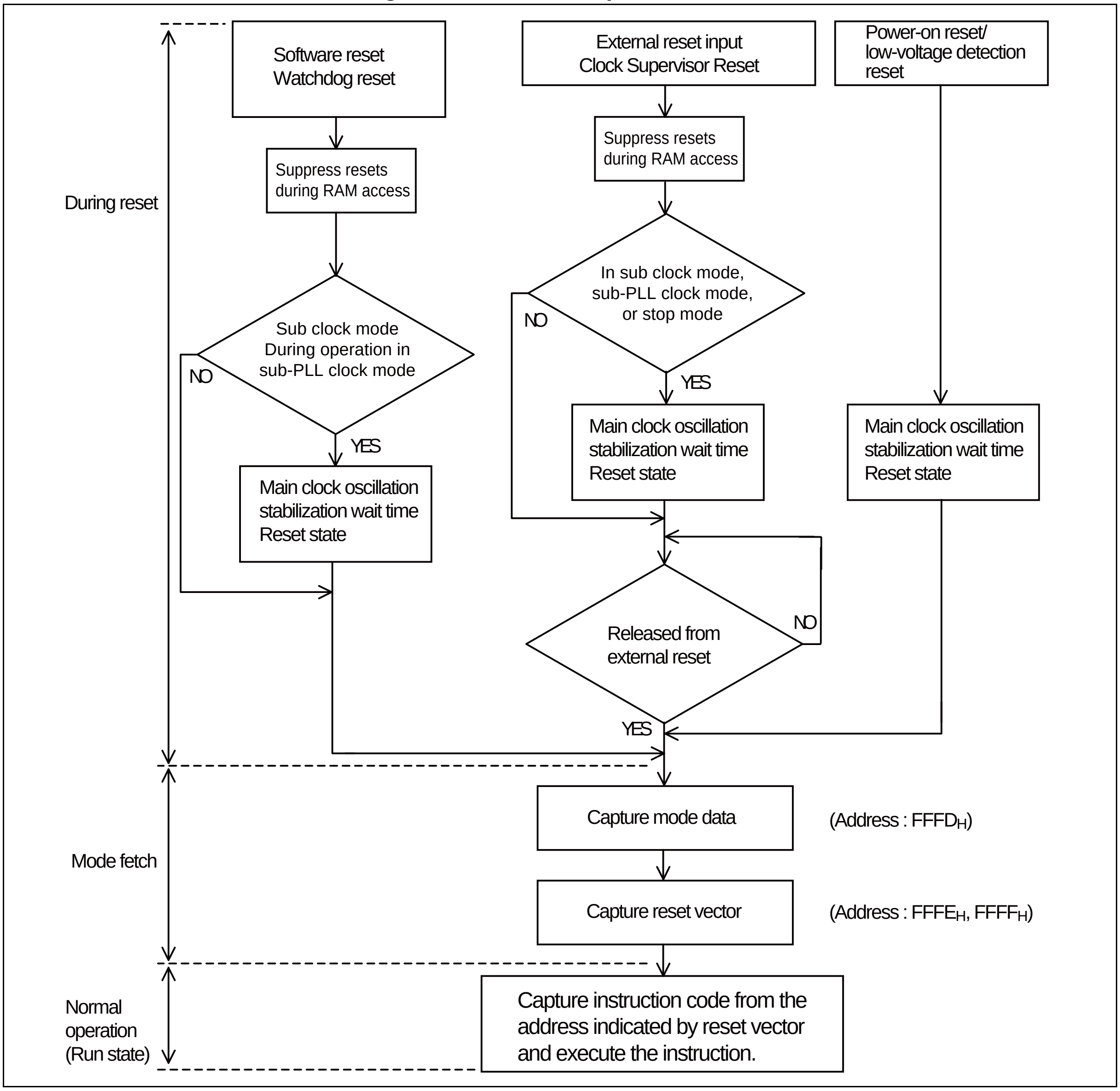
■ Reset Output

The $\overline{\text{RST}}$ pin of 5 V products with the reset (For details, see Table 1.2-1.) outputs "L" level during reset time. However, a reset pin does not output "L" level in the case of an external reset.

The $\overline{\text{RST}}$ pin of 3 V products and 5 V products without the reset outputs do not have an output function.

■ Overview of Reset Operation

Figure 7.1-1 Reset Operation Flow



In the case of a power-on reset/low-voltage detection reset, and a reset when in sub clock mode, sub-PLL clock mode, or stop mode, the CPU performs mode fetch after the main clock oscillation stabilization wait time has elapsed. If the external reset input is not cleared after the oscillation stabilization wait time has elapsed, the CPU performs mode fetch after the external reset input is cleared.

■ Effect of Reset on RAM Contents

When a reset occurs, the CPU halts the operation of the command currently being executed, and enters the reset status. During RAM access execution, however, RAM access protection causes an internal reset signal to be generated in synchronization with the machine clock, after RAM access has ended.This function prevents a word-data write operation from being cut off by a reset after one byte.

■ Pin State During a Reset

When a reset occurs, all of the I/O ports and peripheral resource pins remain in a high impedance state until setup is performed by software after the reset is released.

Note:

Connect a pull-up resistor to those pins which remain at high impedance during a reset to prevent the devices the pins from malfunctioning.

See "APPENDIX D Pin Status of MB95150/M series" for details about the states of all pins during a reset.

7.2 Reset Source Register (RSRR)

The reset source register indicates the source or factor causing a reset that has been generated.

■ Configuration of Reset Source Register (RSRR)

Figure 7.2-1 Reset Source Register (RSRR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0009_H	–	–	CSVR	EXTS	WDTR	PONR	HWR	SWR	$XXXXXXXX_B$
	R0/WX	R0/WX	R,W	R,W	R,W	R,W	R,W	R,W	

SWR	Software reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is software reset	

HWR	Hardware reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is hardware reset	

PONR	Power-on reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is power-on reset	

WDTR	Watchdog reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is watchdog reset	

EXTS	External reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is external reset	

CSVR	Clock supervisor reset flag bit	
	Read	Write
0	–	Writing sets the bit to "0".
1	Factor is clock supervisor reset	

R, W : Readable/writable
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined
X : Indeterminate

Table 7.2-1 Functions of Bits in Reset Source Register (RSRR)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". These bits are read-only. Writing has no effect on operation.
bit5	CSVR: Clock supervisor reset flag bit	This bit is set to "1" to indicate that a clock supervisor reset has occurred. Otherwise, the bit retains the value existing before the clock supervisor reset occurred. • Read or write access (0 or 1) to this bit sets it to "0". • The bit value is always "0" in product types that do not have the clock supervisor function. Writing has no effect on the operation.
bit4	EXTS: External reset flag bit	This bit is set to "1" to indicate that an external reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. • Read or write access (0 or 1) to this bit sets it to "0".
bit3	WDTR: watchdog reset flag bit	This bit is set to "1" to indicate that an watchdog reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. • Read or write access (0 or 1) to this bit sets it to "0".
bit2	PONR: Power-on reset flag bit	This bit is set to "1" to indicate that a power-on reset or low-voltage detection reset (option) has occurred. Otherwise, the bit retains the value existing before the reset occurred. • The low-voltage detection reset function is provided for specific models. • Read or write access (0 or 1) to this bit sets it to "0".
bit1	HWR: Hardware reset flag bit	This bit is set to "1" to indicate that a reset other than a software reset has occurred. When any of bits 2 to 5 is set to "1", therefore, this bit is set to "1" as well. Otherwise, the bit retains the value existing before the reset occurred. • Read or write access (0 or 1) to this bit sets it to "0".
bit0	SWR: Software reset flag bit	This bit is set to "1" to indicate that a software reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. • Read or write access (0 or 1) to this bit or a power-on reset sets it to "0".

Note:

Reading the reset source register clears its contents. To use the reset source register for calculation, therefore, you should move the contents of the register to RAM in advance.

■ Status of Reset Source Register (RSRR)

Table 7.2-2 Status of Reset Source Register

Reset Sources	–	–	CSVR	EXTS	WDTR	PONR	HWR	SWR
Power-on reset/ low-voltage detection reset	–	–	×	×	×	1	1	0
Software reset	–	–	Δ	Δ	Δ	Δ	Δ	1
Watchdog reset	–	–	Δ	Δ	1	Δ	1	Δ
External reset	–	–	Δ	1	Δ	Δ	1	Δ
Clock supervisor reset	–	–	1	Δ	Δ	Δ	1	Δ

1 : Flag set

Δ : Previous state saved

× : Undefined

CSVR : This bit is set to "1" to indicate that a clock supervisor reset has occurred (Always "0" if there is no clock supervisor option)

EXTS : This bit is set to "1" to indicate that an external reset has occurred.

WDTR : This bit is set to "1" to indicate that a watchdog reset has occurred.

PONR : This bit is set to "1" to indicate that a power-on reset or low-voltage detection reset (option) has occurred.

HWR : The bit value "1" indicates that a reset source occurs from either CSVR, EXTS, WDTR, or PONR.

SWR : This bit is set to "1" to indicate that a software reset has occurred.

7.3 Notes on Using Reset

This section explains the notes on using Reset.

■ Notes on Using Reset

● Initialization of the main clock stop detection bit of clock supervisor

The main clock stop detection bit (CSVCR:MM) of clock supervisor is initialized only by power-on reset and external reset.

The bit is not initialized by the watchdog timer reset/software reset/clock supervisor reset.Therefore, if one of these resets is issued, the CR clock mode continues.

● Initialization of register and bit by reset source

Some registers and bits are not initialized by reset source.

For the reset source register (RSRR), which of the bit is initialized depends on the reset source.

- The main clock stop detection bit (CSVCR:MM) of clock supervisor is initialized only by power-on reset and external reset.
- The CR oscillation enable bit (CSVCR:RCE) of clock supervisor is initialized only by power-on reset/ external reset.
- The main clock monitoring enable bit (CSVCR:MSVE) of clock supervisor is initialized only by power-on reset.
- The oscillation stabilization wait time setting register (WATR) of clock control block is initialized only by power-on reset.

CHAPTER 8
INTERRUPTS

This section explains the interrupts.

Code: CM26-00105-1E

8.1 Interrupts

This section explains the interrupts.

■ Overview of Interrupts

The F^2MC-8FX family has 24 interrupt request input lines corresponding to peripheral resources, for each of which an interrupt level can be set independently.

When a peripheral resource generates an interrupt request, the interrupt request is output to the interrupt controller.The interrupt controller checks the interrupt level of that interrupt request and then passes the occurrence of the interrupt to the CPU. The CPU services the interrupt according to the interrupt acceptance status. Interrupt requests also release the device from standby mode to resume instruction execution.

■ Interrupt Requests from Peripheral Resources

Table 8.1-1 shows the interrupt requests corresponding to the peripheral resources. When an interrupt is accepted, a branch to the interrupt service routine takes place with the content of the interrupt vector table address corresponding to the interrupt request as the address of the branch destination.

The priority for each interrupt request can be set to one of four levels using the interrupt level setting registers (ILR0 to ILR5).

If another interrupt request with the same or lower level occurs during execution of the interrupt service routine, the interrupt is processed after the current interrupt handler routine completes. If interrupt requests of the same level occur at the same time, IRQ0 is assigned the highest priority.

Table 8.1-1 Interrupt Requests and Interrupt Vectors

Interrupt request	Vector table address		Bit name of interrupt level setting register	Priority for equal-level Interrupt requests (generated simultaneously)
	Upper	Lower		
(Reset vector)	$FFFE_H$	$FFFF_H$	-	High
(Mode data)	-	$FFFD_H$	-	↑
IRQ0	$FFFA_H$	$FFFB_H$	L00 [1:0]	
IRQ1	$FFF8_H$	$FFF9_H$	L01 [1:0]	
IRQ2	$FFF6_H$	$FFF7_H$	L02 [1:0]	
IRQ3	$FFF4_H$	$FFF5_H$	L03 [1:0]	
IRQ4	$FFF2_H$	$FFF3_H$	L04 [1:0]	
IRQ5	$FFF0_H$	$FFF1_H$	L05 [1:0]	
IRQ6	$FFEE_H$	$FFEF_H$	L06 [1:0]	
IRQ7	$FFEC_H$	$FFED_H$	L07 [1:0]	
IRQ8	$FFEA_H$	$FFEB_H$	L08 [1:0]	
IRQ9	$FFE8_H$	$FFE9_H$	L09 [1:0]	
IRQ10	$FFE6_H$	$FFE7_H$	L10 [1:0]	
IRQ11	$FFE4_H$	$FFE5_H$	L11 [1:0]	
IRQ12	$FFE2_H$	$FFE3_H$	L12 [1:0]	
IRQ13	$FFE0_H$	$FFE1_H$	L13 [1:0]	
IRQ14	$FFDE_H$	$FFDF_H$	L14 [1:0]	
IRQ15	$FFDC_H$	$FFDD_H$	L15 [1:0]	
IRQ16	$FFDA_H$	$FFDB_H$	L16 [1:0]	
IRQ17	$FFD8_H$	$FFD9_H$	L17 [1:0]	
IRQ18	$FFD6_H$	$FFD7_H$	L18 [1:0]	
IRQ19	$FFD4_H$	$FFD5_H$	L19 [1:0]	
IRQ20	$FFD2_H$	$FFD3_H$	L20 [1:0]	
IRQ21	$FFD0_H$	$FFD1_H$	L21 [1:0]	
IRQ22	$FFCE_H$	$FFCF_H$	L22 [1:0]	↓
IRQ23	$FFCC_H$	$FFCD_H$	L23 [1:0]	Low

For interrupt sources, see "APPENDIX B Table of Interrupt Causes".

8.1.1 Interrupt Level Setting Registers (ILR0 to ILR5)

The interrupt level setting registers (ILR0 to ILR5) contain 24 pairs of bits assigned for the interrupt requests from different peripheral resources. Each pair of bits (interrupt level setting bits as two-bit data) sets each interrupt level.

■ Configuration of Interrupt Level Setting Registers (ILR0 to ILR5)

Figure 8.1-1 Configuration of Interrupt Level Setting Registers

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		Initial value
ILR0	00079_H	L03	[1:0]	L02	[1:0]	L01	[1:0]	L00	[1:0]	R/W	11111111_B
ILR1	$0007A_H$	L07	[1:0]	L06	[1:0]	L05	[1:0]	L04	[1:0]	R/W	11111111_B
ILR2	$0007B_H$	L11	[1:0]	L10	[1:0]	L09	[1:0]	L08	[1:0]	R/W	11111111_B
ILR3	$0007C_H$	L15	[1:0]	L14	[1:0]	L13	[1:0]	L12	[1:0]	R/W	11111111_B
ILR4	$0007D_H$	L19	[1:0]	L18	[1:0]	L17	[1:0]	L16	[1:0]	R/W	11111111_B
ILR5	$0007E_H$	L23	[1:0]	L22	[1:0]	L21	[1:0]	L20	[1:0]	R/W	11111111_B

The interrupt level setting registers assign each pair of bits for a different interrupt request. The values of interrupt level setting bits in these registers specify interrupt service priorities (interrupt levels 0 to 3).

The interrupt level setting bits are compared with the interrupt level bits in the condition code register (CCR: IL1, IL0).

When interrupt level 3 is set for an interrupt request, the CPU ignores the interrupt request.

Table 8.1-2 shows the relationships between interrupt level setting bits and interrupt levels.

Table 8.1-2 Relationships Between Interrupt Level Setting Bits and Interrupt Levels

LXX[1:0]	Interrupt Level	Priority
00	0	High
01	1	↕
10	2	
11	3	Low (No interrupt)

XX:00 to 23 Corresponding interrupt number

During execution of a main program, the interrupt level bits in the condition code register (CCR: IL1, IL0) are usually "11_B".

8.1.2 Interrupt Processing Steps

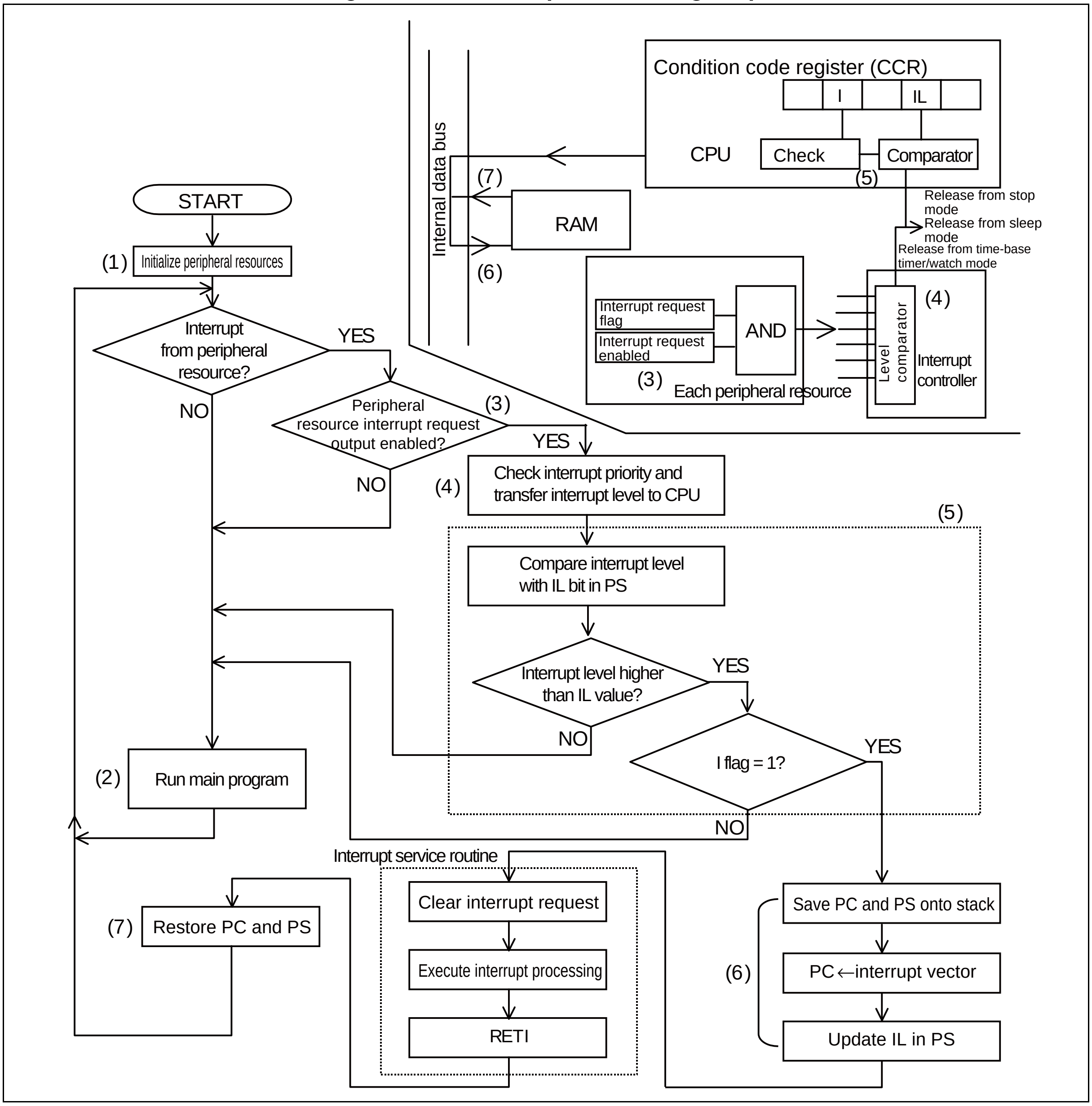
When an interrupt request is generated by a peripheral resource, the interrupt controller passes the interrupt level to the CPU. When the CPU is ready to accept interrupts, it temporarily halts the program currently being executed and executes an interrupt service routine.

■ Interrupt Processing

The procedure of processing an interrupt takes the following steps: the generation of an interrupt resource in a peripheral resource, the execution of the main program, the setting of the interrupt request flag bit, the evaluation of the interrupt request enable bit, the evaluation of interrupt level (ILR0 to ILR5 and CCR:IL1, IL0), the checking for any equal-level interrupt request, and the evaluation of the interrupt enable flag (CCR:I).

Figure 8.1-2 illustrates the steps to take for interrupt processing.

Figure 8.1-2 Interrupt Processing Steps



(1) Any interrupt request is disabled immediately after a reset. In the peripheral resource initialization program, initialize those peripheral resources which generate interrupts and set their interrupt levels in their respective interrupt level setting registers (ILR0 to ILR5) before starting operating the peripheral resources. The interrupt level can be set to 0, 1, 2, or 3. Level 0 is given the highest priority, and level 1 the second highest. Setting level 3 for a peripheral resource disables interrupts from that resource.

(2) Execute the main program (or the interrupt processing routine for nested interrupts).

(3) When an interrupt is triggered in a peripheral resource, the interrupt request flag bit of the peripheral resource is set to "1". If the interrupt request enable bit of the peripheral resource has been set to enable interrupts, the interrupt request is then output to the interrupt controller.

(4) The interrupt controller always monitors interrupt requests from individual peripheral resources and transfers the highest-priority interrupt level to the CPU among the interrupt levels of the currently generated interrupt requests. The relative priority to be assigned if another request with the same interrupt level occurs simultaneously is also determined at this time.

(5) If the received interrupt level or priority is lower than the level set in the interrupt level bits in the condition code register (CCR: IL1, IL0), the CPU checks the content of the interrupt enable flag (CCR:I) and, if interrupts are enabled (CCR:I = 1), accepts the interrupt.

(6) The CPU pushes the contents of the program counter (PC) and program status (PS) register onto the stack, fetches the start address of the interrupt processing routine from the corresponding interrupt vector table, changes the value of the interrupt level bits in the condition code register (CCR: IL1, IL0) to the value of the received interrupt level, then starts the execution of the interrupt processing routine.

(7) Finally, the CPU uses the RETI instruction to restore the program counter (PC) and program status (PS) values from the stack and resumes execution from the instruction that follows the instruction executed prior to the interrupt.

Note:

The interrupt request flag bits of peripheral resources are not automatically cleared to "0" after an interrupt request is accepted. The bits must therefore be cleared to "0" by a program (by writing "0" to the interrupt request flag bit) in the interrupt processing routine.

An interrupt causes the device to recover from standby mode (low power consumption mode).For details, see Section "6.8 Operations in Low-power Consumption Modes (Standby Modes)".

8.1.3 Nested Interrupts

You can set different interrupt levels for two or more interrupt requests from peripheral resources in the interrupt level setting registers (ILR0 to ILR5) to process the nested interrupts.

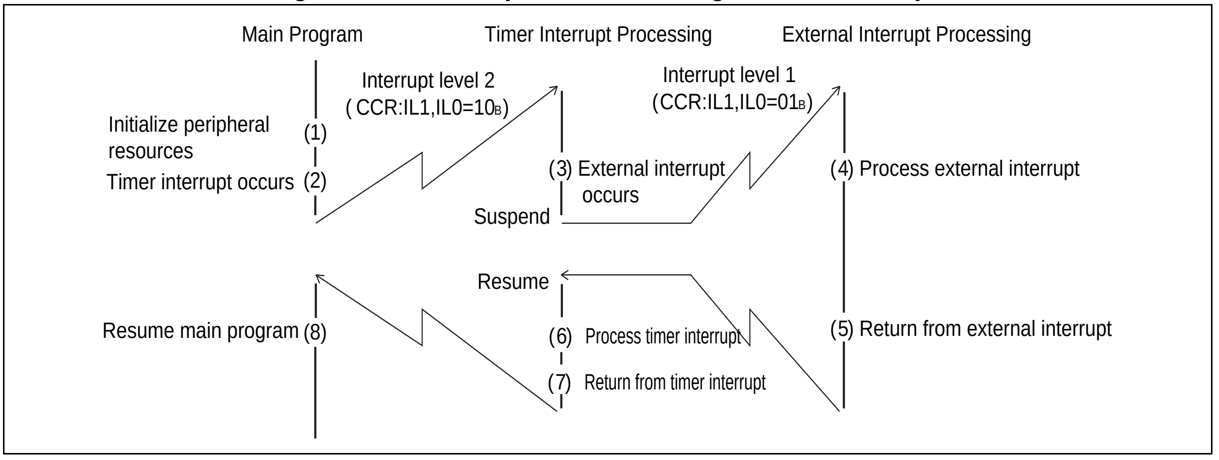
■ Nested Interrupts

If an interrupt request of higher-priority interrupt level occurs while an interrupt service routine is being executed, the CPU halts processing of the current interrupt and accepts the higher-priority interrupt request. The interrupt level can be set to 0 to 3. If it is set to 3, the CPU will accept no interrupt request.

[Example: Nested interrupts]

To assign higher priority to external interrupts over timer interrupts as an example of processing nested-interrupts, set the timer interrupt and external interrupt levels to 2 and 1, respectively. If an external interrupt occurs while a timer interrupt is being processed with these settings in use, the interrupts are processed as shown in Figure 8.1-3.

Figure 8.1-3 Example of Processing Nested Interrupts



- While a timer interrupt is being processed, the interrupt level bits in the condition code register (CCR: IL1, IL0) hold the same value as that of the interrupt level setting registers (ILR0 to ILR5) corresponding to the current timer interrupt (level 2 in this example). If an interrupt request with a higher-priority interrupt level (level 1 in the example) occurs, the higher-priority interrupt is processed preferentially.
- To temporarily disable nested interrupt processing while a timer interrupt is being processed, set the interrupt enable flag in the condition code register to disable interrupts (CCR:I = 0) or set the interrupt level bits (CCR: IL1, IL0) to "00_B".
- Executing the interrupt return instruction (RETI) after interrupt processing is completed restores the program counter (PC) and program status (PS) values saved in a stack and resumes the processing of the interrupted program.Restoring the program status (PS) also restores the condition code register (CCR) to its value existing prior to the interrupt.

8.1.4 Interrupt Processing Time

The time between an interrupt request being generated and control being passed to the interrupt processing routine is equal to the sum of the time until the currently executing instruction completes and the interrupt handling time (time required to initiate interrupt processing). This time consists of a maximum of 26 machine clock cycles.

■ Interrupt Processing Time

The interrupt request sampling wait time and interrupt handling time intervene between the occurrence and acceptance of an interrupt request and the execution of the relevant interrupt service routine.

● Interrupt request sampling wait time

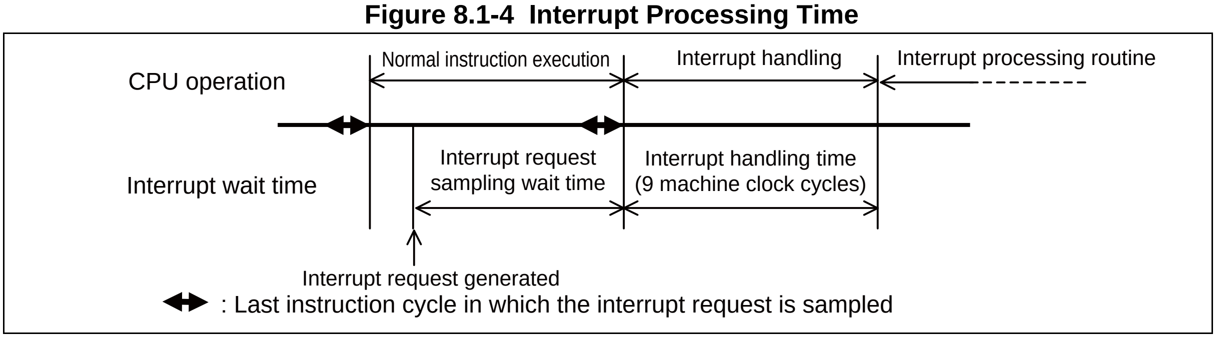
Whether an interrupt request has occurred is determined through the sampling of the interrupt request during the last cycle of each instruction. The CPU cannot therefore recognize interrupt requests during the execution of each instruction. The maximum length of this delay occurs if the interrupt request is generated immediately after the DIVU instruction requiring the longest instruction cycle (17 machine clock cycles) starts executing.

● Interrupt handling time

After receiving an interrupt, the CPU requires 9 machine clock cycles to perform the following interrupt processing setup:

- Saves the program counter (PC) and program status (PS) values.
- Sets the PC to the start address (interrupt vector) of interrupt service routine.
- Updates the interrupt level bits (PS:CCR:IL1, IL0) in the program status (PS) register.

Figure 8.1-4 Interrupt Processing Time



When an interrupt request is generated immediately after the beginning of execution of the DIVU instruction requiring the longest execution cycle (17 machine clock cycles), it takes an interrupt processing time of 17+9=26 machine clock cycles.

The machine clock changes depending on the clock mode and main clock speed switching (gear function). For details, see "CHAPTER 6 CLOCK CONTROLLER".

8.1.5 Stack Operations During Interrupt Processing

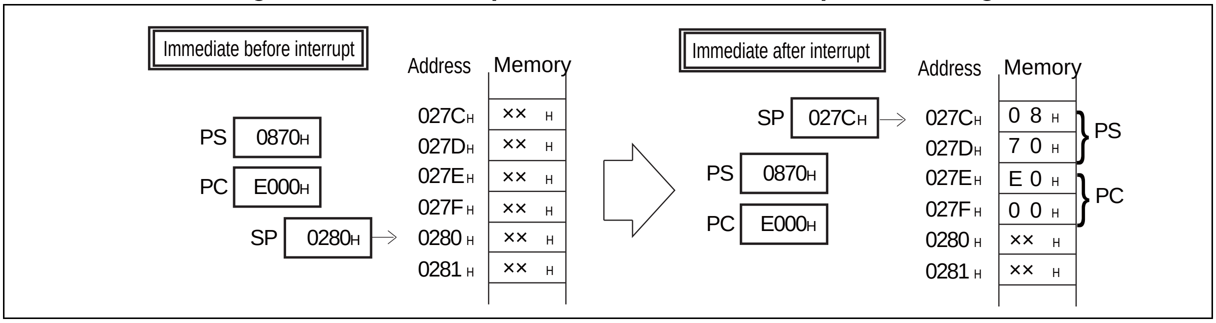
This section describes how registers are saved and restored during interrupt processing.

■ Stack Operation at the Start of Interrupt Processing

Once the CPU accepts an interrupt, it automatically saves the current program counter (PC) and program status (PS) values onto a stack.

Figure 8.1-5 shows how the stack is used at the start of interrupt processing.

Figure 8.1-5 Stack Operation at Start of Interrupt Processing



■ Stack Operation upon Returning from Interrupt

When the interrupt return instruction (RETI) is executed to end interrupt processing, the program status (PS) and then the program counter (PC) are restored from the stack, in the reverse order from which they were saved to the stack when interrupt processing started. This restores the PS and PC values to their states prior to starting interrupt processing.

Note:

As the accumulator (A) and temporary accumulator (T) are not saved onto the stack automatically, use the PUSHW and POPW instructions to save and restore the A and T values.

8.1.6 Interrupt Processing Stack Area

The stack area in RAM is used for interrupt processing. The stack pointer (SP) contains the start address of the stack area.

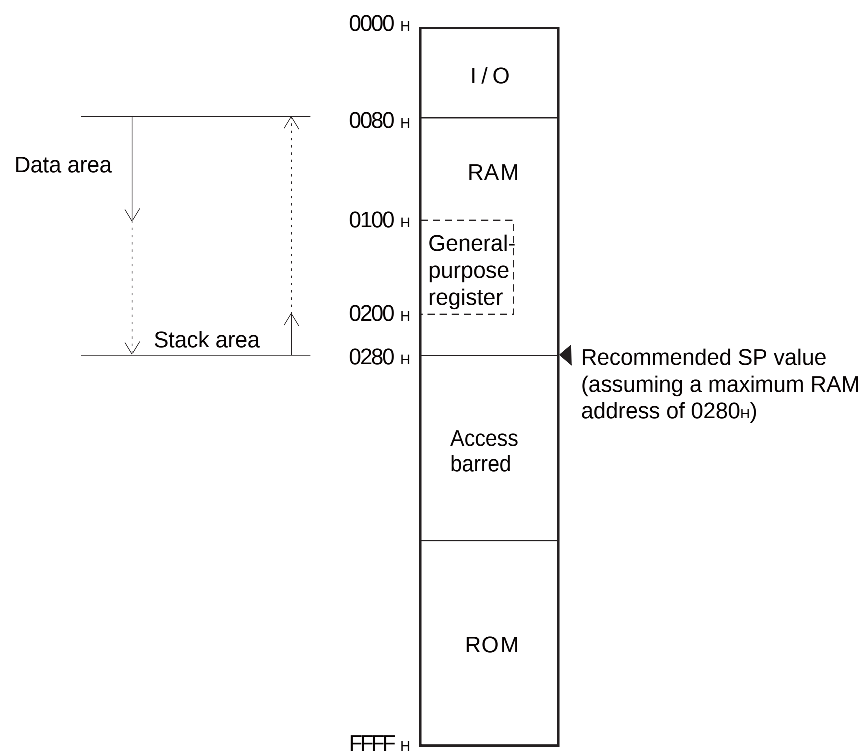
■ Interrupt Processing Stack Area

The stack area is also used to save and restore the program counter (PC) when subroutine call (CALL) or vector call (CALLV) instructions are executed and to temporarily save and restore the registers via the PUSHW and POPW instructions.

- The stack area is located in RAM together with the data area.
- It is advisable to initialize the stack pointer (SP) to the maximum RAM address and allocate data areas starting from the minimum RAM address.

Figure 8.1-6 shows an example of setting the stack area.

Figure 8.1-6 Setting Example of Interrupt Processing Stack Area



Note:

The stack area is allocated in descending order of addresses for interrupts, subroutine calls, and the PUSHW instruction; it is deallocated in ascending order of addresses for return (PETI, RET) and POPW instructions. When the stack area address used decreases for nested interrupts or subroutines, prevent the stack area from overlapping the data area or general-purpose register area containing other data.

CHAPTER 9

I/O PORT

This chapter describes the functions and operations of the I/O ports.

9.1 Overview of I/O Ports

I/O ports are used to control general-purpose I/O pins.

■ Overview of I/O Ports

The I/O port has functions to output data from the CPU and load inputted signals into the CPU, via the port data register (PDR). It is also possible to set the input/output direction of the I/O pins as desired at the bit level, via the port direction register (DDR).

Table 9.1-1 lists the registers for each port.

Table 9.1-1 Each Port Registers

Register name		Read/Write	Initial value
Port 0 data register	(PDR0)	R, RM/W	00000000_B
Port 0 direction register	(DDR0)	R/W	00000000_B
Port 1 data register	(PDR1)	R, RM/W	00000000_B
Port 1 direction register	(DDR1)	R/W	00000000_B
Port 6 data register	(PDR6)	R, RM/W	00000000_B
Port 6 direction register	(DDR6)	R/W	00000000_B
Port 9 data register	(PDR9)	R, RM/W	00000000_B
Port 9 direction register	(DDR9)	R/W	00000000_B
Port A data register	(PDRA)	R, RM/W	00000000_B
Port A direction register	(DDRA)	R/W	00000000_B
Port B data register	(PDRB)	R, RM/W	00000000_B
Port B direction register	(DDRB)	R/W	00000000_B
Port G data register	(PDRG)	R, RM/W	00000000_B
Port G direction register	(DDRG)	R/W	00000000_B
Port 0 pull-up control register	(PUL0)	R/W	00000000_B
Port 1 pull-up control register	(PUL1)	R/W	00000000_B
Port G pull-up control register	(PULG)	R/W	00000000_B
A/D input disable register lower	(AIDRL)	R/W	00000000_B
Input level selection register	(ILSR)	R/W	00000000_B
Input level selection register 2*	(ILSR2)	R/W	00000000_B

R/W: Readable/Writable (Read value is the same as the write value.)

R, RM/W: Readable/Writable (Read value is different from write value, write value is read by read-modify-write (RMW) instruction.)

*: Only for 5V products, it is an effective register.

MB95150/M Series

9.2 Port 0

Port 0 is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port 0 Configuration

Port 0 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 0 data register (PDR0)
- Port 0 direction register (DDR0)
- Port 0 pull-up control register (PUL0)
- A/D input disable register low (AIDRL)
- Input level selection register 2 (ILSR2)

■ Port 0 Pins

Port 0 has eight I/O pins.

Table 9.2-1 lists the port 0 pins.

Table 9.2-1 Port 0 Pins

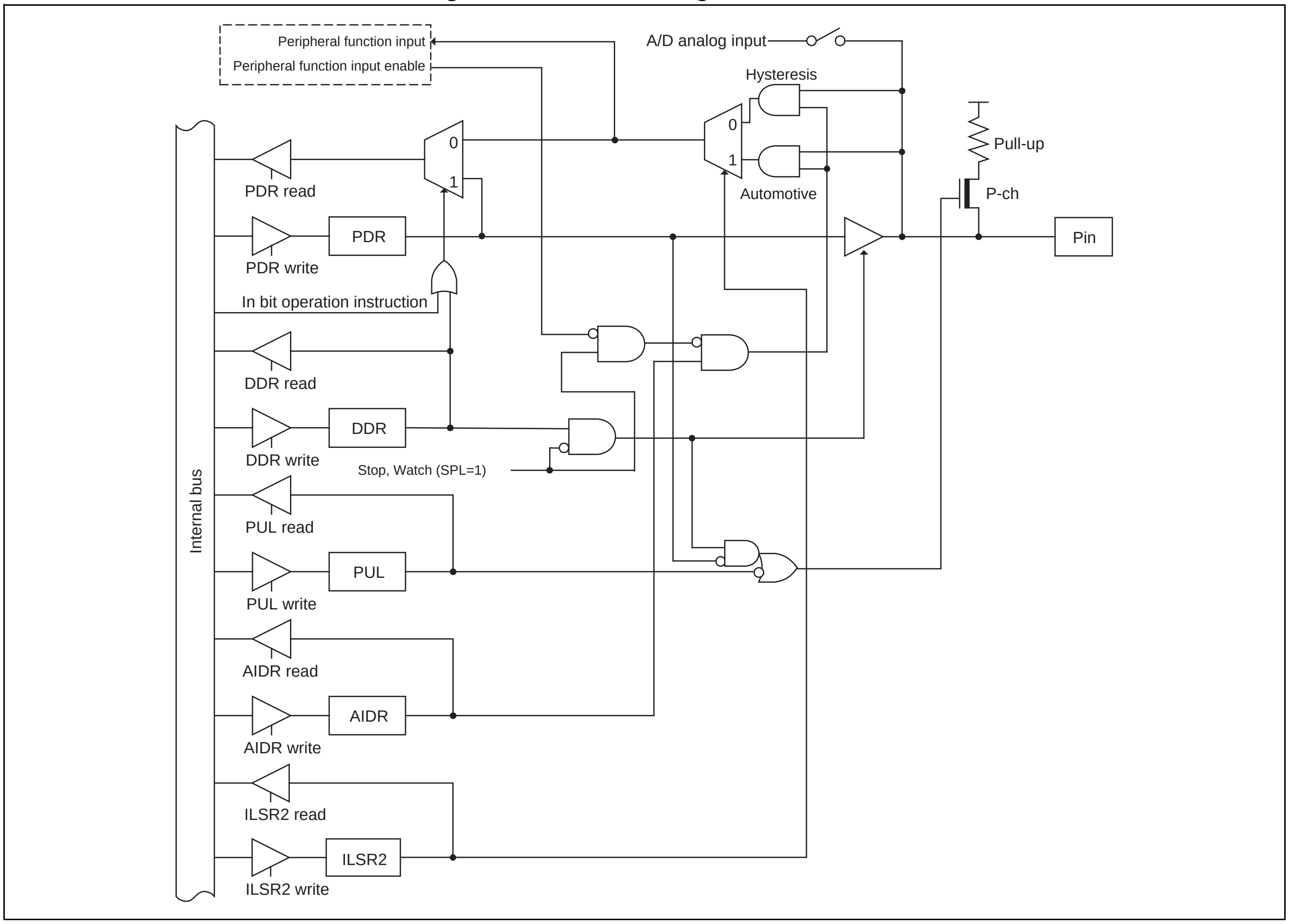
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P00/INT00/ AN00	P00 general-purpose I/O	AN00 analog input INT00 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P01/INT01/ AN01	P01 general-purpose I/O	AN01 analog input INT01 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P02/INT02/ AN02	P02 general-purpose I/O	AN02 analog input INT02 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P03/INT03/ AN03	P03 general-purpose I/O	AN03 analog input INT03 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P04/INT04/ AN04	P04 general-purpose I/O	AN04 analog input INT04 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P05/INT05/ AN05	P05 general-purpose I/O	AN05 analog input INT05 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P06/INT06/ AN06	P06 general-purpose I/O	AN06 analog input INT06 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍
P07/INT07/ AN07	P07 general-purpose I/O	AN07 analog input INT07 external interrupt input	Hysteresis/ automotive/analog	CMOS	-	❍

OD: Open drain, PU: Pull-up

*: For 5V products, the hysteresis input can be switched to an automotive input. It becomes a hysteresis input besides.

■ Block Diagram of Port 0

Figure 9.2-1 Block Diagram of Port 0



MB95150/M Series

9.2.1 Port 0 Registers

This section describes the port 0 registers.

■ Port 0 Register Function

Table 9.2-2 lists the port 0 register functions.

Table 9.2-2 Port 0 Register Function

Register name	Data	Read	Read read-modify-write	Write
PDR0	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR0	0	Port input enabled		
	1	Port output enabled		
PUL0	0	Pull-up disable		
	1	Pull-up enabled		
AIDRL	0	Analog input enabled		
	1	Port input enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.2-3 lists the correspondence between port 0 pins and each register bit.

Table 9.2-3 Correspondence Between Registers and Pins for Port 0

	Correspondence between related register bits and pins							
Pin name	P07	P06	P05	P04	P03	P02	P01	P00
PDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR0								
PUL0								
AIDRL								
ILSR2*	bit0							

*:Only for 5V products, it is an effective register.

9.2.2 Operations of Port 0

This section describes the operations of port 0.

■ Operations of Port 0

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.
  Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

Operation of the external interrupt input pin

- Set the DDR register bit, which is corresponding to the external interrupt input pin, to "0".
- Pin values are continuously input to the external interrupt circuit. When using the pin for a function other than an interrupt, you must disable the corresponding external interrupt.

Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit0 of the ILSR2 register to "1" changes the port 0 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit0 of the ILSR2 register is "0".
- Only modify the port 0 input level setting when the peripheral function inputs are halted.

Table 9.2-4 shows the pin states of the port 0.

Table 9.2-4 Pin State of Port 0

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff (If external interrupts are enabled, the external interrupt can be input.)	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

9.3 Port 1

Port 1 is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port 1 Configuration

Port 1 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 1 data register (PDR1)
- Port 1 direction register (DDR1)
- Port 1 pull-up control register (PUL1)
- Input level selection register (ILSR)
- Input level selection register 2 (ILSR2)

■ Port 1 Pins

Port 1 has five I/O pins.

Table 9.3-1 lists the port 1 pins.

Table 9.3-1 Port 1 Pins

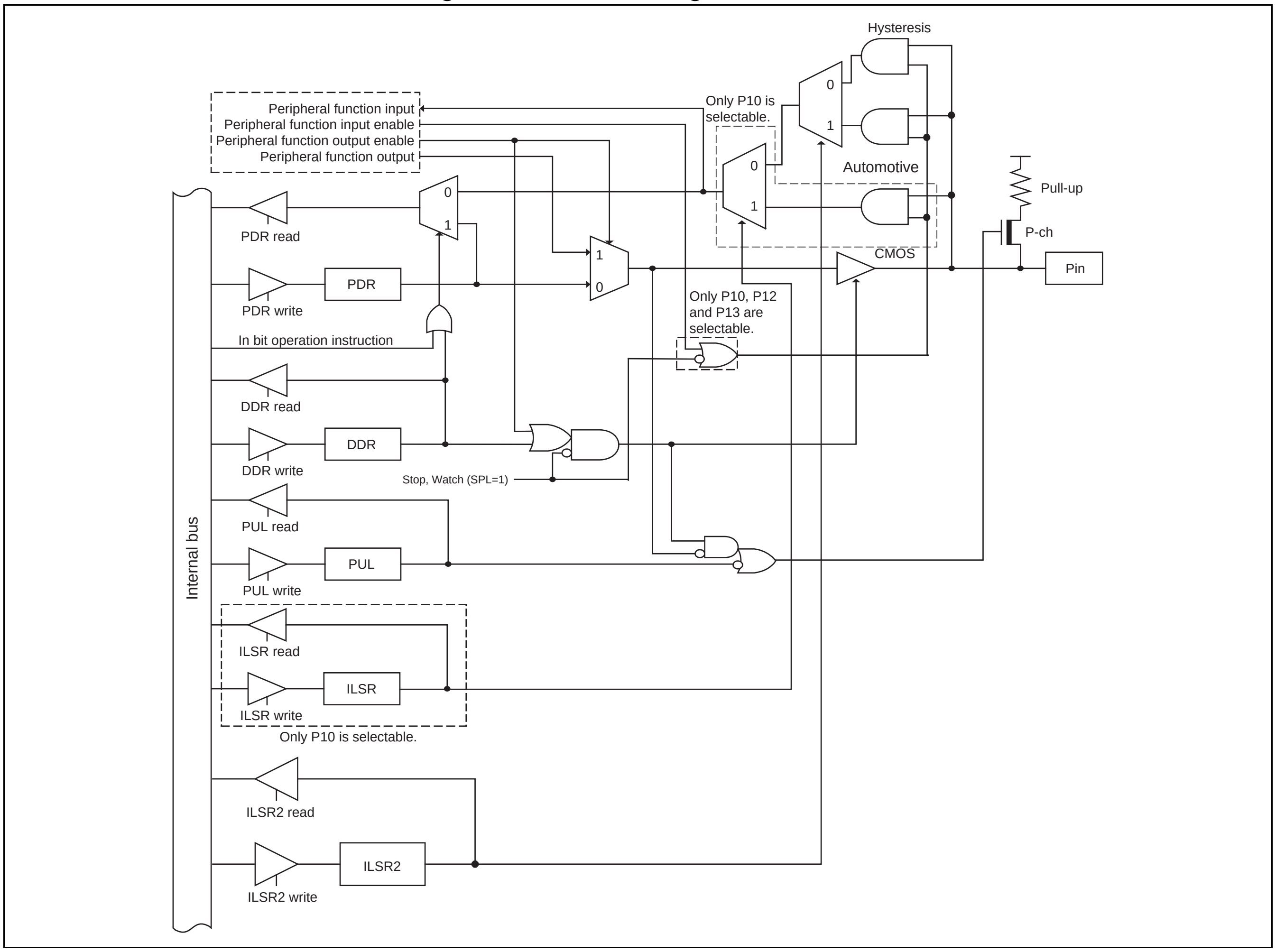
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P10/UI0	P10 general-purpose I/O	UI0 UART/SIO ch.0 data input	Hysteresis/CMOS/ automotive	CMOS	-	❍
P11/UO0	P11 general-purpose I/O	UO0 UART/SIO ch.0 data output	Hysteresis/automotive	CMOS	-	❍
P12/UCK0	P12 general-purpose I/O	UO0 UART/SIO ch.0 clock I/O	Hysteresis/automotive	CMOS	-	❍
P13/ TRG0/ADTG	P13 general-purpose I/O	TRG0 16-bit PPG ch.0 trigger input	Hysteresis/automotive	CMOS	-	❍
		ADTG A/D trigger activation input				
P14/PPG0	P14 general-purpose I/O	PPG0 16-bit PPG ch.0 output	Hysteresis/automotive	CMOS	-	❍

OD: Open drain, PU: Pull-up

*: Only for 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input or CMOS input besides.

■ Block Diagram of Port 1

Figure 9.3-1 Block Diagram of Port 1



9.3.1 Port 1 Registers

This section describes the port 1 registers.

■ Port 1 Register Function

Table 9.3-2 lists the port 1 register functions.

Table 9.3-2 Port 1 Register Function

Register name	Data	Read	Read read-modify-write	Write
PDR1	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR1	0	Port input enabled		
	1	Port output enabled		
PUL1	0	Pull-up disabled		
	1	Pull-up enabled		
ILSR	0	Hysteresis input level selection		
	1	CMOS input level selection		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.3-3 lists the correspondence between port 1 pins and each register bit.

Table 9.3-3 Correspondence Between Registers and Pins for Port 1

	Correspondence between related register bits and pins							
Pin name	-		-	P14	P13	P12	P11	P10
PDR1	-	-	-	bit4	bit3	bit2	bit1	bit0
DDR1								
PUL1								
ILSR	-		-	-	-	-	-	bit0
ILSR2*	-		-	bit1				

*: Only for 5V products, it is an effective register.

9.3.2 Operations of Port 1

This section describes the operations of port 1.

■ Operations of Port 1

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.
  Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input of P10/UI0, P12/UCK0 and P13/TRG0/ADTG port is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", set the DDR register bit, the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

● Operation of the input level selection register

- Writing "1" to the bit0 of ILSR register changes only P10 from the hysteresis input level to the CMOS input level. When the bit0 of ILSR register is "0", it should be the hysteresis input level.
- For pins other than P10, the CMOS input level cannot be selected; however, only the hysteresis input level or the automotive input level can.
- Make sure that the input level for P10 is changed during the peripheral function (UART/SIO) stopped.

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models. Setting bit1 of the ILSR2 register to "1" changes the port 1 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit1 of the ILSR2 register is "0".
- P10 only uses the automotive input level when bit0 of the ILSR register is "0". In the case of P10 only, setting "1" to bit0 of the ILSR register has priority over ILSR2.
- Only modify the port 1 input level setting when the peripheral functions (UART/SIO) are halted.

Table 9.3-4 shows the pin states of the port 1.

Table 9.3-4 Pin State of Port 1

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z (the pull-up setting is enabled) Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

MB95150/M Series

9.4 Port 6

Port 6 is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port 6 Configuration

Port 6 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 6 data register (PDR6)
- Port 6 direction register (DDR6)
- Input level selection register (ILSR)
- Input level selection register 2 (ILSR2)

■ Port 6 Pins

Port 6 has eight I/O pins.

Table 9.4-1 lists the port 6 pins.

Table 9.4-1 Port 6 Pins

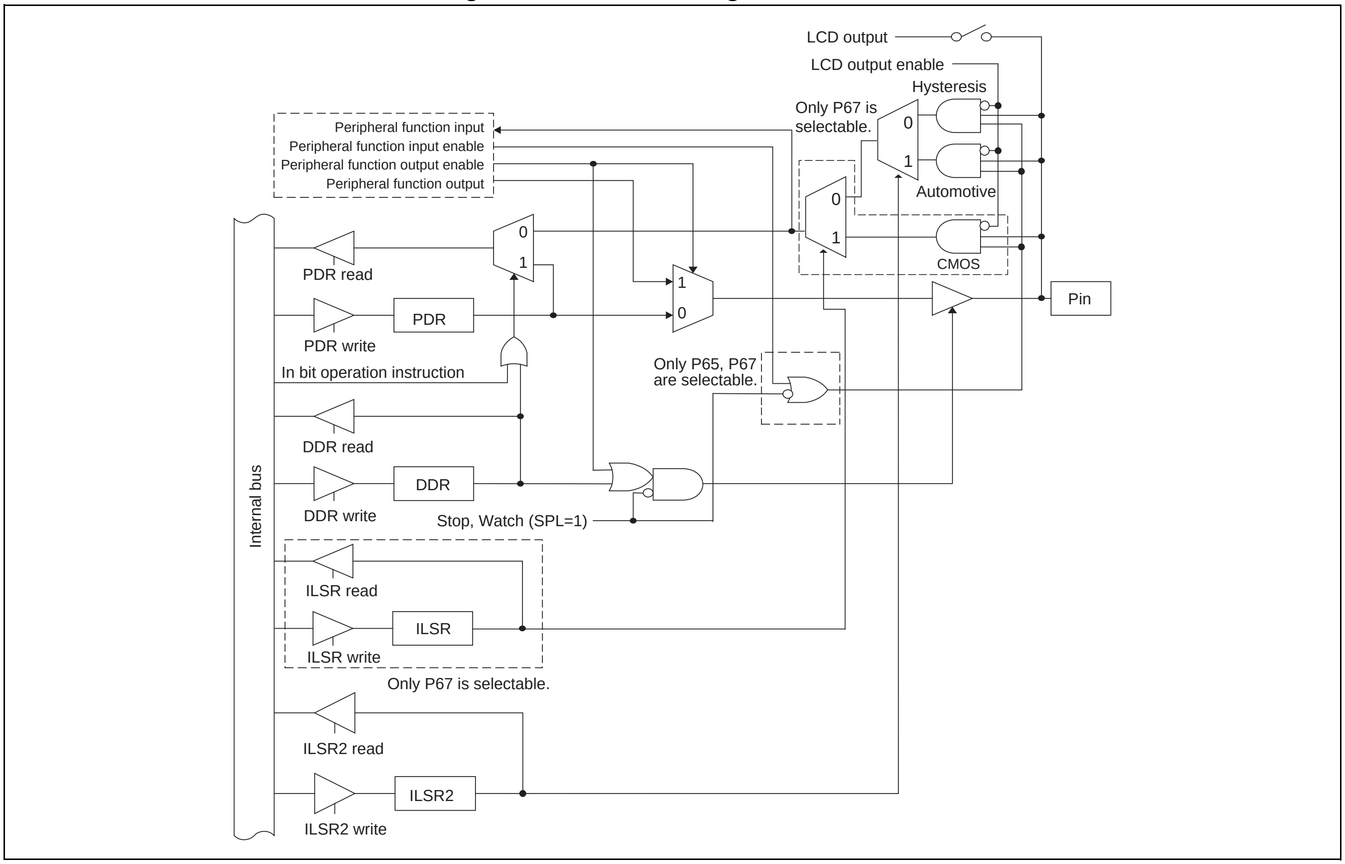
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P60/PPG10/ SEG08	P60 general-purpose I/O	PPG10 8/16-bit PPG1 ch.0 output SEG08 LCDC SEG08 output	Hysteresis/automotive	CMOS	-	-
P61/PPG11/ SEG09	P61 general-purpose I/O	PPG11 8/16-bit PPG1 ch.1 output SEG09 LCDC SEG09 output	Hysteresis/automotive	CMOS	-	-
P62/TO10/ SEG10	P62 general-purpose I/O	TO10 8/16-bit compound timer 10 output SEG10 LCDC SEG10 output	Hysteresis/automotive	CMOS	-	-
P63/TO11/ SEG11	P63 general-purpose I/O	TO11 8/16-bit compound timer 11 output SEG11 LCDC SEG11 output	Hysteresis/automotive	CMOS	-	-
P64/EC1/ SEG12	P64 general-purpose I/O	EC1 8/16-bit compound timer ch.1 clock input SEG12 LCDC SEG12 output	Hysteresis/automotive	CMOS	-	-
P65/SCK/ SEG13	P65 general-purpose I/O	LIN-UART clock I/O SEG13 LCDC SEG13 output	Hysteresis/automotive	CMOS	-	-
P66/SOT/ SEG14	P66 general-purpose I/O	LIN-UART data output SEG14 LCDC SEG14 output	Hysteresis/automotive	CMOS	-	-
P67/SIN/ SEG15	P67 general-purpose I/O	LIN-UART data input SEG15 LCDC SEG15 output	Hysteresis/CMOS/ automotive	CMOS	-	-

OD: Open drain, PU: Pull-up

*: For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

■ Block Diagram of Port 6

Figure 9.4-1 Block Diagram of Port 6



MB95150/M Series

9.4.1 Port 6 Registers

This section describes the port 6 registers.

■ Port 6 Register Function

Table 9.4-2 lists the port 6 register functions.

Table 9.4-2 Port 6 Register Function

Register name	Data	Read	Read read-modify-write	Write
PDR6	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR6	0	Port input enabled		
	1	Port output enabled		
ILSR	0	Hysteresis input level selection		
	1	CMOS input level selection		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.4-3 lists the correspondence between port 6 pins and each register bit.

Table 9.4-3 Correspondence Between Registers and Pins for Port 6

	Correspondence between related register bits and pins							
Pin name	P67	P66	P65	P64	P63	P62	P61	P60
PDR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR6								
ILSR	bit2	-	-	-	-	-	-	-
ILSR2*	bit2							

*: Only for 5V products, it is an effective register.

9.4.2 Operations of Port 6

This section describes the operations of port 6.

■ Operations of Port 6

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) command returns the PDR register value.

● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.
  Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input of P65/SCK and P67/SIN port is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

Operation of the input level selection register

- Setting "1" to the bit2 of ILSR register changes only P67 from the hysteresis input level to the CMOS input level. When the bit2 of ILSR register is "0", it should be the hysteresis input level.
- For pins other than P67, the CMOS input level cannot be selected. Only the hysteresis input level or automotive input level can be selected.
- Make sure that the input level for P67 is changed during the peripheral function (LIN-UART) stopped.

Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit2 of the ILSR2 register to "1" changes the port 6 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit2 of the ILSR2 register is "0".
- Only modify the port 6 input level setting when the peripheral function (LIN-UART) is halted.
- P67 only uses the automotive input level when bit2 of the ILSR register is "0". Setting "1" to bit2 of the ILSR register has priority over the ILSR2 register.

Table 9.4-4 shows the pin states of the port 6.

Table 9.4-4 Pin State of Port 6

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

9.5 Port 9

Port 9 is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port 9 Configuration

Port 9 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 9 data register (PDR9)
- Port 9 direction register (DDR9)
- Input level selection register 2 (ILSR2)

■ Port 9 Pins

Port 9 has six I/O pins.

Table 9.5-1 lists the port 9 pins.

Table 9.5-1 Port 9 Pin

Pin name	Function	Shared peripheral functions	I/O type			
			Input *	Output	OD	PU
P90/V3	P90 general-purpose I/O	V3 LCDC V3 pin	Hysteresis/ automotive	CMOS	-	-
P91/V2	P91 general-purpose I/O	V2 LCDC V2 pin	Hysteresis/ automotive	CMOS	-	-
P92/V1	P92 general-purpose I/O	V1 LCDC V1 pin	Hysteresis/ automotive	CMOS	-	-
P93/V0	P93 general-purpose I/O	V0 LCDC V0 pin	Hysteresis/ automotive	CMOS	-	-
P94	P94 general-purpose I/O	-	Hysteresis/ automotive	CMOS	-	-
P95	P95 general-purpose I/O	-	Hysteresis/ automotive	CMOS	-	-

OD: Open drain, PU: Pull-up

*: For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

■ Block Diagram of Port 9

Figure 9.5-1 Block Diagram of Port 9 (P90 to P93)

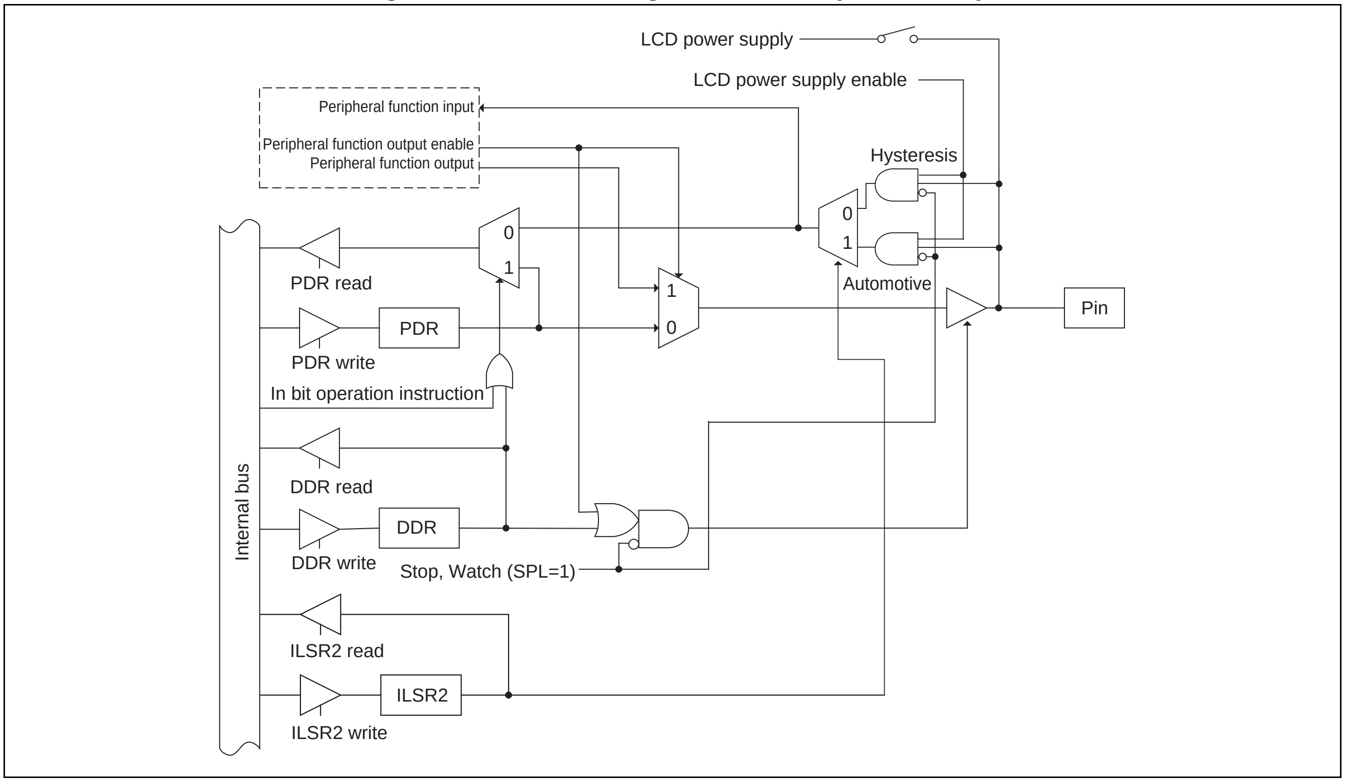
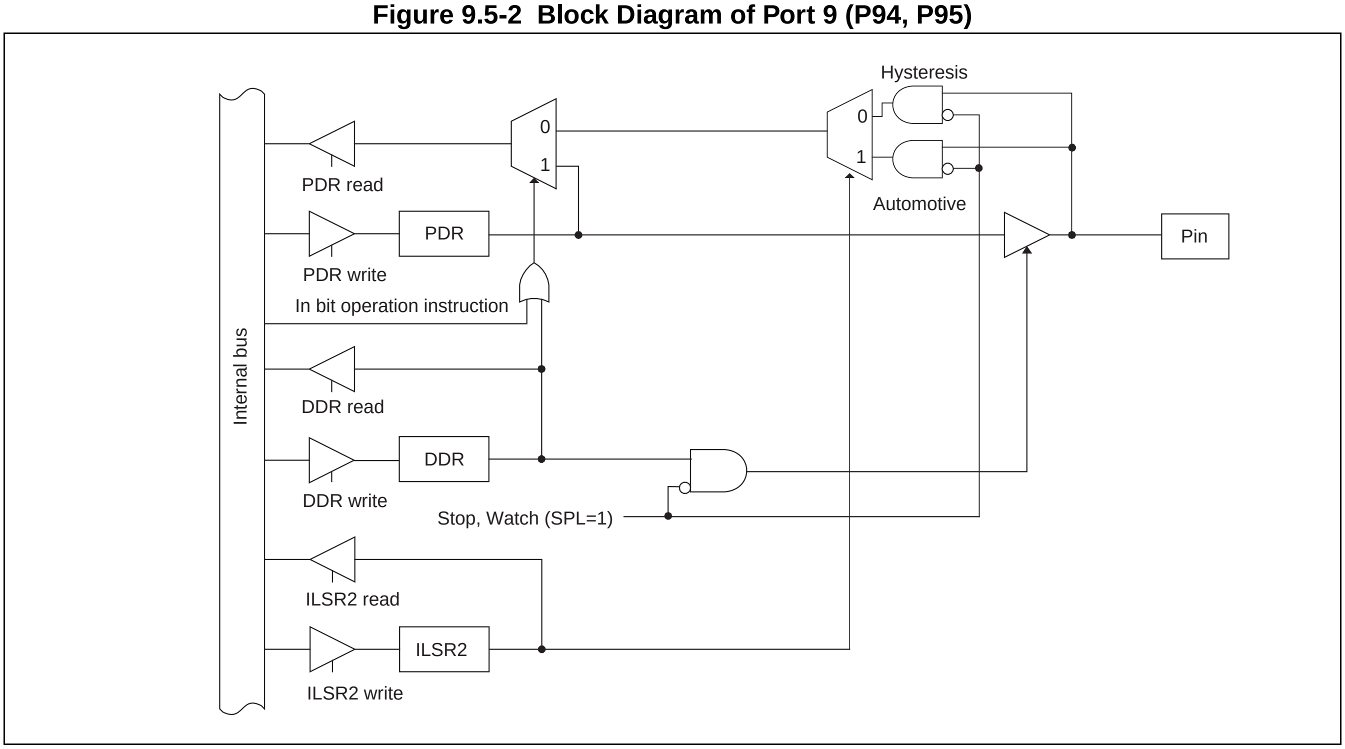


Figure 9.5-2 Block Diagram of Port 9 (P94, P95)



9.5.1 Port 9 Registers

This section describes the port 9 registers.

■ Port 9 Register Function

Table 9.5-2 lists the port 9 register functions.

Table 9.5-2 Port 9 Register Function

Register name	Data	Read	Read read-modify-write	Write
PDR9	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR9	0	Port input enabled		
	1	Port output enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.5-3 lists the correspondence between port 9 pins and each register bit.

Table 9.5-3 Correspondence Between Registers and Pins for Port 9

	Correspondence between related register bits and pins							
Pin name	-	-	P95*1	P94*1	P93	P92	P91	P90
PDR9	-	-	bit5	bit4	bit3	bit2	bit1	bit0
DDR9	-	-						
ILSR2*2	-	-	bit3					

*1: Only for products with built-in resistor

*2: Only for 5V products, it is an effective register.

MB95150/M Series

9.5.2 Operations of Port 9

This section describes the operations of port 9.

■ Operations of Port 9

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When using the LCD shared pin as an input port, set the V0 to V3 bits (VE2/VE1) in the LCDC enable register (LCDCE1) to "0".
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- When using the LCD shared pin as an input port, set the V0 to V3 bits (VE2/VE1) in the LCDC enable register (LCDCE1) to "0".
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

- For P93 to P90, resetting the CPU initializes the DDR values to "0" and the VE2/VE1 bits in LCDCE1 to "1", and port input disabled.
- For P95/P94, resetting the CPU initializes the DDR values to "0", and sets the port input enabled.

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR value. Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation as LCDC pins

- Set the DDR register bit, which is corresponding to the LCDC dedicated pin, to "0".
- Set the V0 to V3 bits (VE2/VE1) in the LCDE enable register (LCDCE1) to "1".

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit3 of the ILSR2 register to "1" changes the port 9 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit3 of the ILSR2 register is "0".

Table 9.5-4 shows the pin states of the port 9.

Table 9.5-4 Pin State of Port 9

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff	Hi-Z Input disabled*1 (P95/P94 Input enabled*2. But not functional.)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*1: "Input disabled" means the state that the operation of the input gate close to the pin is disabled.

*2: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

MB95150/M Series

9.6 Port A

Port A is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port A Configuration

Port A is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port A data register (PDRA)
- Port A direction register (DDRA)
- Input level selection register 2 (ILSR2)

■ Port A Pins

Port A has four I/O pins.

Table 9.6-1 lists the port A pins.

Table 9.6-1 Port A Pin

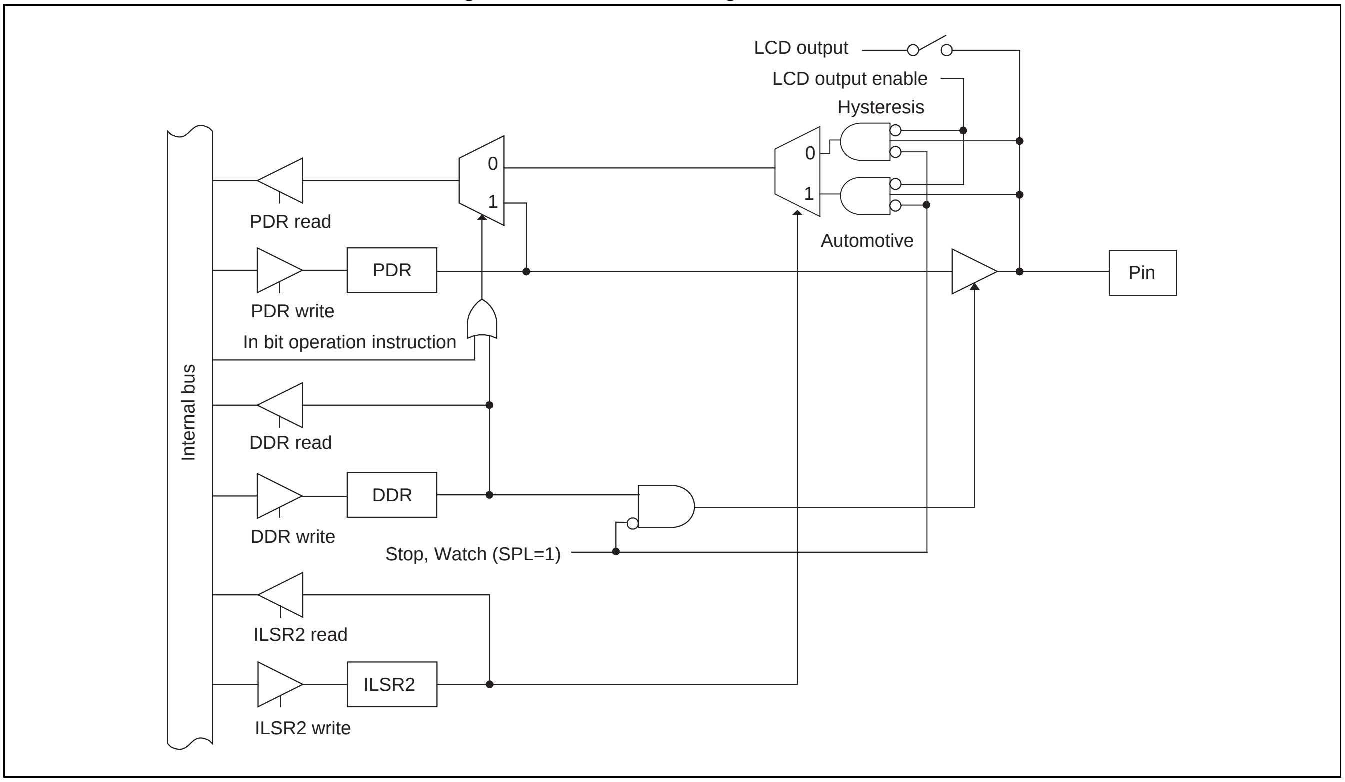
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
PA0/COM0	PA0 general-purpose I/O	COM0 LCDC COM0 output	Hysteresis/ automotive	CMOS/LCD	-	-
PA1/COM1	PA1 general-purpose I/O	COM1 LCDC COM1 output	Hysteresis/ automotive	CMOS/LCD	-	-
PA2/COM2	PA2 general-purpose I/O	COM2 LCDC COM2 output	Hysteresis/ automotive	CMOS/LCD	-	-
PA3/COM3	PA3 general-purpose I/O	COM3 LCDC COM3 output	Hysteresis/ automotive	CMOS/LCD	-	-

OD: Open drain, PU: Pull-up

*: For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

■ Block Diagram of Port A

Figure 9.6-1 Block Diagram of Port A



MB95150/M Series

9.6.1 Port A Registers

This section describes the port 9 registers.

■ Port A Register Function

Table 9.6-2 lists the port A register functions.

Table 9.6-2 Port A Register Function

Register name	Data	Read	Read read-modify-write	Write
PDRA	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDRA	0	Port input enabled		
	1	Port output enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.6-3 lists the correspondence between port A pins and each register bit.

Table 9.6-3 Correspondence Between Registers and Pins for Port A

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	PA3	PA2	PA1	PA0
PDRA	-	-	-	-	bit3	bit2	bit1	bit0
DDRA	-	-	-	-				
ILSR2*	-	-	-	-	bit4			

*: Only for 5V products, it is an effective register.

9.6.2 Operations of Port A

This section describes the operations of port A.

■ Operations of Port A

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR values to "0", and sets the port input enabled. Note that the pin sharing for the LCD output is set its port input disabled since the port input control bit (PICTL) in LCDC enable register (LCDCE1) is set to "0".

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR value. Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation as a LCDC common output

- Set the DDR register bit, which is corresponding to the LCDC common output pin, to "0".
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".

Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit4 of the ILSR2 register to "1" changes the port A input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit4 of the ILSR2 register is "0".

Table 9.6-4 shows the pin states of the port A.

Table 9.6-4 Pin State of Port A

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff	Hi-Z Input disabled*

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input disabled" means the state that the operation of the input gate close to the pin is disabled.

9.7 Port B

Port B is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port B Configuration

Port B is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port B data register (PDRB)
- Port B direction register (DDRB)
- Input level selection register 2 (ILSR2)

■ Port B Pins

Port B has eight I/O pins.

Table 9.7-1 lists the port B pins.

Table 9.7-1 Port B Pin

Pin name	Function	Shared peripheral functions		I/O type			
				Input*	Output	OD	PU
PB0/ SEG00	PB0 general-purpose I/O	SEG00	LCDC SEG00 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB1/ SEG01	PB1 general-purpose I/O	SEG01	LCDC SEG01 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB2/ SEG02	PB2 general-purpose I/O	SEG02	LCDC SEG02 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB3/ SEG03/ PPG00	PB3 general-purpose I/O	SEG03 PPG00	LCDC SEG03 output 8/16 PPG timer 00 ch.0 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB4/ SEG04/ PPG01	PB4 general-purpose I/O	SEG04 PPG01	LCDC SEG04 output 8/16 PPG timer 01 ch.0 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB5/ SEG05/ TO00	PB5 general-purpose I/O	SEG05 TO00	LCDC SEG05 output 8/16 compound timer 00 ch.0 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB6/ SEG06/ TO01	PB6 general-purpose I/O	SEG06 TO01	LCDC SEG06 output 8/16 compound timer 01 ch.0 output	Hysteresis/ automotive	CMOS/LCD	-	-
PB7/ SEG07/ EC0	PB7 general-purpose I/O	SEG07 EC0	LCDC SEG07 output 8/16 compound timer ch.0 clock input	Hysteresis/ automotive	CMOS/LCD	-	-

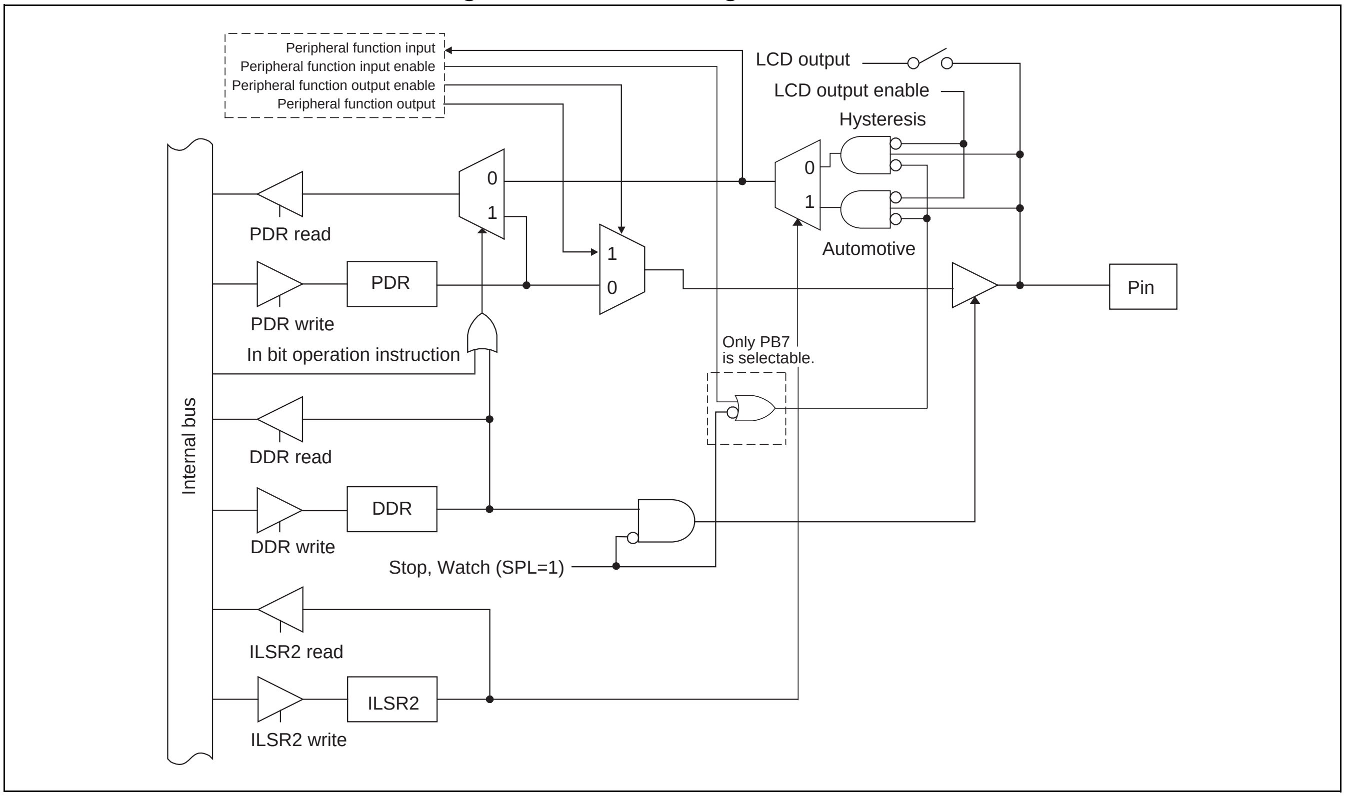
OD: Open drain, PU: Pull-up

*: For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

MB95150/M Series

■ Block Diagram of Port B

Figure 9.7-1 Block Diagram of Port B



9.7.1 Port B Registers

This section describes the port B registers.

■ Port B Register Function

Table 9.7-2 lists the port B register functions.

Table 9.7-2 Port B Register Function

Register name	Data	Read	Read read-modify-write	Write
PDRB	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDRB	0	Port input enabled		
	1	Port output enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

*: Only for 5V products, it is an effective register.

Table 9.7-3 lists the correspondence between port B pins and each register bit.

Table 9.7-3 Correspondence Between Registers and Pins for Port B

	Correspondence between related register bits and pins							
Pin name	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PDRB	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDRB								
ILSR2*	bit5							

*: Only for 5V products, it is an effective register.

9.7.2 Operations of Port B

This section describes the operations of port B.

■ Operations of Port B

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR values to "0", and sets the port input enabled. Note that the pin sharing for the LCD output is set its port input disabled since the port input control bit (PICTL) in LCDC enable register (LCDCE1) is set to "0".

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR value. Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation as a LCDC segment output

- Set the DDR register bit, which is corresponding to the LCDC segment output pin, to "0".
- When using the LCD shared pin as an input port, select the common/segment pin in LCDC enable registers (LCDCE1 to LCDCE3), and then set the port input control bit (PICTL) in LCDC enable register (LCDCE1) to "1".

Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit5 of the ILSR2 register to "1" changes the port B input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit5 of the ILSR2 register is "0".

Table 9.7-4 shows the pin states of the port B.

Table 9.7-4 Pin State of Port B

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff	Hi-Z Input disabled*

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input disabled" means the state that the operation of the input gate close to the pin is disabled.

MB95150/M Series

9.8 Port G

Port G is a general-purpose I/O port.
This section focuses on functions as a general-purpose I/O port.
See the chapters on each peripheral function for details about peripheral functions.

■ Port G Configuration

Port G is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port G data register (PDRG)
- Port G direction register (DDRG)
- Port G pull-up control register (PULG)

■ Port G Pins

Port G has one I/O pin.

Table 9.8-1 lists the port G pins.

Table 9.8-1 Port G Pins

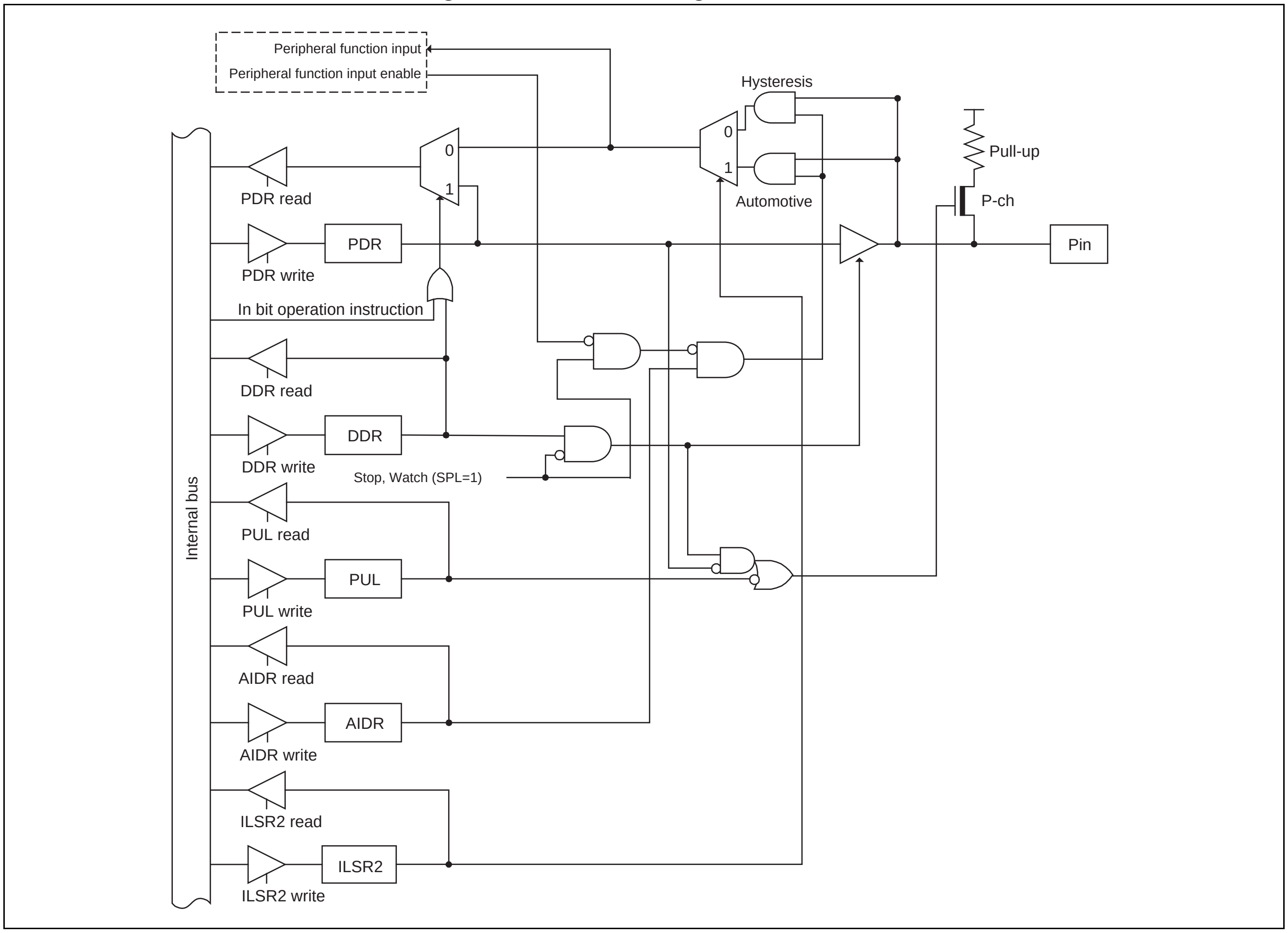
Pin name	Function	Shared peripheral functions	I/O type			
			Input	Output	OD	PU
PG0/C*	PG0 general-purpose I/O	Not shared	Hysteresis/automotive	CMOS	-	❍

OD: Open drain, PU: Pull-up

*: For the 5V product, the C pin is used.

■ Block Diagram of Port G

Figure 9.8-1 Block Diagram of Port G



9.8.1 Port G Registers

This section describes the port G registers.

■ Port G Register Function

Table 9.8-2 lists the port G register functions.

Table 9.8-2 Port G Register Function

Register name	Data	Read	Read read-modify-write	Write
PDRG	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDRG	0	Port input enabled		
	1	Port output enabled		
PULG	0	Pull-up disabled		
	1	Pull-up enabled		

Table 9.8-3 lists the correspondence between port G pins and each register bit.

Table 9.8-3 Correspondence Between Registers and Pins for Port G

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	-	-	-	PG0
PDRG	-	-	-	-	-	-	-	bit0
DDRG								
PULG								

9.8.2 Operations of Port G

This section describes the operations of port G.

■ Operations of Port G

● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.
  Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O and the output is maintained.

● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit6 of the ILSR2 register to "1" changes the port G input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit6 of the ILSR2 register is "0".

Table 9.8-4 shows the pin states of the port G.

Table 9.8-4 Pin State of Port G

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port	Hi-Z Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

CHAPTER 10
TIME-BASE TIMER

This chapter describes the functions and operations of the time-base timer.

10.1 Overview of Time-base Timer

The time-base timer is a 22-bit free-run down-counting counter which is synchronized with the main clock divided by two. The time-base timer has an interval timer function which can repeatedly generate interrupt requests at regular intervals.

■ Interval Timer Function

The interval timer function repeatedly generates interrupt requests at regular intervals by using the main clock divided by two as the count clock.

- The counter of the time-base timer counts down so that an interrupt request is generated every time the selected interval time elapses.
- The interval time can be selected from the following four types.

Table 10.1-1 shows the interval times available to the time-base timer.

Table 10.1-1 Interval Times of Time-base Timer

Internal count clock cycle	Interval time
$2/F_{CH}$(0.5 µs)	$2^{10} \times 2/F_{CH}$(512.0 µs)
	$2^{12} \times 2/F_{CH}$(2.05ms)
	$2^{14} \times 2/F_{CH}$(8.19ms)
	$2^{16} \times 2/F_{CH}$(32.77ms)

F_{CH}: Main clock
The values in parentheses represent the values used when the main clock operates at 4MHz.

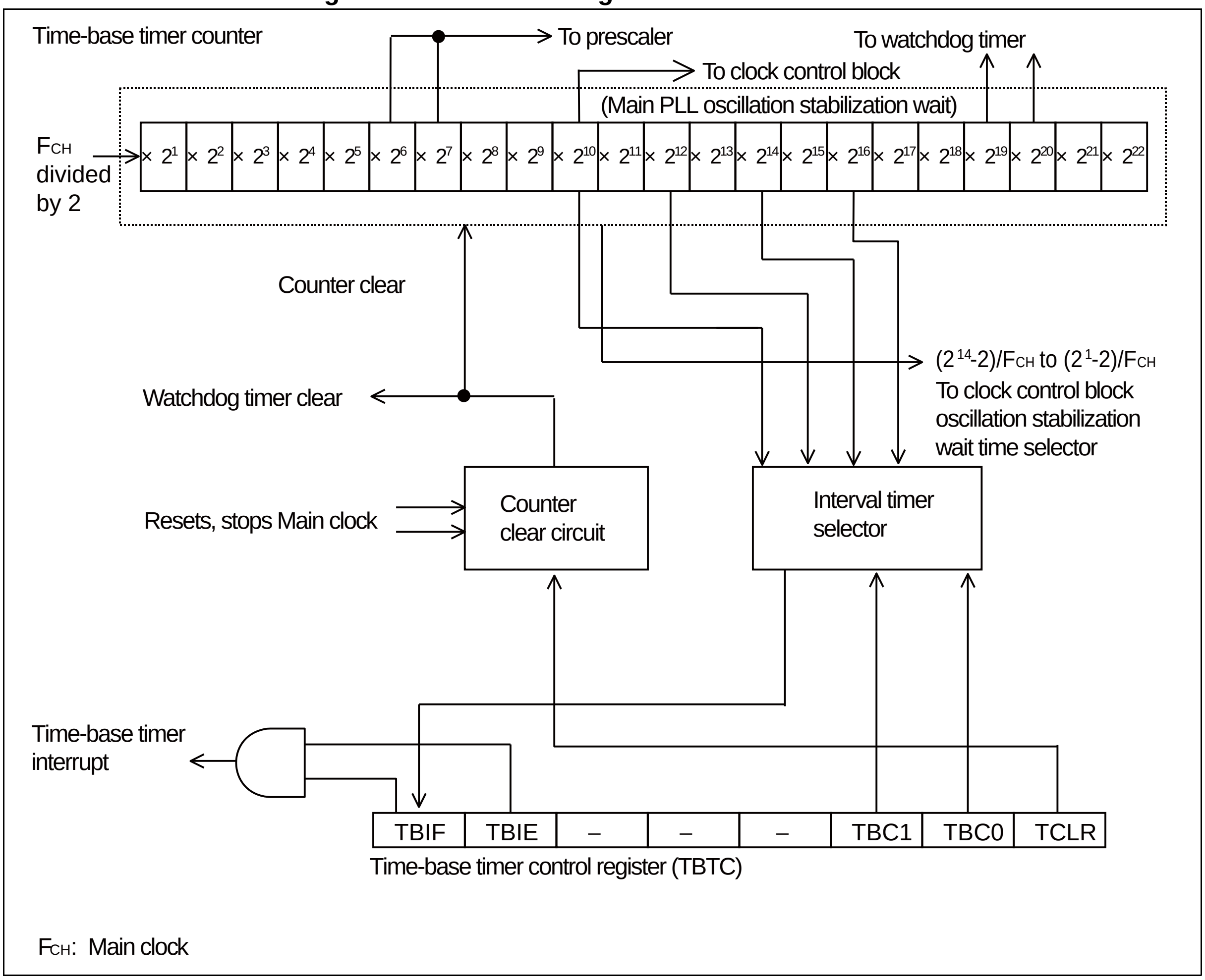
10.2 Configuration of Time-base Timer

The time-base timer consists of the following blocks:

- **Time-base timer counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Time-base Timer Control Register (TBTC)**

■ Block Diagram of Time-base Timer

Figure 10.2-1 Block Diagram of Time-base Timer



● Time-base timer counter

22-bit down-counter that uses the main clock divided by two as the count clock.

● Counter clear circuit

This circuit controls clearing of the time-base counter.

● Interval timer selector

This circuit selects the one bit from four bits in the 22 bits that make up the time-base timer counter to use the interval timer.

● Time-base timer control register (TBTC)

This register selects the interval time, clears the counter, controls interrupts and checks the status.

■ Input Clock

The time-base timer uses the main clock divided by two as its input clock (count clock).

■ Output Clock

The time-base timer supplies clocks to the main clock oscillation stabilization wait time timer, the watchdog timer and the prescaler and LCD control.

10.3 Registers of the Time-base Timer

Figure 10.3-1 shows the register of the Time-base Timer.

■ Registers of the Time-base Timer

Figure 10.3-1 Register of the Time-base Timer

Time-base timer control register (TBTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000A_H$	TBIF	TBIE	–	–	–	TBC1	TBC0	TCLR	00000000_B
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

10.3.1 Time-base Timer Control Register (TBTC)

The time-base timer control register (TBTC) selects the interval time, clears the counter, controls interrupts and checks the status.

■ Time-base Timer Control Register (TBTC)

Figure 10.3-2 Time-base Timer Control Register (TBTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000A_H$	TBIF	TBIE	—	—	—	TBC1	TBC0	TCLR	00000000_B
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

TCLR	Time-base timer initialization bit	
	Read	Write
0	"0" is always read	No change, No effect on operation
1	—	Clears the counter of time-base timer

TBC1	TBC0	Interval time select bit (Main clock F_{CH} = 4MHz)
0	0	$2^{10}\times 2/F_{CH}$ (512.0 µs)
0	1	$2^{12}\times 2/F_{CH}$ (2.05ms)
1	0	$2^{14}\times 2/F_{CH}$ (8.19ms)
1	1	$2^{16}\times 2/F_{CH}$ (32.77ms)

TBIE	Time-base timer interrupt request enable bit
0	Disables output of interrupt request
1	Enables output of interrupt request

TBIF	Time-base timer interrupt request flag bit	
	Read	Write
0	Interval time has not elapsed	Clears bit
1	Interval time has elapsed	No change, No effect on operation

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
— : Undefined
(shaded) : Initial value

Table 10.3-1 Functional Description of Each Bit of Time-base Timer Control Register (TBTC)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>TBIF:
Time-base timer interrupt request flag bit</td><td>Set to "1" when interval time selected by the time-base timer elapses.
Interrupt request is outputted when this bit and the time-base timer interrupt request enable bit (TBIE) are set to "1".
Writing "0": clears the bit.
Writing "1": has no effect on operation.
"1" is always read in read-modify-write (RMW) instruction.</td></tr>
<tr><td>bit6</td><td>TBIE:
Time-base timer interrupt request enable bit</td><td>This bit enables/disables output of interrupt requests to the interrupt controller.
Writing "0": disables output of time-base timer interrupt requests.
Writing "1": enables output of time-base timer interrupt requests.
Interrupt request is outputted when this bit and the time-base timer interrupt request flag bit (TBIF) are set to "1".</td></tr>
<tr><td>bit5
to
bit3</td><td>Undefined bits</td><td>These bits are undefined.
• The read value is always "0".
• Writing has no effect on the operation.</td></tr>
<tr><td>bit2,
bit1</td><td>TBC1, TBC0:
Interval
time select bits</td><td>These bits select the interval time.
<table>
<tr><th>TBC1</th><th>TBC0</th><th>Interval time select bits
(Main clock F_{CH} = 4MHz)</th></tr>
<tr><td>0</td><td>0</td><td>$2^{10} \times 2/F_{CH}$(512.0 µs)</td></tr>
<tr><td>0</td><td>1</td><td>$2^{12} \times 2/F_{CH}$(2.05ms)</td></tr>
<tr><td>1</td><td>0</td><td>$2^{14} \times 2/F_{CH}$(8.19ms)</td></tr>
<tr><td>1</td><td>1</td><td>$2^{16} \times 2/F_{CH}$(32.77ms)</td></tr>
</table></td></tr>
<tr><td>bit0</td><td>TCLR:
Time-base timer initialization bit</td><td>This bit clears the time-base timer counter.
Writing "0": ignored and has no effect on the operation.
Writing "1": initializes all counter bits to "1".
The read value is always "0".
Note: When the output of the time-base timer is selected as the count clock for the watchdog timer, using this bit to clear the time-base timer also clears the watchdog timer.</td></tr>
</table>

10.4 Interrupts of Time-base Timer

An interrupt request is triggered when the interval time selected by the time-base timer elapses (interval timer function).

■ Interrupt when Interval Function is in Operation

When the time-base timer counter counts down using the internal count clock and the selected time-base timer counter underflows, the time-base timer interrupt request flag bit (TBTC:TBIF) is set to "1". If the time-base timer interrupt request enable bit is enabled (TBTC:TBIE=1), an interrupt request (IRQ19) will be generated to interrupt controller.

- Regardless of the value of TBIE bit, TBIF bit is set to "1", when the selected bit underflows.
- When TBIF bit is set to "1" and TBIE bit is changed from the disable state to the enable state (0 → 1), an interrupt request is generated immediately.
- TBIF bit is not set when the counter is cleared (TBTC:TCLR = 1) and the time-base timer counter underflows at the same time.
- Write "1" to TBIF bit to clear an interrupt request in an interrupt processing routine.

Note:

When enabling the output of interrupt requests after canceling a reset (TBTC:TBIE = 1), always clear TBIF bit at the same time (TBTC:TBIF = 0).

Table 10.4-1 Interrupts of Time-base Timer

Item	Description
Interrupt condition	Interval time set by "TBTC:TBC1" and "TBC0" has elapsed
Interrupt flag	TBTC:TBIF
Interrupt enable	TBTC:TBIE

■ Register and Vector Table for Interrupts of Time-base Timer

Table 10.4-2 Register and Vector Table for Interrupts of Time-base Timer

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Time-base timer	IRQ19	ILR4	L19	$FFD4_H$	$FFD5_H$

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

Note:

If the interval time set for the time-base timer is shorter than the main clock oscillation stabilization wait time, an interrupt request of the time-base timer is generated during the main clock oscillation wait time derived from the transition to the clock mode or standby mode. To prevent this, set the time-base timer interrupt request enable bit of the time-base timer control register (TBTC:TBIE) to "0" to disable interrupts of the time-base timer when entering a mode in which the main clock stops oscillating (stop mode, sub clock mode or sub PLL clock mode).

10.5 Explanation of Time-base Timer Operations and Setup Procedure Example

This section describes the operations of the interval timer function of the time-base timer.

■ Operations of Time-base Timer

The counter of the time-base timer is initialized to "$3FFFFF_H$" after a reset and starts counting while being synchronized with the main clock divided by two.

The time-base timer continues to count down as long as the main clock is oscillating. Once the main clock halts, the counter stops counting and is initialized to "$3FFFFF_H$".

The settings shown in Figure 10.5-1 are required to use the interval timer function.

Figure 10.5-1 Settings of Interval Timer Function

TBTC

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Address: $000A_H$	TBIF	TBIE	-	-	-	TBC1	TBC0	TCLR
	0	1				◎	◎	0

◎: Bit used
1: Set to "1"
0: Set to "0"

When the time-base timer initialization bit in the time-base timer control register (TBTC:TCLR) is set to "1", the counter of the time-base timer is initialized to "$3FFFFF_H$" and continues to count down. When the selected interval time has elapsed, the time-base timer interrupt request flag bit of the time-base timer control register (TBTC:TBIF) becomes "1". In other words, an interrupt request is generated at each interval time selected, based on the time when the counter was last cleared.

■ Clearing Time-base Timer

If the time-base timer is cleared when the output of the time-base timer is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

When clearing the counter by using the time-base timer initialization bit (TBTC:TCLR), perform setup so that this does not have unexpected effects on other peripheral functions.

When the output of the time-base timer is selected as the count clock for the watchdog timer, clearing the time-base timer also clears the watchdog timer.

The time-base timer is cleared not only by the time-base timer initialization bit (TBTC:TCLR), but also when the main clock is stopped and a count is required for the oscillation stabilization wait time. More specifically, the time-base timer is cleared in the following situations:

- When moving from the main clock mode or main PLL clock mode to the stop mode
- When moving from the main clock mode or main PLL clock mode to the sub clock mode or sub PLL clock mode
- At power on
- At low-voltage detection reset

The counter of the time-base timer is also cleared and stops the operation if a reset occurs while the main clock is still running after the main clock oscillation stabilization wait time has elapsed. The counter, however, continues to operate during a reset if a count is required for the oscillation stabilization wait time.

■ Operating Examples of Time-base Timer

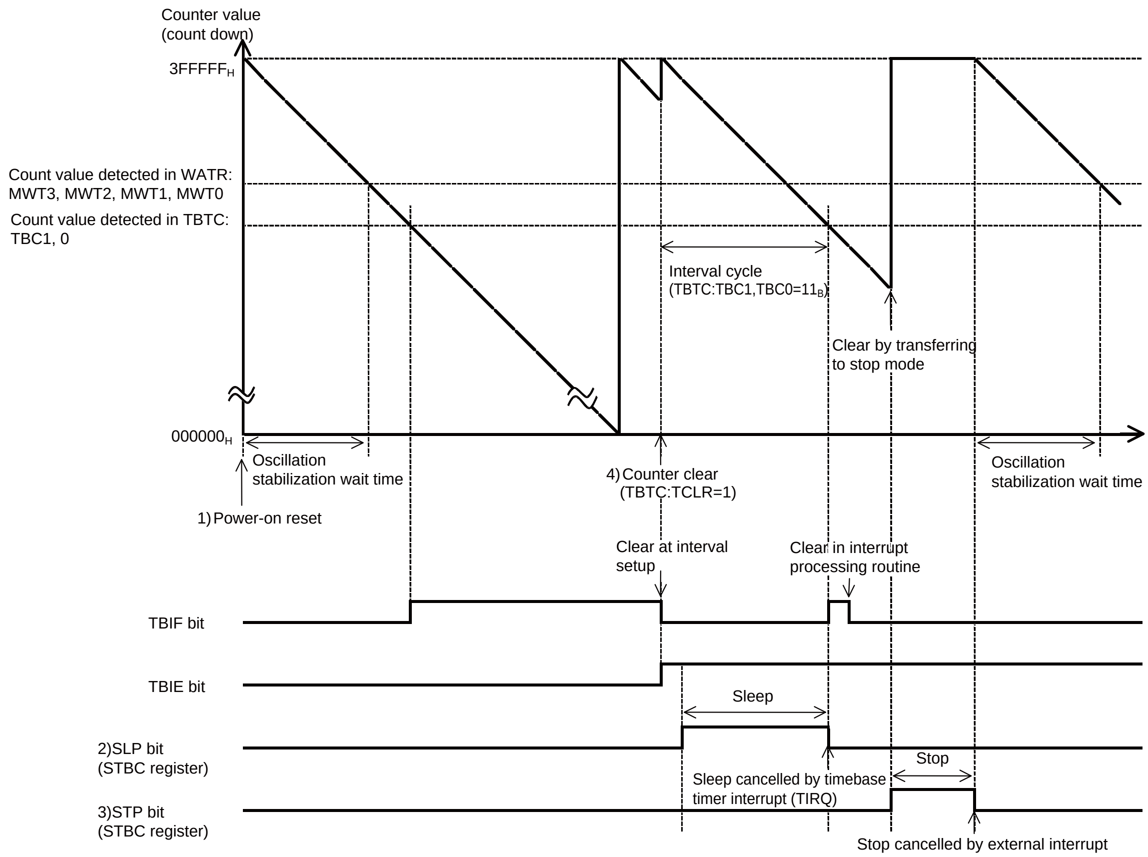
Figure 10.5-2 shows operating examples of operation under the following conditions:

1) When a power-on reset is generated
2) When entering the sleep mode during the operation of the interval timer function in the main clock mode or main PLL clock mode
3) When entering the stop mode during the main clock mode or main PLL clock mode
4) When a request is issued to clear the counter

The same operation is performed when changing to the time-base timer mode as for when changing to the sleep mode.

In the sub clock mode, sub PLL clock mode, main clock mode and main PLL clock mode, the timer operation is stopped during the stop mode, as the time-base timer is cleared and the main clock halts. Upon recovering from the stop mode, the time-base timer is used to count the oscillation stabilization wait time.

Figure 10.5-2 Operations of Time-base Timer



- When setting "11$_B$" to interval time select bits of time-base timer control register (TBTC:TBC1, TBC0) (2^{16} x 2/F_{CH})

- TBTC:TBC1,TBC0 : Interval time select bits of time-base timer control register
- TBTC:TCLR : Time-base timer initialization bit of time-base timer control register
- TBTC:TBIF : Time-base timer interrupt request flag bit of time-base timer control register
- TBTC:TBIE : Time-base timer interrupt request enable bit of time-base timer control register
- STBC:SLP : Sleep bit of standby control register
- STBC:STP : Stop bit of standby control register
- WATR:MWT3 to MWT0 : Main clock oscillation stabilization wait time select bit of oscillation stabilization wait time setup register

■ Setup Procedure Example

● Initial setting

The time-base timer is set up in the following procedure:

1) Disable interrupts. (TBTC:TBIE = 0)
2) Set the interval time. (TBTC:TBC1, TBC0)
3) Enable interrupts. (TBTC:TBIE = 1)
4) Clear the counter. (TBTC:TCLR = 1)

● Interrupt processing

1) Clear the interrupt request flag.(TBTC:TBIF = 0)
2) Clear the counter. (TBTC:TCLR = 1)

10.6 Notes on Using Time-base Timer

Care must be taken for the following points when using the Time-base Timer.

■ Notes on Using Time-base Timer

● When setting the timer by program

The timer cannot be recovered from interrupt processing, when the time-base timer interrupt request flag bit (TBTC:TBIF) is set to "1" and the interrupt request enable bit is enabled (TBTC:TBIE = 1). Always clear TBIF bit in the interrupt processing routine.

● Clearing time-base timer

The time-base timer is cleared not only by the time-base timer initialization bit (TBTC:TCLR=1) but also when the oscillation stabilization wait time is required for the main clock. When the time-base timer is selected for the count clock of the watchdog timer (WDTC:CS1, CS0 = 00_B or CS1, CS0 = 01_B), clearing the time-base timer also clears the watchdog timer.

● Peripheral functions receiving clock from time-base timer

In the mode where the source oscillation of the main clock is stopped, the counter is cleared and the time-base timer stops operation. In addition, if the time-base timer is cleared when the output of the time-base timer is used in other peripheral functions, this will affect the operation such as cycle change. The clock for the watchdog timer is also outputted from the initial state. However, as the watchdog timer counter is cleared at the same time, the watchdog timer operates in the normal cycles.

CHAPTER 11
WATCHDOG TIMER

This chapter describes the functions and operations of the watchdog timer.

Code: CM26-00106-3E

11.1 Overview of Watchdog Timer

The watchdog timer functions as a counter used to prevent programs from running out of control.

■ Watchdog Timer Function

The watchdog timer functions as a counter used to prevent programs from running out of control. Once the watchdog timer is activated, its counter needs to be cleared at specified intervals regularly. A watchdog reset is generated if the timer is not cleared within a certain amount of time due to a problem such as the program entering an infinite loop.

The output of either the time-base timer or watch prescaler can be selected as the count clock for the watchdog timer.

The interval times of the watchdog timer are shown in Table 11.1-1. If the counter of the watchdog timer is not cleared, a watchdog reset is generated between the minimum time and the maximum time. Clear the counter of the watchdog timer within the minimum time.

Table 11.1-1 Interval Times of Watchdog Timer

Count clock type	Count clock switch bits (WDTC:CS1, CS0)*	Interval time	
		Minimum time	Maximum time
Time-base timer output (main clock = 4MHz)	00_B	524 ms	1.05 s
	01_B	262 ms	524 ms
Watch prescaler output (sub clock = 32.768kHz)	10_B	500 ms	1.00 s
	11_B	250 ms	500 ms

*: WDTC:CS1, 0: Count clock switch bit of watchdog timer control register

For information about the minimum and maximum times of the watchdog timer interval, refer to "11.4 Explanation of Watchdog Timer Operations and Setup Procedure Example".

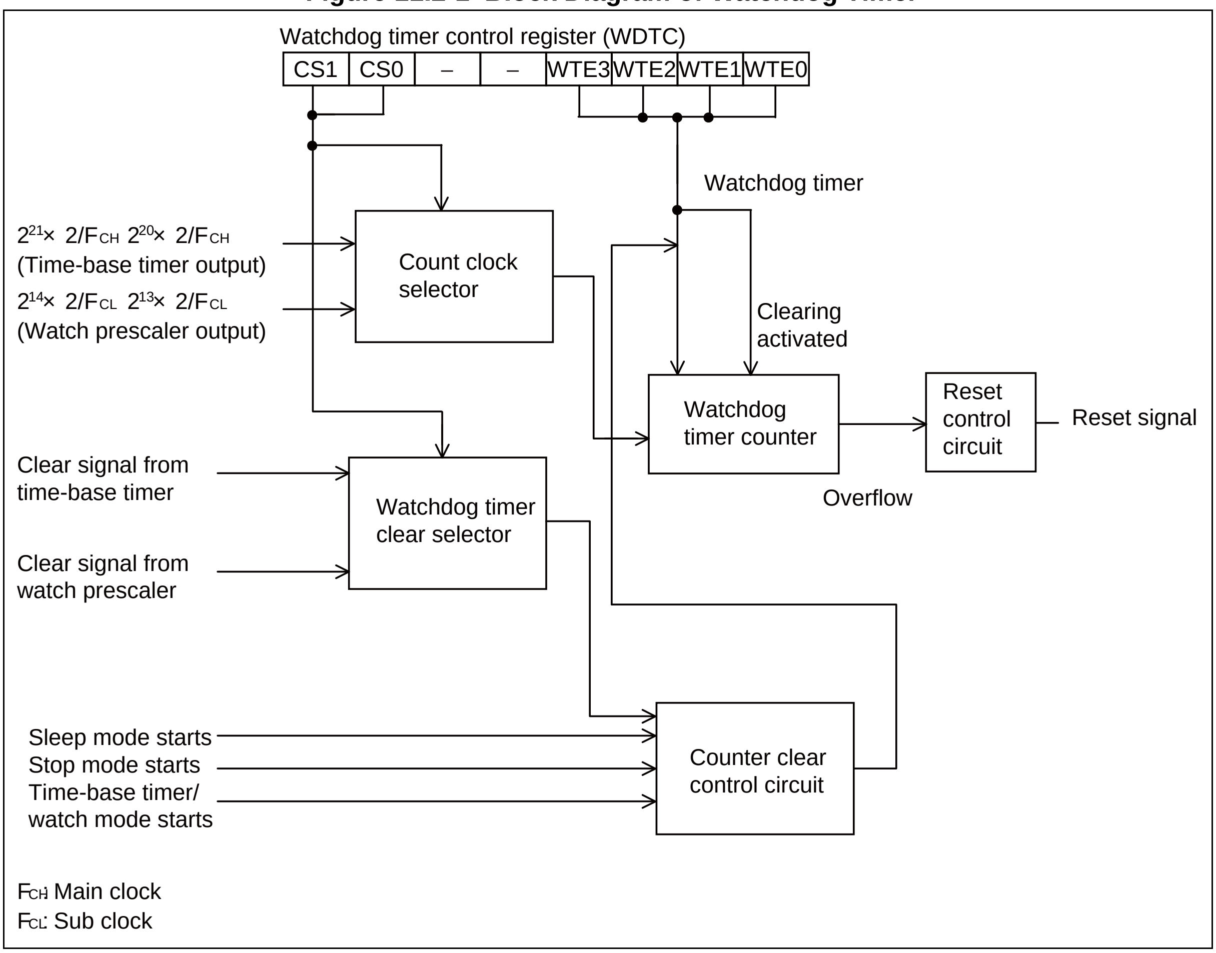
11.2 Configuration of Watchdog Timer

The watchdog timer consists of the following blocks:

- **Count clock selector**
- **Watchdog timer counter**
- **Reset control circuit**
- **Watchdog timer clear selector**
- **Counter clear control circuit**
- **Watchdog Timer Control Register (WDTC)**

■ Block Diagram of Watchdog Timer

Figure 11.2-1 Block Diagram of Watchdog Timer



- Count clock selector

  This selector selects the count clock of the watchdog timer counter.

- Watchdog timer counter

  This is a 1-bit counter that uses the output of either the time-base timer or watch prescaler as the count clock.

- Reset control circuit

  This circuit generates a reset signal when the watchdog timer counter overflows.

- Watchdog timer clear selector

  This selector selects the watchdog timer clear signal.

- Counter clear control circuit

  This circuit controls the clearing and stopping of the watchdog timer counter.

- Watchdog timer control register (WDTC)

  This register performs setup for activating/clearing the watchdog timer counter as well as for selecting the count clock.

■ Input Clock

The watchdog timer uses the output clock from either the time-base timer or watch prescaler as the input clock (count clock).

11.3 Register of The Watchdog Timer

Figure 11.3-1 shows the register of the watchdog timer.

■ Register of The Watchdog Timer

Figure 11.3-1 Register of The Watchdog Timer

Watchdog timer control register (WDTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000C_H$	CS1	CS0	–	–	WTE3	WTE2	WTE1	WTE0	00000000_B
	R/W	R/W	R0/WX	R0/WX	R0,W	R0,W	R0,W	R0,W	

R/W : Readable/writable (Read value is the same as write value)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

11.3.1 Watchdog Timer Control Register (WDTC)

The watchdog timer control register (WDTC) activates or clears the watchdog timer.

■ Watchdog Timer Control Register (WDTC)

Figure 11.3-2 Watchdog Timer Control Register (WDTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000C_H$	CS1	CS0	–	–	WTE3	WTE2	WTE1	WTE0	00000000_B
	R/W	R/W	R0/WX	R0/WX	R0,W	R0,W	R0,W	R0,W	

WTE3	WTE2	WTE1	WTE0	Watchdog control bits
0	1	0	1	• Activate watchdog timer (in first write after reset) • Clear watchdog timer (in second or succeeding write after reset)
Other than above				No effect on operation

CS1	CS0	Count clock switch bits
0	0	Output cycle of time-base timer ($2^{21}/F_{CH}$)
0	1	Output cycle of time-base timer ($2^{20}/F_{CH}$)
1	0	Output cycle of watch prescaler ($2^{14}/F_{CL}$)
1	1	Output cycle of watch prescaler ($2^{13}/F_{CL}$)

R/W : Readable/writable (Read value is the same as write value)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined
(shaded) : Initial value
F_{CH} : Main clock
F_{CL} : Sub clock

Table 11.3-1 Functional Description of Each Bit of Watchdog Timer Control Register (WDTC)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7,
bit6</td><td>CS1, CS0:
Count clock switch bits</td><td>These bits select the count clock of the watchdog timer.
<table>
<tr><th>CS1</th><th>CS0</th><th>Count clock switch bits</th></tr>
<tr><td>0</td><td>0</td><td>Output cycle of time-base timer ($2^{21}/F_{CH}$)</td></tr>
<tr><td>0</td><td>1</td><td>Output cycle of time-base timer ($2^{20}/F_{CH}$)</td></tr>
<tr><td>1</td><td>0</td><td>Output cycle of watch prescaler ($2^{14}/F_{CL}$)</td></tr>
<tr><td>1</td><td>1</td><td>Output cycle of watch prescaler ($2^{13}/F_{CL}$)</td></tr>
</table>
• Write to these bits at the same time as activating the watchdog timer by the watchdog control bits.
• No change can be made once the watchdog timer is activated.
Note: Always select the output of the watch prescaler in the sub clock mode or sub PLL clock mode, as the time-base timer is stopped in these modes. Do not select the output of the watch prescaler in single clock product.</td></tr>
<tr><td>bit5,
bit4</td><td>Undefined bits</td><td>These bits are undefined.
• The read value is "00_B".
• Writing has no effect on the operation.</td></tr>
<tr><td>bit3
to
bit0</td><td>WTE3, WTE2, WTE1, WTE0:
Watchdog control bits</td><td>These bits are used to control the watchdog timer.
Writing "0101_B": activates the watchdog timer (in first write after reset) or clears it (in second or succeeding write after reset).
Writing other than "0101_B": has no effect on operation.
• The read value is "0000_B".</td></tr>
</table>

Read-modify-write (RMW) instructions cannot be used.

11.4 Explanation of Watchdog Timer Operations and Setup Procedure Example

The watchdog timer generates a watchdog reset when the watchdog timer counter overflows.

■ Operations of Watchdog Timer

● How to activate the watchdog timer

- The timer of the watchdog timer is activated when "0101_B" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0) for the first time after a reset. The count clock switch bits of the watchdog timer control register (WDTC:CS1,CS0) should also be set at the same time.
- Once the watchdog timer is activated, a reset is the only way to stop its operation.

● Clearing the watchdog timer

- When the counter of the watchdog timer is not cleared within the interval time, it overflows, allowing the watchdog timer to generate a watchdog reset.
- The counter of the watchdog timer is cleared when "0101_B" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0) for the second or any succeeding time.
- The watchdog timer is cleared at the same time as the timer selected as the count clock (time-base timer or watch prescaler) is cleared.

● Operations in standby mode

Regardless of the clock mode selected, the watchdog timer clears its counter and stops the operation when entering a standby mode (sleep/stop/time-base timer/watch).

Once released from the standby mode, the timer restarts the operation.

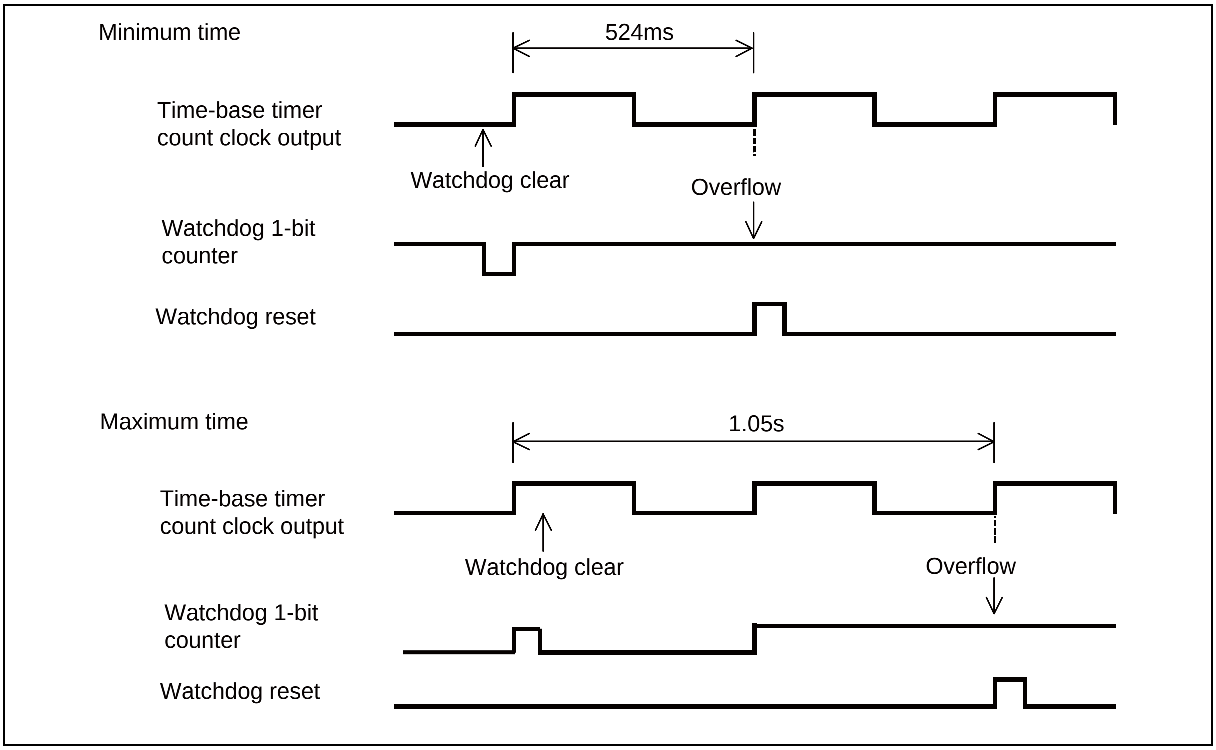
Note:

The watchdog timer is also cleared when the timer selected as the count clock (time-base timer or watch prescaler) is cleared.
For this reason, the watchdog timer cannot function as such, if the software is set to clear the selected timer repeatedly during the interval time of the watchdog timer.

● Interval time

The interval time varies depending on the timing for clearing the watchdog timer. Figure 11.4-1 shows the correlation between the clearing timing of the watchdog timer and the interval time.

Figure 11.4-1 Clearing Timing and Interval Time of Watchdog Timer (Main clock = 4MHz, WDTC:CS1, CS0=00_B)



● Operation in the sub clock mode

When a watchdog reset is generated in the sub clock mode, the timer starts operating in the main clock mode after the oscillation stabilization wait time has elapsed. The reset signal is outputted during this oscillation stabilization wait time.

■ Setup Procedure Example

The watchdog timer is set up in the following procedure:

1) Select the count clock. (WDTC:CS1, CS0)

2) Activate the watchdog timer. (WDTC:WTE3 to WTE0 = 0101_B)

3) Clear the watchdog timer. (WDTC:WTE3 to WTE0 = 0101_B)

11.5 Notes on Using Watchdog Timer

Care must be taken for the following points when using the watchdog timer.

■ Notes on Using Watchdog Timer

- Stopping the watchdog timer

  Once activated, the watchdog timer cannot be stopped until a reset is generated.

- Selecting the count clock

  The count clock switch bits (WDTC:CS1, 0) can be rewritten only when the watchdog control bits (WDTC:WTE3 to WTE0) are set to "0101_B" upon the activation of the watchdog timer. The count clock switch bits cannot be written by a bit operation instruction. Moreover, the bit settings should not be changed once the timer is activated.

  In the sub clock mode, the time-base timer does not operate because the main clock stops oscillating.

  In order to operate the watchdog timer in the sub clock mode, it is necessary to select the watch prescaler as the count clock beforehand and set "WDTC:CS1, 0" to "10_B" or "11_B".

- Clearing the watchdog timer

  Clearing the counter used for the count clock of the watchdog timer (time-base timer or watch prescaler) also clears the counter of the watchdog timer.

  The counter of the watchdog timer is cleared when entering the sleep mode, stop mode or watch mode.

- Programming precaution

  When creating a program in which the watchdog timer is cleared repeatedly in the main loop, set the processing time of the main loop including the interrupt processing time to the minimum watchdog timer interval time or shorter.

CHAPTER 12
WATCH PRESCALER

This chapter describes the functions and operations of the watch prescaler.

Code: CM26-00107-1E

12.1 Overview of Watch Prescaler

The watch prescaler is a 15-bit down-counting, free-run counter, which is synchronized with the sub clock divided by two. It has an interval timer function that continuously generates interrupt requests at regular intervals.

■ Interval Timer Function

The interval timer function continuously generates interrupt requests at regular intervals, using the sub clock divided by two as its count clock.

- The counter of the watch prescaler counts down and an interrupt request is generated every time the selected interval time has elapsed.
- The interval time can be selected from the following four types:

Table 12.1-1 shows the interval times of the watch prescaler.

Table 12.1-1 Interval Times of Watch Prescaler

Internal count clock cycle	Interval time
$2/F_{CL}$ (61.0 µs)	$2^{11} \times 2/F_{CL}$(125ms)
	$2^{12} \times 2/F_{CL}$(250ms)
	$2^{13} \times 2/F_{CL}$(500ms)
	$2^{14} \times 2/F_{CL}$(1.00s)

F_{CL}: sub clock
The values in parentheses represent the values achieved when the sub clock operates at 32.768kHz.

Note:

The watch prescaler cannot be used in single clock product.

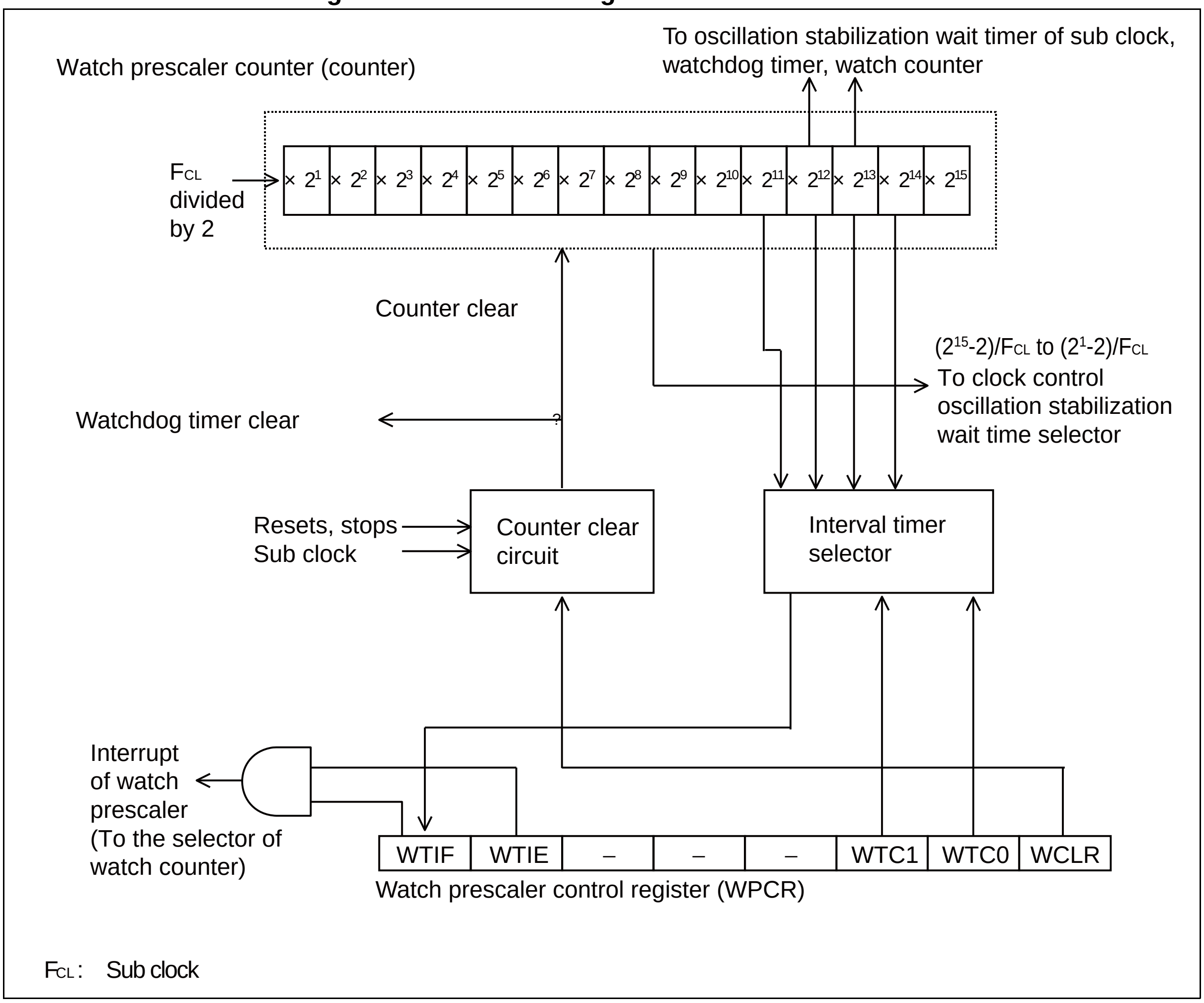
12.2 Configuration of Watch Prescaler

The watch prescaler consists of the following blocks:

- **Watch prescaler counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Watch Prescaler Control Register (WPCR)**

■ Block Diagram of Watch Prescaler

Figure 12.2-1 Block Diagram of Watch Prescaler



● Watch prescaler counter (counter)

This is a 15-bit down-counter that uses the sub clock divided by two as its count clock.

● Counter clear circuit

This circuit controls the clearing of the watch prescaler.

● Interval timer selector

This circuit selects one out of the four bits used for the interval timer among 15 bits available in the watch prescaler counter.

● Watch prescaler control register (WPCR)

This register selects the interval time, clears the counter, controls interrupts and checks the status.

■ Input Clock

The watch prescaler uses the sub clock divided by two as its input clock (count clock).

■ Output Clock

The watch prescaler supplies its clock to the timer for the oscillation stabilization wait time of the sub clock, the watchdog timer and the watch counter.

12.3 Registers of the Watch Prescaler

Figure 12.3-1 shows the register of the watch prescaler.

■ Register of the Watch Prescaler

Figure 12.3-1 Register of the Watch Prescaler

Watch Prescaler Control Register (WPCR)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000B_H$	WTIF	WTIE	–	–	–	WTC1	WTC0	WCLR	00000000_B
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

12.3.1 Watch Prescaler Control Register (WPCR)

The watch prescaler control register (WPCR) is a register used to select the interval time, clear the counter, control interrupts and check the status.

■ Watch Prescaler Control Register (WPCR)

Figure 12.3-2 Watch Prescaler Control Register (WPCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000B_H$	WTIF	WTIE	–	–	–	WTC1	WTC0	WCLR	00000000_B
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

WCLR	Watch timer initialization bit	
	Read	Write
0	"0" is always read	No change, No effect on operation
1	–	Clears watch prescaler counter

WTC1	WTC0	Watch interrupt interval timer time select bit (Sub clock F_{CL} = 32.768kHz)
0	0	$2^{11} \times 2/F_{CL}$ (125ms)
0	1	$2^{12} \times 2/F_{CL}$ (250ms)
1	0	$2^{13} \times 2/F_{CL}$ (500ms)
1	1	$2^{14} \times 2/F_{CL}$ (1.00s)

WTIE	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

WTIF	Watch interrupt request flag bit	
	Read	Write
0	Interval time has not elapsed	Clears the bit
1	Interval time has elapsed	No change, No effect on operation

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0,W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined
(shaded) : Initial value

Table 12.3-1 Functional Description of Each Bit of Watch Prescaler Control Register (WPCR)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>WTIF:
Watch interrupt request flag bit</td><td>This bit becomes "1" when the selected interval time of the watch prescaler has elapsed.
• Interrupt requests are generated when this bit and the interrupt request enable bit (WTIE) are set to "1".
Writing "0": sets this bit to "0".
Writing "1": ignored and has no effect on the operation.
• "1" is always read in read-modify-write (RMW) instruction.</td></tr>
<tr><td>bit6</td><td>WTIE:
Interrupt request enable bit</td><td>This bit enables/disables output of interrupt requests to the interrupt controller.
Writing "0": disables the interrupt request output of the watch prescaler.
Writing "1": enables the interrupt request output of the watch prescaler.
Interrupt requests are outputted when this bit and the watch interrupt request flag bit (WTIF) are set to "1".</td></tr>
<tr><td>bit5
to
bit3</td><td>Undefined bits</td><td>These bits are undefined.
• The read value is always "0".
• Writing has no effect on the operation.</td></tr>
<tr><td>bit2,
bit1</td><td>WTC1, WTC0:
Watch interrupt interval time select bits</td><td>These bits select the interval time.
<table>
<tr><th>WTC1</th><th>WTC0</th><th>Interval time select bits
(sub clock F_{CL} = 32.768kHz)</th></tr>
<tr><td>0</td><td>0</td><td>$2^{11} \times 2/F_{CL}$(125ms)</td></tr>
<tr><td>0</td><td>1</td><td>$2^{12} \times 2/F_{CL}$(250ms)</td></tr>
<tr><td>1</td><td>0</td><td>$2^{13} \times 2/F_{CL}$(500ms)</td></tr>
<tr><td>1</td><td>1</td><td>$2^{14} \times 2/F_{CL}$(1.00s)</td></tr>
</table></td></tr>
<tr><td>bit0</td><td>WCLR:
Watch timer initialization bit</td><td>This bit clears the counter for the watch prescaler.
Writing "0": ignored and has no effect on the operation.
Writing "1": initializes all counter bits to "1".
The read value is always "0".
Note: When the output of the watch prescaler is selected as the count clock of the watchdog timer, clearing the watch prescaler with this bit also clears the watchdog timer.</td></tr>
</table>

12.4 Interrupts of Watch Prescaler

An interrupt request is generated when the selected interval time of the watch prescaler has elapsed (interval timer function).

■ Interrupts in Operation of Interval Timer Function (Watch Interrupts)

In any mode other than the main clock stop mode, the watch interrupt request flag bit is set to "1" (WPCR:WTIF = 1), when the watch prescaler counter counts up by using the source oscillation of the sub clock and the time of the interval timer has elapsed. If the interrupt request enable bit is also enabled (WPCR:WTIE = 1) and watch counter start interrupt request enable bit of the watch counter is disabled (WCSR:ISEL=0), an interrupt request (IRQ20) occurs from watch prescaler to an interrupt controller.

- Regardless of the value in the WTIE bit, the WTIF bit is set to "1" when the time set by the watch interrupt interval time select bits has been reached.
- When the WTIF bit is set to "1", changing the WTIE bit from the disable state to the enable state (WPCR:WTIE = 0 $\rightarrow$ 1) immediately generates an interrupt request.
- The WTIF bit cannot be set when the counter is cleared (WPCR:WCLR = 1) at the same time as the selected bit overflows.
- Write "0" to the WTIF bit in the interrupt processing routine to clear an interrupt request to "0".

Note:

When enabling the output of interrupt requests (WPCR:WTIE = 1) after canceling a reset, always clear the WTIF bit at the same time (WPCR:WTIF=0).

■ Interrupts of Watch Prescaler

Table 12.4-1 Interrupts of Watch Prescaler

Item	Description
Interrupt condition	Interval time set by "WPCR:WTC1" and "WTC0" has elapsed.
Interrupt flag	WPCR:WTIF
Interrupt enable	WPCR:WTIE

■ Register and Vector Table Related to Interrupts of Watch Prescaler

Table 12.4-2 Register and Vector Table Related to Interrupts of Watch Prescaler

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Watch prescaler*	IRQ20	ILR5	L20	$FFD2_H$	$FFD3_H$

*: The watch prescaler shares the same interrupt request number and vector table as the watch counter.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

Note:

If the interval time set for the watch prescaler is shorter than the oscillation stabilization wait time of the sub clock, an interrupt request of the watch prescaler is generated during the oscillation stabilization wait time of the sub clock required for recovery by an external interrupt upon the transition from the sub clock mode or the sub PLL clock mode to the stop mode. To prevent this, set the interrupt request enable bit (WPCR:WTIE) in the watch prescaler control register to "0" to disable interrupts of the watch prescaler when entering the stop mode during the sub clock mode or the sub PLL clock mode.

12.5 Explanation of Watch Prescaler Operations and Setup Procedure Example

The watch prescaler operates as an interval timer.

■ Operations of Interval Timer Function (Watch Prescaler)

The counter of the watch prescaler continues to count down using the sub clock divided by two as its count clock as long as the sub clock oscillates.

When cleared (WPCR:WCLR=1), the counter starts to count down from "$7FFF_H$". Once "0000_H" is reached, the counter returns to "$7FFF_H$" to continue the count. When the time set by the interrupt interval time select bits is reached during down-counting, the watch interrupt request flag bit (WPCR:WTIF) is set to "1" in any mode other than the main clock stop mode. In other words, a watch interrupt request is generated at each selected interval time, based on the time when the counter was last cleared.

■ Clearing Watch Prescaler

If the watch prescaler is cleared when the output of the watch prescaler is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

When clearing the counter by using the watch prescaler initialization bit (WPCR:WCLR), perform setup so that this does not have unexpected effects on other peripheral functions.

When the output of the watch prescaler is selected as the count clock, clearing the watch prescaler also clears the watchdog timer.

The watch prescaler is cleared not only by the watch prescaler initialization bit (WPCR:WCLR) but also when the sub clock is stopped and a count is required for the oscillation stabilization wait time.

- When moving from the sub clock mode or sub PLL clock mode to the stop mode
- When the sub clock oscillation stop bit in the system clock control register (SYCC:SUBS) is set to "1" in the main clock mode or main PLL clock mode

In addition, the counter of the watch prescaler is cleared and stops operation when a reset is generated.

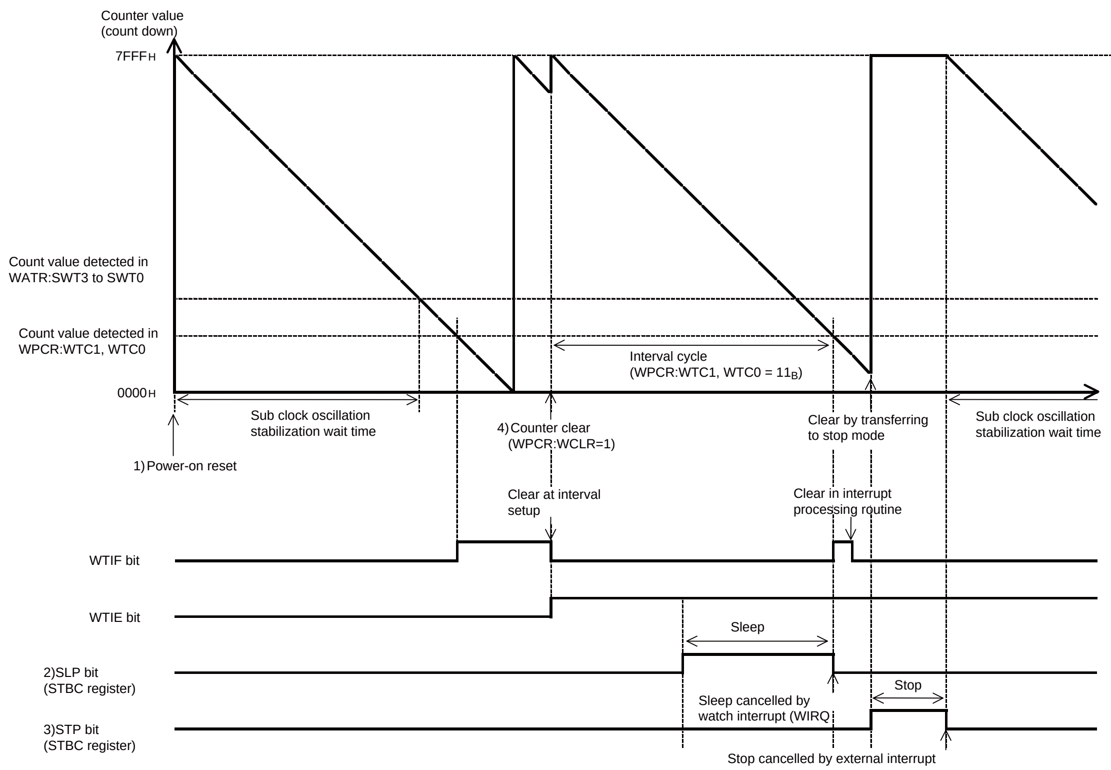
■ Operating Examples of Watch Prescaler

Figure 12.5-1 shows operating examples under the following conditions:

1) When a power-on reset is generated
2) When entering the sleep mode during the operation of the interval timer function in the sub clock mode or sub PLL clock mode
3) When entering the stop mode during the operation of the interval timer function in the sub clock mode or sub PLL clock mode
4) When a request is issued to clear the counter

The same operation is performed when changing to the watch mode as for when changing to the sleep mode.

Figure 12.5-1 Operating Examples of Watch Prescaler



• When setting interval time select bits in the watch prescaler control register (WPCR:WTC1, WTC0) to "11_B" (2^{14}x $2/F_{CL}$)

• WPCR:WTC1,WTC0 : Interval time select bits in watch prescaler control register
• WPCR:WCLR : Watch timer initialization bit in watch prescaler control register
• WPCR:WTIF : Watch interrupt request flag bit in watch prescaler control register
• WPCR:WTIE : Watch interrupt request enable bit in watch prescaler control register
• STBC:SLP : Sleep bit in standby control register
• STBC:STP : Stop bit in standby control register
• WATR:SWT3 to SWT0 : Sub clock oscillation stabilization wait time select bit in oscillation stabilization wait time setup register

■ Setup Procedure Example

The watch prescaler is set up in the following procedure:

● Initial setting

1) Set the interrupt level. (ILR5)
2) Set the interval time. (WPCR:WTC1, WTC0)
3) Enable interrupts. (WPCR:WTIE = 1)
4) Clear the counter. (WPCR:WCLR = 1)

● Interrupt processing

1) Clear the interrupt request flag.(WPCR:WTIF = 0)
2) Arbitrary processing

12.6 Notes on Using Watch Prescaler

Shown below are the precautions that must be followed when using the watch prescaler.
The watch prescaler cannot be used in single clock option product.

■ Notes on Using Watch Prescaler

- When setting the prescaler by program

  The prescaler cannot be recovered from interrupt processing when the watch interrupt request flag bit (WPCR:WTIF) is set to "1" and the interrupt request enable bit is enabled (WPCR:WTIE = 1). Always clear the WTIF bit within the interrupt routine.

- Clearing the watch prescaler

  When the watch prescaler is selected as the count clock of the watchdog timer (WDTC:CS1, CS0= 10_B or CS1, CS0 = 11_B), clearing the watch prescaler also clears the watchdog timer.

- Watch interrupts

  In the main clock stop mode, the watch prescaler performs counting but does not generate the watch prescaler interrupts (IRQ20).

- Peripheral functions receiving clock from the watch prescaler

  If the watch prescaler is cleared when the output of the watch prescaler is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

  The clock for the watchdog timer is also outputted from the initial state. However, as the watchdog timer counter is cleared at the same time as the prescaler counter, the watchdog timer operates in the normal cycles.

12.7 Sample Programs for Watch Prescaler

We provide sample programs that can be used to operate the watch prescaler.

■ Sample Programs for Watch Prescaler

For information about sample programs for the watch prescaler, refer to "■ Sample Programs" in Preface.

■ Setting Methods not Covered by Sample Programs

● How to initialize the watch prescaler

The watch timer initialization bit (WPCR:WCLR) is used.

Control item	Watch timer initialization bit (WCLR)
When initializing watch prescaler	Set the bit to "1".

● How to select the interval time

The watch interrupt interval time select bits (WPCR:WTC1/WTC0) are used to select the interval time.

● Interrupt-related register

The interrupt level is set using the interrupt level register shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
Watch prescaler	Interrupt level register (ILR5) Address: $0007E_H$	#20 Address: $0FFD2_H$

● How to enable/disable/clear interrupts

The interrupt request enable bit (WPCR:WTIE) is used to enable interrupts.

Control item	Interrupt request enable bit (WTIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The watch interrupt request flag (WPCR:WTIF) is used to clear interrupt requests.

Control item	Watch interrupt request flag (WTIF)
To clear an interrupt request	Set the bit to "0".

CHAPTER 13
WATCH COUNTER

This chapter describes the functions and operations of the watch counter.

Code: CM26-00108-2E

13.1 Overview of Watch Counter

The watch counter can generate interrupt requests ranging from min. 125ms to max. 63s intervals.

■ Watch Counter

The watch counter performs counting for the number of times specified in the register by using the selected count clock and generates an interrupt request. The count clock can be selected from the four types shown in Table 13.1-1. The count value can be set to any number from 0 to 63. "When "0" is selected, no interrupt is generated.

When the count cycle is set to 1s and the count value is set to "60", an interrupt is generated every one minute.

Table 13.1-1 Count Clock Types

Count clock	Count cycle when F_{CL} operates at 32.768kHz
$2^{12}/F_{CL}$	125ms
$2^{13}/F_{CL}$	250 ms
$2^{14}/F_{CL}$	500 ms
$2^{15}/F_{CL}$	1s

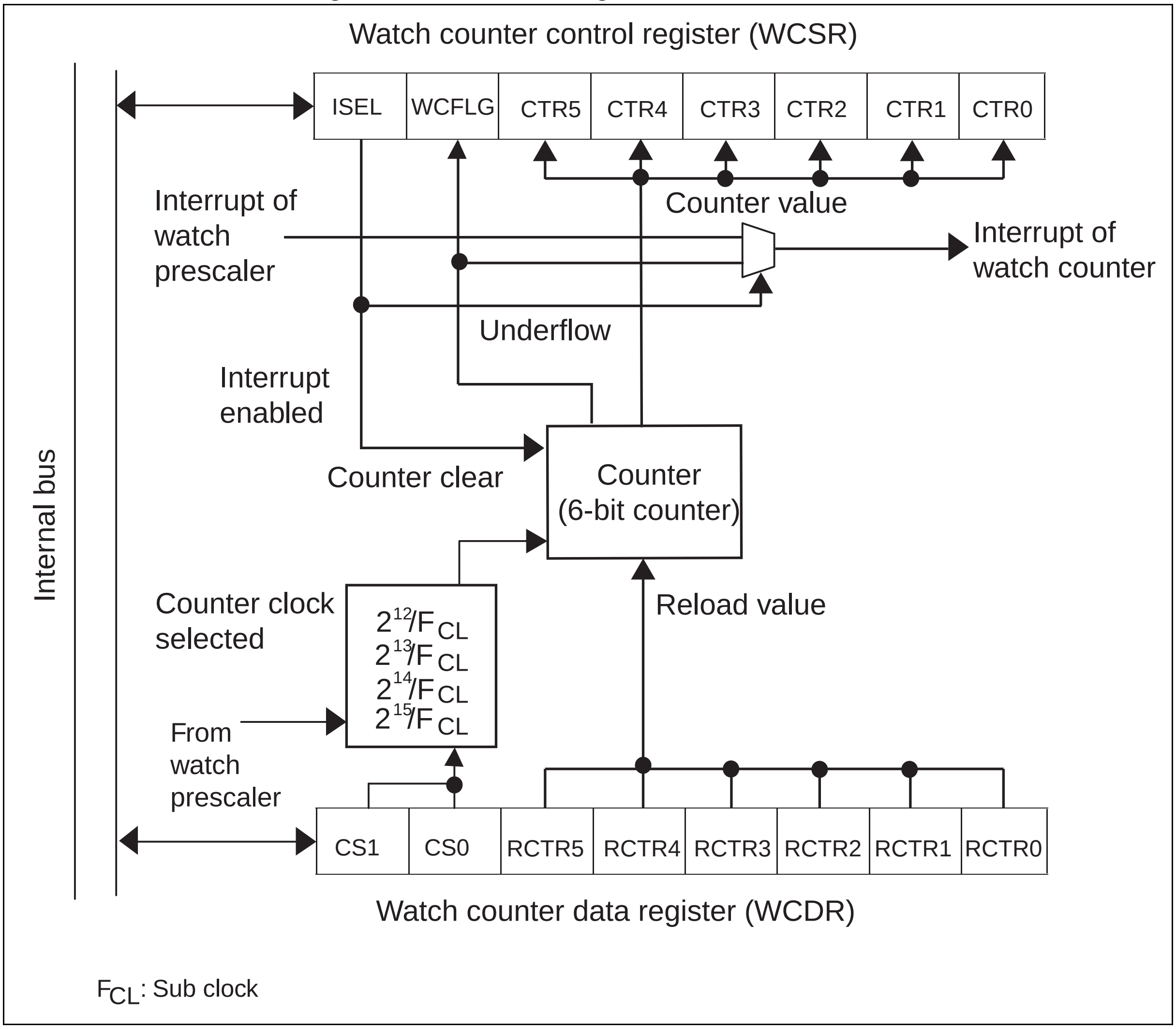
F_{CL}: sub clock

13.2 Configuration of Watch Counter

Figure 13.2-1 shows the block diagram of the watch counter.

■ Block Diagram of Watch Counter

Figure 13.2-1 Block Diagram of Watch Counter



● Counter

This is a 6-bit down-counter that uses the output clock of the watch prescaler as its count clock.

● Watch counter control register (WCSR)

This register controls interrupts and checks the status.

● Watch counter data register (WCDR)

This register sets the interval time and selects the count clock.

■ Input Clock

The watch counter uses the output clock of the watch prescaler as its input clock (count clock).

13.3 Registers of Watch Counter

Figure 13.3-1 shows the registers of the watch counter.

■ Registers of Watch Counter

Figure 13.3-1 Registers Related to Watch Counter

Watch counter data register (WCDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FE3_H$	CS1	CS0	RCTR5	RCTR4	RCTR3	RCTR2	RCTR1	RCTR0	00111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Watch counter control register (WCSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0070_H	ISEL	WCFLG	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0	00000000_B
	R/W	R(RM1)/W	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W: Readable/writable (Read value is the same as write value)

R(RM1),W : Readable/writable
(Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

R/WX : Read only (Readable, writing has no effect on operation)

13.3.1 Watch Counter Data Register (WCDR)

The watch counter data register (WCDR) is used to select the count clock and set the counter reload value.

■ Watch Counter Data Register (WCDR)

Figure 13.3.1-1 Watch Counter Data Register (WCDR)

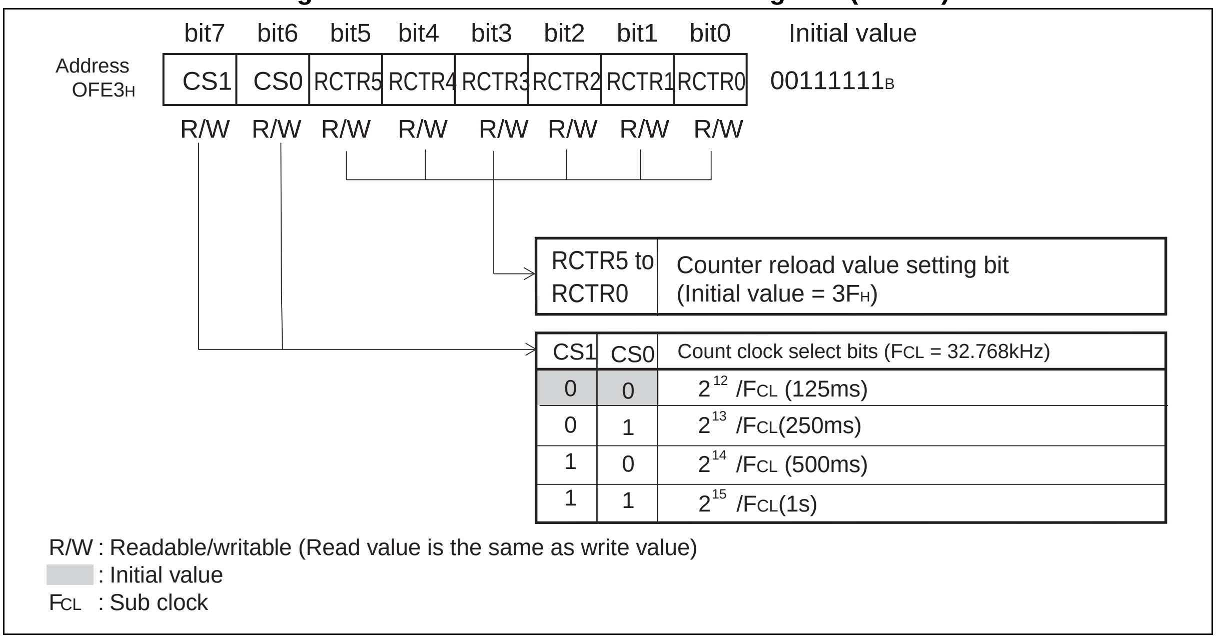


Table 13.3.1-1 Functional Description of Each Bit of Watch Counter Data Register (WCDR)

Bit name		Function
bit7, bit6	CS1, CS0: Count clock select bits	These bits select the clock for the watch counter. $00_B = 2^{12}/F_{CL}$, $01_B = 2^{13}/F_{CL}$, $10_B = 2^{14}/F_{CL}$, $11_B = 2^{15}/F_{CL}$ (F_{CL}: sub clock) These bits should be modified when the WCSR:ISEL bit is "0".
bit5 to bit0	RCTR5 to RCTR0: Counter reload value setting bits	These bits set the counter reload value. If the value is modified during counting, the modified value will become effective upon a reload after the counter underflows. When set to **"0":**No interrupt requests will be generated. If the reload value (RCTR5 to RCTR0) is modified at the same time as an interrupt is generated (WCSR:WCFLG = 1), the correct value will not be reloaded. Therefore, the reload value must be modified before an interrupt is generated, such as when the watch counter is stopped (WCSR:ISEL=0), during the interrupt routine.

13.3.2 Watch Counter Control Register (WCSR)

The watch counter control register (WCSR) is used to control the operation and interrupts of the watch counter. It can also read the count value.

■ Watch Counter Control Register (WCSR)

Figure 13.3.2-1 Watch Counter Control Register (WCSR)

Address 0070H — Initial value 00000000B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Name	ISEL	WCFLG	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0
Access	R/W	R(RM1)/W	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX

CTR5 to CTR0	Counter read bit
	These bits can read the counter value.

WCFLG	Interrupt request flag bit	
	Read	Write
0	No interrupt request generated	Clears this bit
1	An interrupt request generated	No change, no effect on operation

ISEL	Watch counter start & interrupt enable bit
0	Stops watch counter and disables interrupt request of watch counter (Enables interrupt request of watch prescaler)
1	Activates watch counter and enables interrupt request of watch counter (Disables interrupt request of watch prescaler)

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
(shaded) : Initial value

Table 13.3.2-1 Functional Description of Each Bit of Watch Counter Status Register (WCSR)

Bit name		Function
bit7	ISEL: Watch counter start & interrupt request enable bit	This bit activates the watch counter and selects whether to enable interrupts of the watch counter or those of the watch prescaler. When set to **"0"**: The watch counter is cleared and stopped. Moreover, interrupt requests of the watch counter are disabled, while interrupt requests of the watch prescaler are enabled. When set to **"1"**: The interrupt request output of the watch counter is enabled and the counter starts operation. On the other hand, interrupt requests of the watch prescaler are disabled. • Always disable interrupts of the watch prescaler before setting this bit to "1" to select interrupts of the watch counter. • The watch counter performs counting, using an asynchronous clock from the watch prescaler. For this reason, an error of up to one count clock may occur at the beginning of a count cycle, depending on the timing for setting ISEL bit to "1".
bit6	WCFLG: Interrupt request flag bit	This bit is set to "1" when the counter underflows. • When this bit and the ISEL bit are both set to "1", a watch counter interrupt is generated. • Writing "0" clears the bit. • Writing "1" to this bit has no effects on the operation. • "1" is always read in read-modify-write (RMW) instruction.
bit5 to bit0	CTR5 to CTR0: Counter read bits	• These bits can read the counter value during counting. It should be noted that the correct counter value may not be read if a read is attempted while the counter value is being changed. Therefore, read the counter value twice to check if the same value is read on both occasions before using it. • Writing has no effect on the operation.

MB95150/M Series

13.4 Interrupts of Watch Counter

The watch counter outputs interrupt requests when the counter underflows (counter value = 000001_B).

■ Interrupts of Watch Counter

When the counter of the watch counter underflows, the interrupt request flag bit (WCFLG) of the watch counter control register (WCSR) is set to "1". If the interrupt request enable bit (ISEL) of the watch counter is set to "1", an interrupt request of the watch counter is outputted to the interrupt controller.

Table 13.4-1 shows the interrupt control bits and interrupt sources of the watch timer.

Table 13.4-1 Interrupt Control Bits and Interrupt Sources of Watch Timer

Item	Description
Interrupt request flag bit	WCFLG bit of the WCSR register
Interrupt request enable bit	ISEL bit of the WCSR register
Interrupt source	Counter underflow

■ Register and Vector Table Related to Interrupts of Watch Counter

Table 13.4-2 Register and Vector Table Related to Interrupts of Watch Counter

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Watch counter*	IRQ20	ILR5	L20	$FFD2_H$	$FFD3_H$

*: The watch counter shares the same interrupt request number and vector table as the watch prescaler.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

13.5 Explanation of Watch Counter Operations and Setup Procedure Example

The watch counter counts down for the number of times specified in the count value by RCTR5 to RCTR0 bits, using the count clock selected by CS1 and CS0 bits, when the ISEL bit is set to "1". Once the counter underflows, WCFLG bit of the WCSR register is set to "1", generating an interrupt.

■ Setup Procedure of Watch Counter

The setup procedure of the watch counter is described below.

(1) Select the count clock (CS1 and CS0 bits) and set the counter reload value (RCTR5 to RCTR0 bits).

(2) Set the ISEL bit of the WCSR register to "1" to start a down count and enable interrupts. Also disable interrupts of the watch prescaler.
The watch counter performs counting by using a divided clock (asynchronous) from the watch prescaler. An error of up to one count clock may occur at the beginning of a count cycle, depending on the timing for setting the ISEL bit to "1".

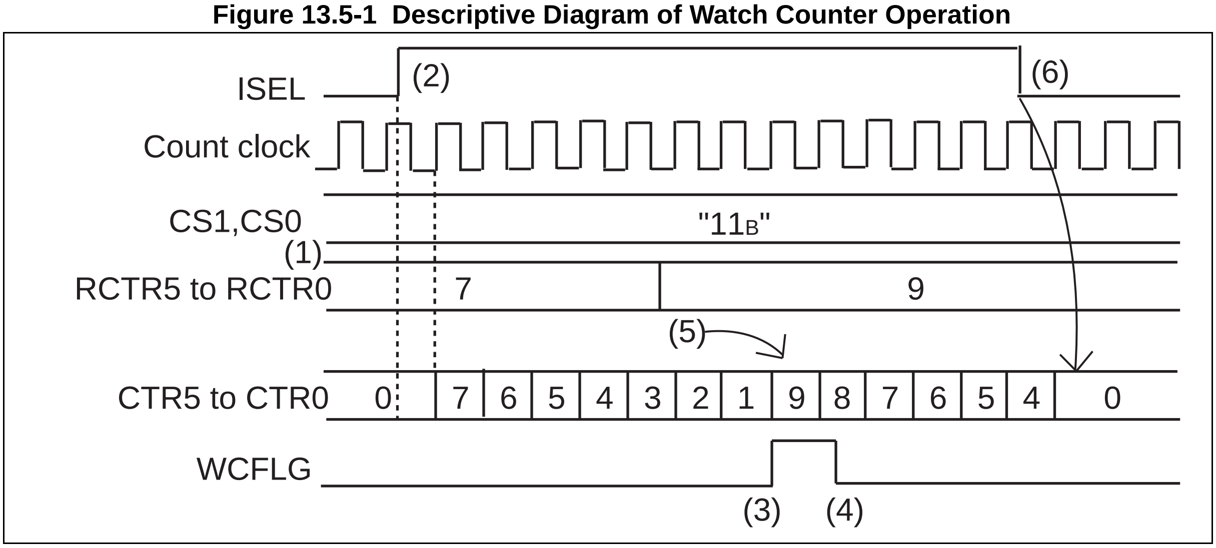
(3) When the counter underflows, the WCFLG bit of the WCSR register is set to "1", generating an interrupt.

(4) Write "0" to the WCFLG bit to clear it.

(5) If RCTR5 to RCTR0 bits are modified during counting, the reload value will be updated during a reload after the counter is set to "1".

(6) When writing "0" to the ISEL bit, the counter becomes "0" and stops operation.

Figure 13.5-1 Descriptive Diagram of Watch Counter Operation



Note:

When the operation is reactivated by WCSR:ISEL=0 after counter stop, please reactivate after confirming reading WCSR:CTR[5:0] twice, and clearing to CTR[5:0]=000000_B.

■ Operation in Sub Clock Stop Mode

When the device enters the sub clock stop mode, the watch counter stops the count operation and the watch prescaler is also cleared. Therefore, the watch counter cannot count the correct value after the sub clock stop mode is cancelled. After the sub clock stop mode is cancelled, the ISEL bit must always be set to "0" to clear the counter before use. In any standby mode other than the sub clock stop mode, the watch counter continues to operate.

■ Operation at the Main Clock Stop Mode

The interrupt is not generated though the clock counter continues the count operation when entering the main clock stop mode. Moreover, the clock counter stops, too, when sub clock oscillation stop bit (SYCC: SUBS) of the system clock control register is set to "1".

■ Setup Procedure Example

The watch counter is set up in the following procedure:

● Initial setting

1) Set the interrupt level. (ILR5)
2) Select the count clock. (WCDR:CS1, CS0)
3) Set the counter reload value. (WCDR:RCTR5 to RCTR0)
4) Activate the watch counter and enable interrupts.(WCSR:ISEL=1)

● Interrupt processing

1) Clear the interrupt request flag.(WCSR:WCFLG=0)
2) Arbitrary processing

13.6 Notes on Using Watch Counter

Shown below are the precautions that must be followed when using the watch counter.

- If the watch prescaler is cleared during the operation of the watch counter, the watch counter may not be able to perform normal operation. When clearing the watch prescaler, set the ISEL bit of the WCSR register to "0" to stop the watch counter in advance.
- When the operation is reactivated by WCSR:ISEL=0 after counter stop, please reactivate after confirming reading WCSR:CTR[5:0] twice, and clearing to CTR[5:0]=000000_B.

13.7 Sample Programs for Watch Counter

We provide sample programs that can be used to operate the watch counter.

■ Sample Programs for Watch Counter

For information about sample programs for the watch counter, refer to "■ Sample Programs" in Preface.

■ Setting Methods not Covered by Sample Programs

● How to enable/stop the watch counter

Use the interrupt request enable bit (WCSR:ISEL).

Control item	Watch timer initialization bit (ISEL)
When enabling watch counter	Set the bit to "1".
When stopping watch counter	Set the bit to "0".

● How to select the count clock

The count clock select bits (WCDR:CS1/CS0) are used to select the clock.

● Interrupt-related register

The interrupt level is set in the interrupt level register shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
Watch counter	Interrupt level register (ILR5) Address: $0007E_H$	#20 Address: $0FFD2_H$

● How to enable/disable/clear interrupts

The interrupt request enable bit (WCSR:ISEL) is used to enable interrupts.

Control item	Interrupt request enable bit (ISEL)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The interrupt request flag (WCSR:WCFLG) is used to clear interrupt requests.

Control item	Interrupt request flag (WCFLG)
To clear an interrupt request	Set the bit to "0".

CHAPTER 14
WILD REGISTER

This chapter describes the functions and operations of the wild register.

Code: CM26-00109-1E

14.1 Overview of Wild Register

**The wild register can be used to patch bugs in the program by using the addresses set in the built-in register and amendment data.
The following section describes the wild register function.**

■ Wild Register Function

The wild register consists of 3 data setup registers, 3 upper-address setup registers, 3 lower-address setup registers, a 1-byte address compare enable register and a 1-byte data test setup register. When certain addresses and modified data are specified in these registers, the ROM data can be replaced with the modified data specified in the registers. Data of up to three different addresses can be modified.

The wild register function can be used to debug the program after creating the mask and patch bugs in the program.

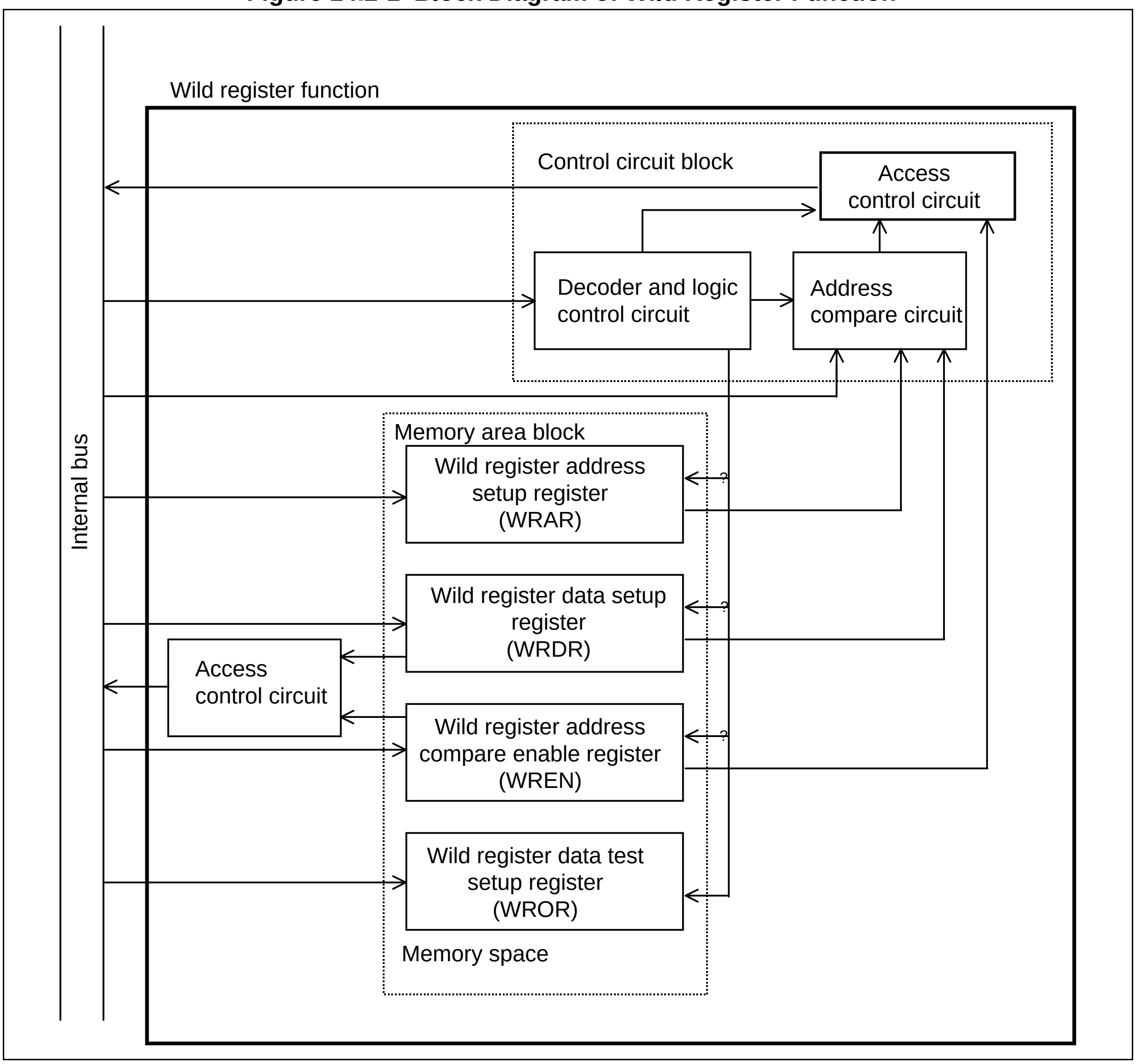
14.2 Configuration of Wild Register

The block diagram of the wild register is shown below. The wild register consists of the following blocks:

- **Memory area block**
  Wild register data setup register (WRDR0 to WRDR2)
  Wild register address setup register (WRAR0 to WRAR2)
  Wild register address compare enable register (WREN)
  Wild register data test setup register (WROR)
- **Control circuit block**

■ Block Diagram of Wild Register Function

Figure 14.2-1 Block Diagram of Wild Register Function



● Memory area block

The memory area block consists of the wild register data setup registers (WRDR), wild register address setup registers (WRAR), wild register address compare enable register (WREN) and wild register data test setup register (WROR). The wild register function is used to specify the addresses and data that need to be replaced. The wild register address compare enable register (WREN) enables the wild register function for each wild register data setup register (WRDR). Moreover, the wild register data test setup register (WROR) enables the normal read function for each wild register data setup register (WRDR).

● Control circuit block

This circuit compares the actual address data with addresses set in the wild register address setup registers (WRDR), and if the values match, outputs the data from the wild register data setup register (WRDR) to the data bus. The control circuit block uses the wild register address compare enable register (WREN) to control the operation.

MB95150/M Series

14.3 Registers of Wild Register

The registers of the wild register include the wild register data setup registers (WRDR), wild register address setup registers (WRAR), wild register address compare enable register (WREN) and wild register data test setup register (WROR).

■ Registers Related to Wild Register

Figure 14.3-1 Registers Related to Wild Register

Wild register data setup registers (WRDR0 to WRDR2)

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WRDR0	$0F82_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
WRDR1	$0F85_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRDR2	$0F88_H$									

Wild register address setup registers (WRAR0 to WRAR2)

Register	Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
WRAR0	$0F80_H$, $0F81_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000_B
WRAR1	$0F83_H$, $0F84_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRAR2	$0F86_H$, $0F87_H$									

		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
		RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Wild register address compare enable register (WREN)

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WREN	0076_H	–	–	Reserved	Reserved	Reserved	EN2	EN1	EN0	00000000_B
		R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

Wild register data test setup register (WROR)

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WROR	0077_H	–	–	Reserved	Reserved	Reserved	DRR2	DRR1	DRR0	00000000_B
		R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R0/W0 : Reserved bit (Write value is "0", read value is "0")
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

■ Wild Register Number

Each wild register address setup register (WRAR) and wild register data setup register (WRDR) has its corresponding wild register number.

Table 14.3-1 Wild Register Numbers Corresponding to Wild Register Address Setup Registers and Wild Register Data Setup Registers

Wild register number	Wild registers address setup register (WRAR)	Wild registers data setup register (WRDR)
0	WRAR0	WRDR0
1	WRAR1	WRDR1
2	WRAR2	WRDR2

14.3.1 Wild Register Data Setup Registers (WRDR0 to WRDR2)

The wild register data setup registers (WRDR0 to WRDR2) use the wild register function to specify the data to be amended.

■ Wild Register Data Setup Registers (WRDR0 to WRDR2)

Figure 14.3-2 Wild Register Data Setup Registers (WRDR0 to WRDR2)

WRDR0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F82_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F85_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F88_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

Table 14.3-2 Functional Description of Each Bit of Wild Register Data Setup Register (WRDR0 to WRDR2)

Bit name		Function
bit7 to bit0	RD7 to RD0: Wild registers data setup bits	These bits specify the data to be amended by the wild register function. • These bits are used to set the amendment data at the address assigned by the wild register address setup register (WRAR). Data is enabled at the address corresponding to each wild register number. • Read access of these bits is enabled only when the corresponding data test setting bit in the wild register data test setup register (WROR) is set to "1".

14.3.2 Wild Register Address Setup Registers (WRAR0 to WRAR2)

The wild register address setup registers (WRAR0 to WRAR2) set the address to be amended by the wild register function.

■ Wild Register Address Setup Registers (WRAR0 to WRAR2)

Figure 14.3-3 Wild Register Address Setup Registers (WRAR0 to WRAR2)

WRAR0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
$0F80_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F81_H$	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRAR1

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
$0F83_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F84_H$	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRAR2

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
$0F86_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F87_H$	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

Table 14.3-3 Functional Description of Each Bit of Wild Register Address Setup Register (WRAR0 to WRAR2)

Bit name		Function
bit15 to bit0	RA15 to RA0: Wild Registers address setting bits	These bits set the address to be amended by the wild register function. These bits are used to specify the address to be allocated. The address is specified in accordance with its corresponding wild register number.

14.3.3 Wild Register Address Compare Enable Register (WREN)

The wild register address compare enable register (WREN) enables/disables the operation of the wild register in accordance with each wild register number.

■ Wild Register Address Compare Enable Register (WREN)

Figure 14.3-4 Wild Register Address Compare Enable Register (WREN)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0076_H	–	–	Reserved	Reserved	Reserved	EN2	EN1	EN0	00000000_B
	R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R0/W0 : Reserved bit (Write value is "0", read value is "0")
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

Table 14.3-4 Functional Description of Wild Register Address Compare Enable Register (WREN)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. • The read value is "0". • Writing has no effect on the operation.
bit5 to bit3	Reserved bit	These bits are reserved. • The read value is "0". • Always set "0".
bit2 to bit0	EN2, EN1, EN0: Wild register address compare enable bits	These bits enable/disable the operation of the wild register. • EN0 corresponds to wild register number 0. • EN1 corresponds to wild register number 1. • EN2 corresponds to wild register number 2. When set to **"0":** disable the operation of the wild register function. When set to **"1":** enable the operation of the wild register function.

14.3.4 Wild Register Data Test Setup Register (WROR)

The wild register data test setup register (WROR) enables/disables reading from the corresponding wild register data setup register (WRDR0 to WRDR2).

■ Wild Register Data Test Setup Register (WROR)

Figure 14.3-5 Wild Register Data Test Setup Register (WROR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0077_H	–	–	Reserved	Reserved	Reserved	DRR2	DRR1	DRR0	00000000_B
	R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R0/W0 : Reserved bit (Write value is "0", read value is "0")
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
– : Undefined

Table 14.3-5 Functional Description of Wild Register Data Test Setup Register (WROR)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. • The read value is "0". • Writing has no effect on the operation.
bit5 to bit3	Reserved bits	These bits are reserved. • The read value is "0". • Always set "0".
bit2 to bit0	DRR2, DRR1, DRR0: Wild registers data test setup bits	These bits enable/disable the normal reading from the corresponding data setup register of the wild register. • DRR0 enables/disables reading from the wild register data setup register (WRDR0). • DRR1 enables/disables reading from the wild register data setup register (WRDR1). • DRR2 enables/disables reading from the wild register data setup register (WRDR2). When set to **"0":** disable reading. When set to **"1":** enable reading.

14.4 Operating Description of Wild Register

This section describes the setup procedure for the wild register.

■ Setup Procedure for Wild Register

Prepare a special program that can read the value to be set in the wild register from external memory (e.g. E^2PROM or FRAM) in the user program before executing the program. The setup method for the wild register is shown below.

It should be noted that this section does not explain how to communicate between the external memory and the device.

- Write the address of the built-in ROM code that will be modified to the wild register address setup register (WRAR0 to WRAR2).
- Write a new code into the corresponding wild register data setup register (WRDR0 to WRDR2).
- Write the corresponding bits to the wild register address compare enable register (WREN) to enable the wild register function.

Table 14.4-1 shows the register setup procedure for the wild register.

Table 14.4-1 Register Setup Procedure for Wild Register

Operating step	Operation	Example operation
1	Read replacement data from outside through its specific communication method.	The built-in ROM code to be modified is in the address "$F011_H$" and the data to be modified is "$B5_H$". Three built-in ROM codes can be modified.
2	Write the replacement address into the wild register address setup register (WRAR0 to WRAR2).	Set Wild register address setup registers (WRAR0 = $F011_H$, WRAR1 = ..., WRAR2 = ...).
3	Write a new ROM code (replacement for the built-in ROM code) to the wild register data setup register (WRDR0 to WRDR2).	Set Wild register data setup registers (WRDR0 = $B5_H$, WRDR1 = ..., WRDR2 = ...).
4	Enable the corresponding bits in the wild register address compare enable register (WREN).	Setting bit0 of the address compare enable register (WREN) to "1" enables the wild register function for the wild register number 0. If the address matches the value set in the address setup register (WRAR), the value of the data setup register (WRDR) will replace the built-in ROM code. When replacing more than one built-in ROM code, enable the corresponding bits of the address compare enable register (WREN).

■ Wild Register Applicable Addresses

The wild register is applicable to all addresses in the address space except "0078_H".

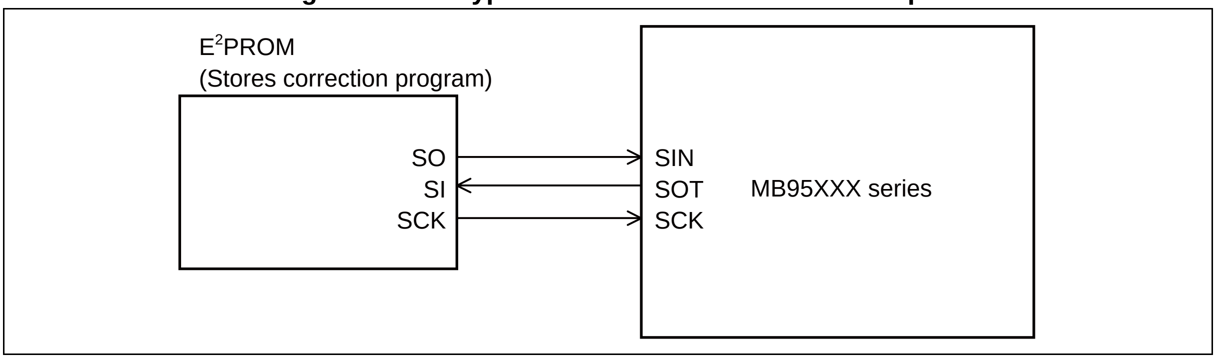
As address "0078_H" is used as a mirror address for the register bank pointer and direct bank pointer, this address cannot be patched.

14.5 Typical Hardware Connection Example

Shown below is a typical hardware connection example applied when using the wild register function.

■ Hardware Connection Example

Figure 14.5-1 Typical Hardware Connection Example



CHAPTER 15

8/16-BIT COMPOUND TIMER

This chapter describes the functions and operations of the 8/16-bit compound timer.

15.1 Overview of 8/16-bit Compound Timer

The 8/16-bit compound timer consists of two 8-bit counters and can be used as two 8-bit timers, or one 16-bit timer if they are connected in cascade.
The 8/16-bit compound timer has the following functions:

- **Interval timer function**
- **PWM timer function**
- **PWC timer function (pulse width measurement)**
- **Input capture function**

■ Interval Timer Function (One-shot Mode)

When the interval timer function (one-shot mode) is selected, the counter starts counting from "00_H" as the timer is started. When the counter value matches the register setting value, the timer output is inverted, the interrupt request occurs, and the count operation is stopped.

■ Interval Timer Function (Continuous Mode)

When the interval timer function (continuous mode) is selected, the counter starts counting from "00_H" as the timer is started. When the counter value matches the register setting value, the timer output is inverted, the interrupt request occurs, and the count operation is continued from "00_H" again. The timer output a square wave as a result of this repeated operation.

■ Interval Timer Function (Free-run Mode)

When the interval timer function (free-run mode) is selected, the counter starts counting from "00_H". When the counter value matches the register setting value, the timer output is inverted and the interrupt request occurs. When the counter continues to count until reaching "FF_H", it restarts counting from "00_H" to continue the counting operation. The timer outputs a square wave as a result of this repeated operation.

■ PWM Timer Function (Fixed-cycle Mode)

When the PWM timer function (fixed-cycle mode) is selected, a PWM signal with a variable "H" pulse width is generated in fixed cycles. The cycle is fixed to "FF_H" during 8-bit operation or "$FFFF_H$" during 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by setting a register.

■ PWM Timer Function (Variable-cycle Mode)

When the PWM timer function (variable-cycle mode) is selected, two 8-bit counters are used to generate an 8-bit PWM signal in any cycles and duty depending on the cycle and "L" pulse width specified by registers.

In this operation mode, the compound timer cannot serve as a 16-bit counter, as two 8-bit counters are used.

■ PWC Timer Function

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured.

In this operation mode, the counter starts counting from "00_H" upon detection of a count start edge of an external input signal and transfers the count value to a register to generate an interrupt upon detection of a count end edge.

■ Input Capture Function

When the input capture function is selected, the counter value is stored in a register upon detection of an edge for an external input signal.

This function is available in either free-run mode or clear mode for count operation.

In the clear mode, the counter starts counting from "00_H" and transfers its value to a register to generate an interrupt upon detection of an edge. In this case, the counter continues to count from "00_H".

In the free-run mode, the counter transfers its value to a register to generate an interrupt upon detection of an edge. In this case, however, the counter continues to count without being cleared.

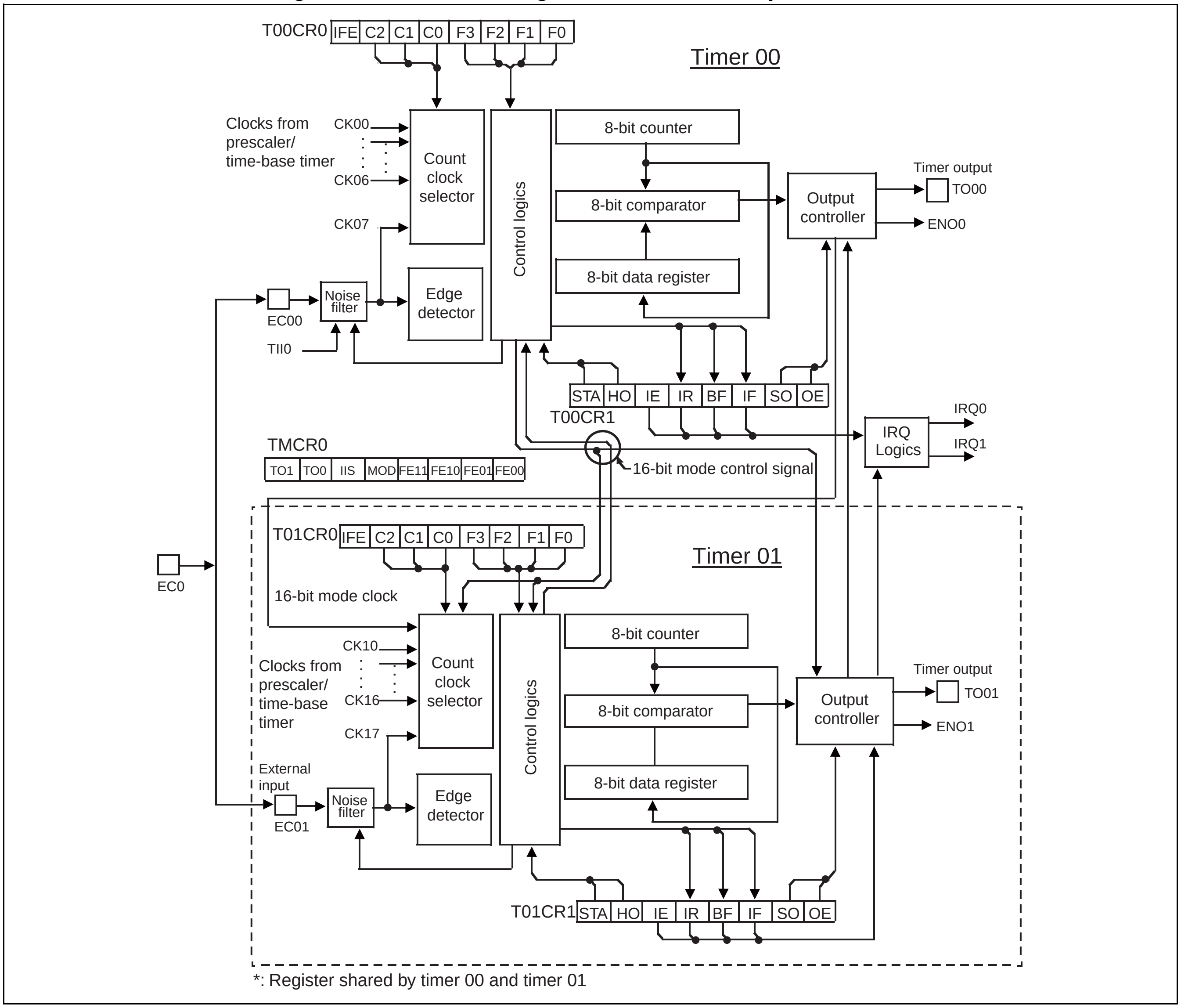
15.2 Configuration of 8/16-bit Compound Timer

The 8/16-bit compound timer consists of the following blocks:

- **8-bit counter × 2 channels**
- **8-bit comparator (including a temporary latch) × 2 channels**
- **8/16-bit compound timer 00/01 data register × 2 channels (T00DR/T01DR)**
- **8/16-bit compound timer 00/01 control status register 0 × 2 channels (T00CR0/T01CR0)**
- **8/16-bit compound timer 00/01 control status register 1 × 2 channels (T00CR1/T01CR1)**
- **8/16-bit compound timer 00/01 timer mode control register (TMCR0)**
- **Output controller × 2 channels**
- **Control logic × 2 channels**
- **Count clock selector × 2 channels**
- **Edge detector × 2 channels**
- **Noise filter × 2 channels**

■ Block Diagram of 8/16-bit Compound Timer

Figure 15.2-1 Block Diagram of 8/16-bit Compound Timer



● 8-bit counter

This counter serves as the basis for various timer operations. It can be used either as two 8-bit counters or as a 16-bit counter.

● 8-bit comparator

The comparator compares the values in the 8/16-bit compound timer 00/01 data register and counter. It incorporates a latch to temporarily store the 8/16-bit compound timer 00/01 data register value.

● 8/16-bit compound timer 00/01 data register

The 8/16-bit compound timer 00/01 data register is used to write the maximum value counted during interval timer or PWM timer operation and to read the count value during PWC timer or input capture operation.

● 8/16-bit compound timer 00/01 control status registers 0 (T00CR0/T01CR0)

These registers are used to select the timer operation mode, select the count clock, and to enable or disable IF flag interrupts.

● 8/16-bit compound timer 00/01 control status registers 1 (T00CR1/T01CR1)

These registers are used to control interrupt flags, timer output, and timer operation.

● 8/16-bit compound timer 00/01 timer mode control register (TMCR0)

This register is used to select the noise filter function, 8-bit or16-bit operation mode, and signal input to timer 00 and to indicate the timer output value.

● Output controller

The output controller controls timer output. The timer output is supplied to the external pin when the pin output has been enabled.

● Control logic

The control logic controls timer operation.

● Count clock selector

The selector selects the counter operation clock signal from among prescaler outputs (machine clock divided signal and time-base timer output).

● Edge detector

The edge detector selects the edge of an external input signal to be used as an event for PWC timer operation or input capture operation.

● Noise filter

This filter serves as a noise filter for external input signals. "H" pulse noise, "L" pulse noise, or "H"/"L"-pulse noise elimination can be selected as the filter function.

● TII0 internal pin (internally connected to the LIN-UART, available only in channel 0)

The TII0 pin serves as the signal input pin for timer 00; it is connected to the LIN-UART inside the chip. For information about how to use the pin, refer to "CHAPTER 22 LIN-UART". Note that the TII0 pin in channel 1 is internally fixed to "0".

■ Input Clock

The 8/16-bit compound timer uses the output clock from the prescaler as its input clock (count clock).

15.3 Channels of 8/16-bit Compound Timer

This section describes the channels of 8/16-bit compound timer.

■ Channels of 8/16-bit Compound Timer

MB95150/M series contains two channels of 8/16-bit compound timer.

In one channel, there are two 8-bit counters. Each counter can be used as two 8-bit timers or one 16-bit timer. The following table lists the external pins and registers corresponding to each channel.

Table 15.3-1 8/16-bit Compound Timer Channels and Corresponding External Pins

Channel	Pin name	Pin function
0	TO00	Timer 00 output
	TO01	Timer 01 output
	EC0	Timer 00 input and timer 01 input
1	TO10	Timer 10 output
	TO11	Timer 11 output
	EC1	Timer 10 input and timer 11 input

Table 15.3-2 8/16-bit Compound Timer Channels and Corresponding Registers

Channel	Register name	Registers
0	T00CR0	Timer 00 control status register 0
	T01CR0	Timer 01 control status register 0
	T00CR1	Timer 00 control status register 1
	T01CR1	Timer 01 control status register 1
	T00DR	Timer 00 data register
	T01DR	Timer 01 data register
	TMCR0	Timer 00/01 timer mode control register
1	T10CR0	Timer 10 control status register 0
	T11CR0	Timer 11 control status register 0
	T10CR1	Timer 10 control status register 1
	T11CR1	Timer 11 control status register 1
	T10DR	Timer 10 data register
	T11DR	Timer 11 data register
	TMCR1	Timer 10/11 timer mode control register

The following sections describe only the 8/16-bit compound timer in channel 0.

The other channels are the same as channel 0. The 2-digit number in the pin names and register names corresponds to channel and timer. The upper number corresponds to channel and the lower number corresponds to timer.

15.4 Pins of 8/16-bit Compound Timer

This section describes the pins related to the 8/16-bit compound timer.

■ Pins Related to 8/16-bit Compound Timer

The external pins related to the 8/16-bit compound timer are TO00, TO01, EC0, and EC1. TII0 is for internal chip connection.

● TO00 pin

TO00:

This pin serves as the timer output pin for timer 00 during 8-bit operation or for timers 00 and 01 during 16-bit operation. When the output is enabled (T00CR1:OE = 1) in interval timer, PWM timer, or PWC timer function, the pin is set for output automatically regardless of the port direction register (DDR2:bit2) to serve as the timer output TO00 pin.

The output remains undefined value when the input capture function has been selected enabling output.

● TO01 pin

TO01:

This pin serves as the timer output pin for timer 01 during 8-bit operation. When the output is enabled (T01CR1:OE = 1) in interval timer, PWM timer (fixed cycle mode), or PWC timer function, the pin is set for output automatically regardless of the port direction register (DDR2:bit3) to serve as the timer output TO01 pin.

The output remains undefined value during 16-bit operation when the PWM timer function (variable-cycle mode) or input capture function has been selected enabling output.

● EC0 pin

The EC0 pin is connected to the EC00 and EC01 internal pins.

EC00 internal pin:

This pin serves as the external count clock input pin for timer 00 when the interval timer or PWM timer function has been selected, or as the signal input pin for timer 00 when the PWC timer or input capture function has been selected. The pin cannot be set as the external count clock input pin when the PWC timer or input capture function has been selected.

To use this input feature, set the port direction register (DDR2:bit4) to "0" to set the pin as an input port.

EC01 internal pin:

This pin serves as the external count clock input pin for timer 01 when the interval timer or PWM timer function has been selected or the signal input pin for timer 01 when the PWC timer or input capture function has been selected. The pin cannot be set as the external count clock input pin when the PWC timer or input capture function has been selected.

This input is not used during 16-bit operation. The input can be used as well when the PWM timer function has been selected (variable-cycle mode).

To use this input feature, set the port direction register (DDR2:bit4) to set the pin as an input port.

■ Block Diagram of Pins Related to 8/16-bit Compound Timer

Figure 15.4-1 Block Diagram of Pins (EC0, TO00 and TO01) Related to 8/16-bit Compound Timer

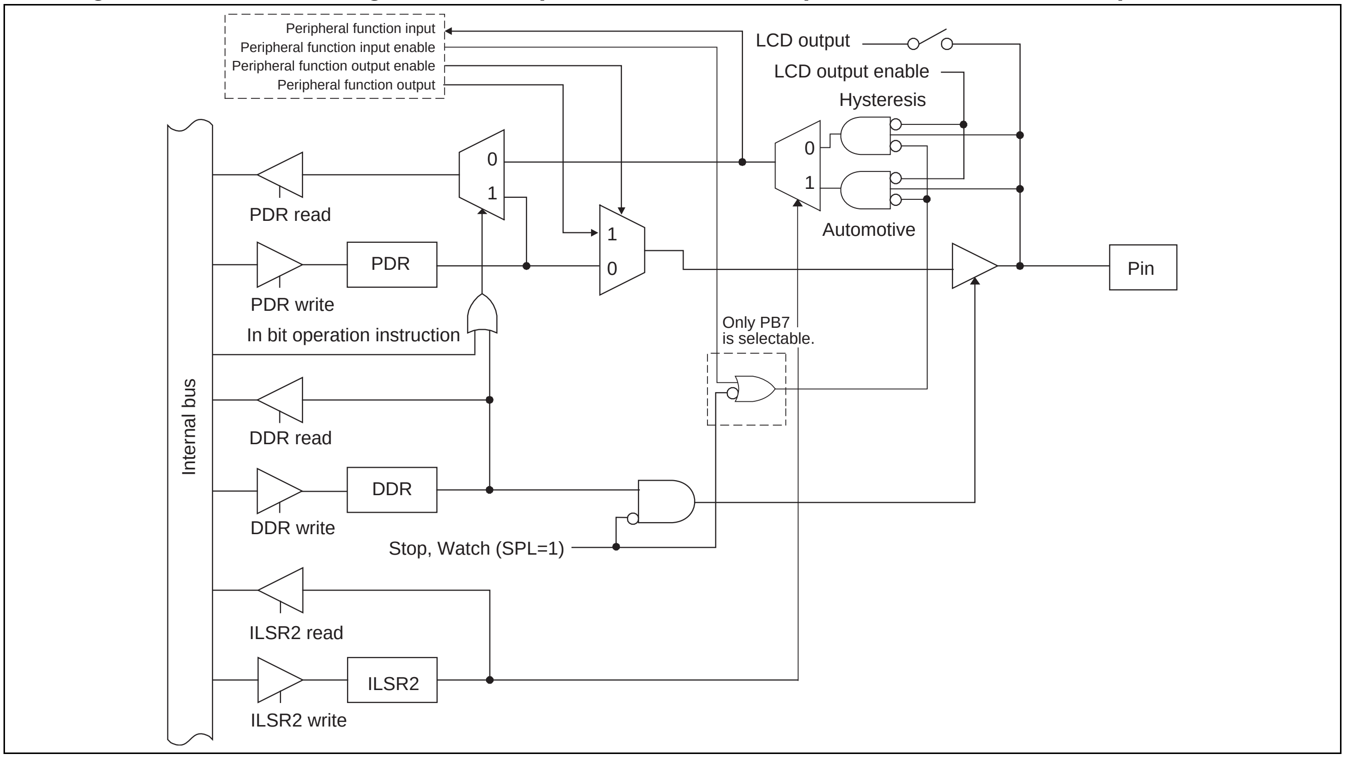
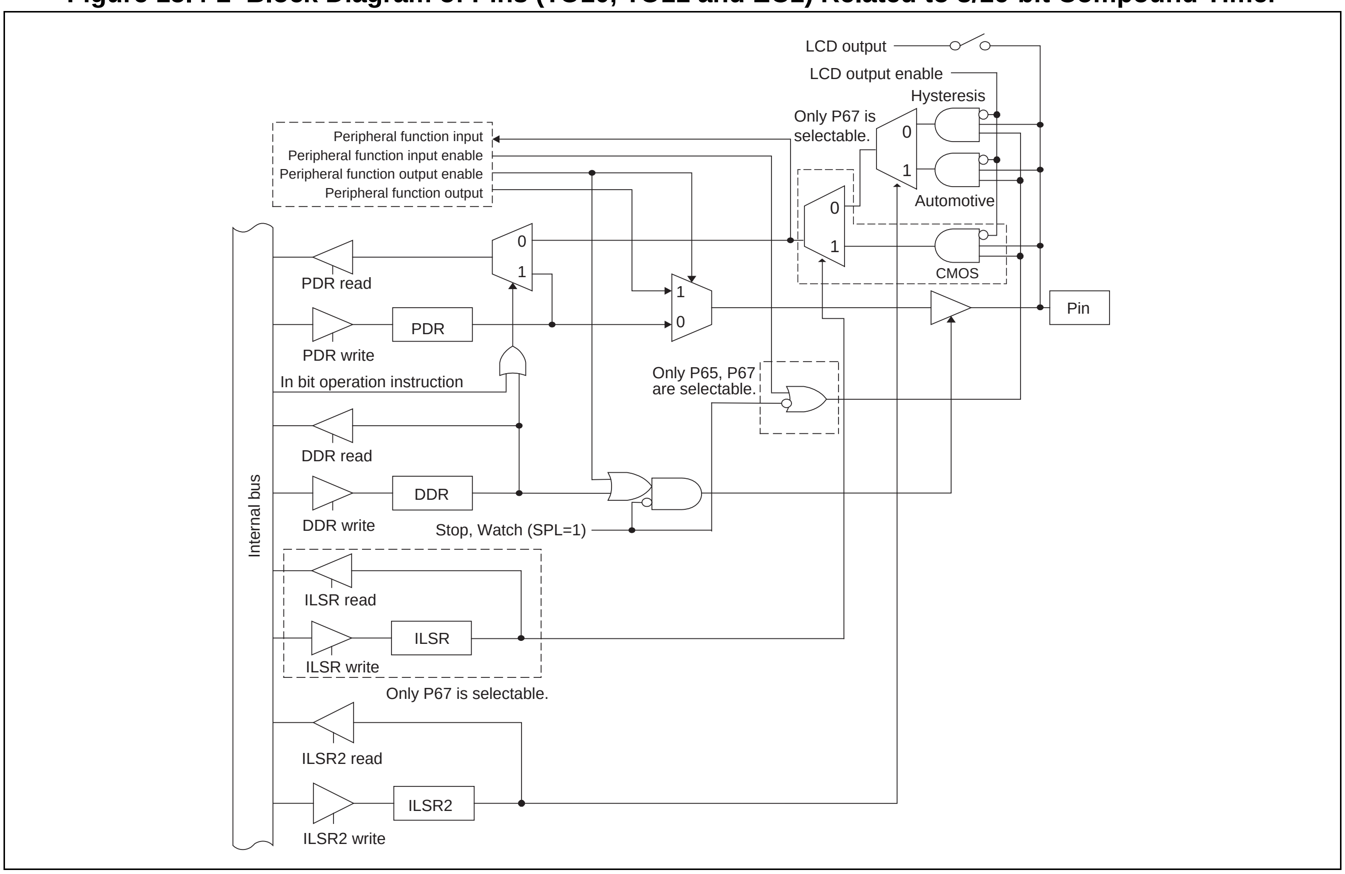


Figure 15.4-2 Block Diagram of Pins (TO10, TO11 and EC1) Related to 8/16-bit Compound Timer



15.5 Registers of 8/16-bit Compound Timer

This section describes the registers related to the 8/16-bit compound timer.

■ Registers Related to 8/16-bit Compound Timer

Figure 15.5-1 Registers Related to 8/16-bit Compound Timer

8/16-bit compound timer 00/01 control status register 0 (T00CR0/T01CR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01CR0	$0F92_H$	IFE	C2	C1	C0	F3	F2	F1	F0	00000000_B
T00CR0	$0F93_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit compound timer 00/01 control status register 1 (T00CR1/T01CR1)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01CR1	0036_H	STA	HO	IE	IR	BF	IF	SO	OE	00000000_B
T00CR1	0037_H	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

8/16-bit compound timer 00/01 data register (T00DR/T01DR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01DR	$0F94_H$	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000_B
T00DR	$0F95_H$	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	

8/16-bit compound timer 00/01 timer mode control register (TMCR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
TMCR0	$0F96_H$	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00	00000000_B
		R/WX	R/WX	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R,W : Readable/Writable (Read value is different from write value)

15.5.1 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)

The 8/16-bit compound timer 00/01 control status register 0 (T00CR0/T01CR0) selects the timer operation mode, selects the count clock, and enables or disables IF flag interrupts. The T00CR0 and T01CR0 registers correspond to timers 00 and 01, respectively.

■ 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)

Figure 15.5-2 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01CR0 0F92H T00CR0 0F93H	IFE	C2	C1	C0	F3	F2	F1	F0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

F3	F2	F1	F0	Timer operation mode select bits
0	0	0	0	Interval timer (one-shot mode)
0	0	0	1	Interval timer (continuous mode)
0	0	1	0	Interval timer (free-run mode)
0	0	1	1	PWM timer (fixed-cycle mode)
0	1	0	0	PWM timer (variable-cycle mode)
0	1	0	1	PWC timer ("H" pulse = rising to falling)
0	1	1	0	PWC timer ("L" pulse = falling to rising)
0	1	1	1	PWC timer (cycle = rising to rising)
1	0	0	0	PWC timer (cycle = falling to falling)
1	0	0	1	PWC timer ("H" pulse = rising to falling; Cycle = rising to rising)
1	0	1	0	Input capture (rising, free-run counter)
1	0	1	1	Input capture (falling, free-run counter)
1	1	0	0	Input capture (both edges, free-run counter)
1	1	0	1	Input capture (rising, counter clear)
1	1	1	0	Input capture (falling, counter clear)
1	1	1	1	Input capture (both edges, counter clear)

C2	C1	C0	Count clock select bits
0	0	0	1 x MCLK (machine clock)
0	0	1	1/2 x MCLK (machine clock)
0	1	0	1/4 x MCLK (machine clock)
0	1	1	1/8 x MCLK (machine clock)
1	0	0	1/16 x MCLK (machine clock)
1	0	1	1/32 x MCLK (machine clock)
1	1	0	$1/2^7$ x F_{CH}
1	1	1	External clock

IFE	IF flag interrupt enable
0	IF flag interrupt disabled
1	IF flag interrupt enabled

R/W : Readable/Writable (Read value is the same as write value)

▭ : Initial value

Table 15.5-1 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) (1 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>IFE:
IF flag interrupt enable</td><td>This bit enables or disables IF flag interrupts.
Setting this bit to "0": disables IF flag interrupts.
Setting this bit to "1": an IF flag interrupt request is outputted when both the IE bit (T00CR1/T01CR1:IE) and the IF flag (T00CR1/T01CR1:IF) are set to "1".</td></tr>
<tr><td>bit6
to
bit4</td><td>C2, C1, C0:
Count clock select bits</td><td>These bits select the count clock.
• The count clock is generated by the prescaler. Refer to "6.12 Operating Explanation of Prescaler".
• Write access to these bits is nullified during timer operation (T00CR1/T01CR1:STA = 1).
• The clock selection of T01CR0 (timer 01) is nullified during 16-bit operation.
• These bits cannot be set to "111_B" when the PWC or input capture function is used. An attempt to write "111_B" with the PWC or input capture function in use resets the bits to "000_B". The bits are also reset to "000_B" if the timer enters the input capture operation mode with the bits set to "111_B".
<table>
<tr><th>C2</th><th>C1</th><th>C0</th><th>Count clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1 × MCLK (machine clock)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1/2 × MCLK (machine clock)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1/4 × MCLK (machine clock)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1/8 × MCLK (machine clock)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1/16 × MCLK (machine clock)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1/32 × MCLK (machine clock)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$1/2^7 \times F_{CH}$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock</td></tr>
</table></td></tr>
</table>

Table 15.5-1 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) (2 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit3
to
bit0</td><td>F3, F2, F1, F0:
Timer operation mode select bits</td><td>These bits select the timer operation mode.
• The PWM timer function (variable-cycle mode; F3, F2, F1, F0 = 0100_B) is set by either the T00CR0 (timer 00) register or T01CR0 (timer 01) register. In this case, the other register is set to F3, F2, F1, F0 = 0100_B automatically when the timer starts operation (T00CR1/T01CR1: STA= 1).
• The MOD bit is set to "0" automatically when the timer set for 16-bit operation (TMCR0:MOD = 1) starts operation (T00CR1/T01CR1:STA = 1) in the PWM timer function (variable-cycle mode).
• Write access to these bits is nullified during timer operation (T00CR1/T01CR1:STA = 1).

<table>
<tr><th>F3</th><th>F2</th><th>F1</th><th>F0</th><th>Timer operation mode select bits</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Interval timer (one-shot mode)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Interval timer (continuous mode)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Interval timer (free-run mode)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>PWM timer (fixed-cycle mode)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PWM timer (variable-cycle mode)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>PWC timer ("H" pulse = rising to falling)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>PWC timer ("L" pulse = falling to rising)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>PWC timer (cycle = rising to falling)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>PWC timer (cycle = falling to rising)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PWC timer
("H" pulse = rising to falling; Cycle = rising to rising)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Input capture
(rising, free-run counter)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Input capture
(falling, free-run counter)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Input capture
(both edges, free-run counter)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Input capture
(rising, counter clear)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Input capture
(falling, counter clear)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Input capture
(both edges, counter clear)</td></tr>
</table>
</td></tr>
</table>

15.5.2 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)

8/16-bit compound timer 00/01 control status register 1 (T00CR1/T01CR1) controls the interrupt flag, timer output, and timer operations. T00CR1 and T01CR1 registers correspond to timers 00 and 01, respectively.

■ 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)

Figure 15.5-3 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01CR1 0036H T00CR1 0037H	STA	HO	IE	IR	BF	IF	SO	OE	00000000_B
	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

OE	Timer output enable bit
0	Timer output disabled
1	Timer output enabled

SO	Timer output initial value bit
0	Timer initial value "0"
1	Timer initial value "1"

IF	Timer reload/overflow flag	
	Read	Write
0	No reload or overflow	Flag clear
1	Reload and overflow	No effect on operation

BF	Data register full flag
0	Measurement data absent in data register
1	Measurement data present in data register

IR	Pulse width measurement complete and edge detection flag	
	Read	Write
0	Measurement complete, edge undetected	Flag clear
1	Measurement complete, edge detected	No effect on operation

IE	Interrupt request bit
0	Interrupt disabled
1	Interrupt enabled

HO	Timer pause bit
0	Timer operable
1	Timer paused

STA	Timer operation enable bit
0	Timer stopped
1	Timer operable

R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
(shaded) : Initial value (value 0 in each table above)

Table 15.5-2 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) (1 / 2)

Bit name		Function
bit7	STA: Timer operation enable bit	This bit enables or stops timer operation. **Writing "0"**: stops the timer operation and sets the count value to "00_H". • When the PWM timer function (variable-cycle mode) has been selected (T00CR0/T01CR0: F3, F2, F1, F0 = 0100_B), the STA bit can be used to enable or disable timer operation from within either the T00CR1 (timer 00) or T01CR1 (timer 01) register. In this case, the STA bit in the other register is set to the same value automatically. • During 16-bit operation (TMCR0:MOD = 1), use the STA bit in the T00CR1 (timer 00) register to enable or disable timer operation. In this case, the STA bit in the other register is set to the same value automatically. **Writing "1"**: allows timer operation to start from count value "00_H". • Set this bit to "1" after setting the count clock select bits (T00CR0/T01CR0:C2, C1, C0), timer operation select bits (T00CR0/T01CR0:F3, F2, F1, F0), timer output initial value bit (T00CR1/ T01CR1:SO), 16-bit mode enable bit (TMCR0:MOD), and filter function select bits (TMCR0: FE11, FE10, FE01, FE00).
bit6	HO: Timer suspend bit	This bit suspends or resumes timer operation. • Writing "1" to this bit during timer operation suspends the timer operation. • Writing "0" to the bit when timer operation has been enabled (T00CR1/T01CR1:STA = 1) resumes the timer operation. • When the PWM timer function (variable-cycle mode) has been selected (T00CR0/T01CR0: F3, F2, F1, F0=0100_B), the HO bit can be used to suspend or resume timer operation from within either the T00CR1 (timer 00) or T01CR1 (timer 01) register. In this case, the HO bit in the other register is set to the same value automatically. • During 16-bit operation (TMCR0:MOD = 1), use the HO bit in the T00CR1 (timer 00) register to suspend or resume timer operation. In this case, the STA bit in the other register is set to the same value automatically.
bit5	IE: Interrupt request enable bit	This bit enables or disables the output of interrupt requests. **Writing "0"**: disables interrupt request. **Writing "1"**: outputs an interrupt request when the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) or timer reload/overflow flag (T00CR1/ T01CR1:IF) is "1". Note, however, that an interrupt request from the timer reload/overflow flag (T00CR1/T01CR1:IF) is not outputted unless the IF flag interrupt enable (T00CR0/ T01CR0:IFE) bit is also set to "1".
bit4	IR: Pulse width measurement completion/edge detection flag	This bit shows the completion of pulse width measurement or the detection of an edge. • The bit is set to "1" upon completion of pulse width measurement when the PWC timer function has been selected. • The bit is set to "1" upon detection of an edge when the input capture function has been selected. • The bit is "0" when any timer function other than the PWC timer and input capture functions has been selected. • This bit always returns "1" to a read modify write (RMW) instruction. • The IR bit in T01CR1 (timer 01) register is set to "0" during 16-bit operation. • Writing "0" to the bit sets it to "0". • An attempt to write "1" to the bit is ignored.

Table 15.5-2 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) (2 / 2)

Bit name		Function
bit3	BF: Data register full flag	• This bit is set to "1" when a count value is stored in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) upon completion of pulse width measurement in PWC timer function. • This bit is set to "0" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is read during 8-bit operation. • The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) holds data with this bit containing "1". Even when the next edge is detected with this bit containing "1", the count value is not transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and thus the next measurement result is lost. However, as the exception, when the "H" pulse and cycle measurement (T00CR0/T01CR0: F3, F2, F1, F0= 1001_B) is selected, the "H" pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) with this bit set to "1". The cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the bit set to "1". For cycle measurement, therefore, the "H" pulse measurement result must be read before the cycle is completed. Note also that the result of "H" pulse measurement or cycle measurement is lost unless read before the completion of the next "H" pulse. • The BF bit in the T00CR1 (timer 00) register is set to "0" when the T01DR (timer 01) register is read during 16-bit operation. • The BF bit in T01CR1 (timer 01) register is set to "0" during 16-bit operation. • This bit is "0" when any timer function other than the PWC timer function has been selected. • Writing to this bit has no effects on the operation.
bit2	IF: Timer reload/overflow flag	This bit detects a match with a count value or a counter overflow. • The bit is set to "1" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) value matches the count value during interval timer function (both one-shot and continuous mode) or PWM timer function (variable-cycle mode). • The bit is set to "1" when a counter overflow occurs during PWC or input capture function. • This bit always returns "1" to a read-modify-write (RMW) instruction. • Writing "0" to the bit sets it to "0". • Writing "1" to this bit has no effects on the operation. • The bit is "0" when the PWM function (variable-cycle mode) has been selected. • The IF bit in the T01CR1 (timer 01) register is "0" during 16-bit operation.
bit1	SO: Timer output initial value bit	Writing to this bit sets the timer output (TMCR0:TO1/TO0) initial value. The value in this bit is reflected in the timer output when the timer operation enable bit (T00CR1/T01CR1:STA) changes from "0" to "1". • During 16-bit operation (TMCR0:MOD = 1), use the SO bit in the T00CR1 (timer 00) register to set the timer output initial value. In this case, the value of the S bit in the other register is meaningless. • An attempt to write to this bit is nullified during timer operation (T00CR1/T01CR1:STA = 1). During 16-bit operation, however, a value can be written to the SO bit in the T01CR1 (timer 01) register even during timer operation but it has no direct effect on the timer output. • The value of this bit is meaningless when the PWM timer function (either fixed-cycle or variable-cycle mode) or input capture function has been selected.
bit0	OE: Timer output enable bit	This bit enables or disabled timer output. **Writing "0"**: prevents the timer output from being supplied to the external pin. In this case, the external pin serves as a general-purpose port. **Writing "1"**: supplies timer output (TMCR0:TO1/TO0) to the external pin.

15.5.3 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

The 8/16-bit compound timer 00/01 timer mode control register ch.0 (TMCR0) selects the filter function, 8-bit or 16-bit operation mode, and signal input to timer 00 and to indicate the timer output value. This register serves for both of timers 00 and 01.

■ 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

Figure 15.5-4 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

TMCR0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F96_H$	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00	00000000_B
	R/WX	R/WX	R/W	R/W	R/W	R/W	R/W	R/W	

FE01	FE00	Timer 00 filter function select bits
0	0	No filtering
0	1	Removing "H" pulse noise
1	0	Removing "L" pulse noise
1	1	Removing "H"/"L" pulse noise

FE11	FE10	Timer 01 filter function select bits
0	0	No filtering
0	1	Removing "H" pulse noise
1	0	Removing "L" pulse noise
1	1	Removing "H"/"L" pulse noise

MOD	8-bit/16-bit operation mode select bit
0	8-bit operation
1	16-bit operation

IIS	Timer 00 internal signal select bit
0	Selecting external signal (EC00) as timer 00 input
1	Selecting internal signal (TII0) as timer 00 input

TO0	Timer 00 output bit
0	Output value of timer 00
1	

TO1	Timer 01 output bit
0	Output value of timer 01
1	

R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
(shaded) : Initial value

Table 15.5-3 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) (1 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>TO1:
Timer 01 output bit</td><td>This bit indicates the output value of timer 01. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value in the bit changes depending on the selected timer function.
• Writing to this bit has no effect on the operation.
• The value in the bit remains undefined during 16-bit operation when the PWM timer function (variable-cycle mode) or input capture function has been selected.
• When the timer stops operation (T00CR1/T01CR1:STA = 0) in interval timer or PWC timer function, this bit holds the last value.
• When the timer stops operation in PWM timer function (fixed-cycle mode), this bit holds the last value.
• When the timer operation mode select bit (T00CR0/T01CR0: F3, F2, F1, F0) is changed with the timer being stopped, the bit indicates the last value of timer operation if the same timer operation has ever been performed or otherwise contains "0".</td></tr>
<tr><td>bit6</td><td>TO0:
Timer 00 output bit</td><td>This bit indicates the output value of timer 00. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value in the bit changes depending on the selected timer function.
• Writing to this bit has no effect on the operation.
• The value in the bit remains undefined when the input capture function has been selected.
• When the timer stops operation (T00CR1/T01CR1:STA = 0) in interval timer, PWM timer (variable-cycle mode), or PWC timer function, this bit holds the last value.
• When the timer stops operation in PWM timer function (fixed-cycle mode), this bit holds the last value.
• When the timer operation mode select bit (T00CR0/T01CR0: F3, F2, F1, F0) is changed with the timer being stopped, the bit indicates the last value of timer operation if the same timer operation has ever been performed or otherwise contains "0".</td></tr>
<tr><td>bit5</td><td>IIS:
Timer 00 internal signal select bit</td><td>This bit selects the signal input to timer 00 when the PWC timer or input capture function has been selected.
Writing "0": selects the external signal (EC00) as the signal input for timer 00.
Writing "1": selects the internal signal (TII0) as the signal input for timer 00.</td></tr>
<tr><td>bit4</td><td>MOD:
16-bit mode enable bit</td><td>This bit selects 8-bit or 16-bit operation mode.
Writing "0": allows timers 00 and 01 to operate as separate 8-bit timers.
Writing "1": allows timers 00 and 01 to operate as a 16-bit timer.
• This bit is set to "0" automatically when the timer starts operation (T00CR1/T01CR1:STA=1) in PWM timer mode (variable-cycle mode).
• Write access to this bit is nullified during timer operation (T00CR1:STA = 1 or T01CR1:STA = 1).</td></tr>
<tr><td>bit3, bit2</td><td>FE11, FE10:
Timer 01 filter function select bits</td><td>These bits select the filter function for the external signal (EC01) to timer 01 when the PWC timer or input capture function has been selected.
<table><tr><th>FE11</th><th>FE10</th><th>Timer 01 filter function select bits</th></tr><tr><td>0</td><td>0</td><td>No filtering</td></tr><tr><td>0</td><td>1</td><td>Removing "H" pulse noise</td></tr><tr><td>1</td><td>0</td><td>Removing "L" pulse noise</td></tr><tr><td>1</td><td>1</td><td>Removing "H"/"L" pulse noise</td></tr></table>• Write access to these bits is nullified during timer operation (T01CR1:STA = 1).
• The settings of the bits have no effect on operation when the interval timer or PWM timer function has been selected (filter function does not operate.).</td></tr>
</table>

Table 15.5-3 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) (2 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit1, bit0</td><td>FE01, FE00:
Timer 00 filter function select bits</td><td>These bits select the filter function for the external signal (EC00) to timer 00 when the PWC timer or input capture function has been selected.
<table><tr><th>FE01</th><th>FE00</th><th>Timer 00 filter function</th></tr><tr><td>0</td><td>0</td><td>No filtering</td></tr><tr><td>0</td><td>1</td><td>Removing "H" pulse noise</td></tr><tr><td>1</td><td>0</td><td>Removing "L" pulse noise</td></tr><tr><td>1</td><td>1</td><td>Removing "H"/"L" pulse noise</td></tr></table>• An attempt to write to these bits is nullified during timer operation (T00CR1:STA = 1).
• The settings of these bits have no effect on operation when the interval timer or PWM timer function has been selected (filter function does not operate.).</td></tr>
</table>

15.5.4 8/16-bit Compound Timer 00/01 Data Register ch.0 (T00DR/T01DR)

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to write the maximum value counted during interval timer or PWM timer operation and to read the count value during PWC timer or input capture operation. The T00DR and T01DR registers correspond to timers 00 and 01, respectively.

■ 8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR)

Figure 15.5-5 8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01DR	$0F94_H$	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000_B
T00DR	$0F95_H$	R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	

R, W: Readable/Writable (Read value is different from write value)

● Interval timer function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to set the interval time. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value of this register is transferred to the latch in the 8-bit comparator and the counter starts counting. When the count value matches the value held in the latch in the 8-bit comparator, the value of this register is transferred again to the latch and the count value is reset to "00_H" to continue to count.

The current count value can be read from this register.

An attempt to write "00_H" to this register is disabled in interval timer function.

In 16-bit operation, set the upper data to T01DR and lower data to T00DR. And, write and read T01DR and T00DR in this order.

● PWM timer functions (fixed-cycle)

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to set "H" pulse width time. When the timer starts operation (T00CR1/T01CR1:STA=1), the value of this register is transferred to the latch in the 8-bit comparator and the counter starts counting from timer output "H". When the count value matches the value held in the latch, the timer output becomes "L" and the counter continues to count until the count value reaches "FF_H". When an overflow occurs, the value of this register is transferred again to the latch in the 8-bit comparator and the counter performs the next cycle of counting.

The current value can be read from this register. In 16-bit operation, set the upper data to T01DR and lower data to T00DR. And, write and read T01DR and T00DR in this order.

- PWM timer functions (variable-cycle)

The 8/16-bit compound timer 00 data register (T00DR) and 8/16-bit compound timer 01 data register (T01DR) are used to set "L" pulse width timer and cycle, respectively. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value of each register is transferred to the latch in the 8-bit comparator and two counters start counting from timer output "L". When the T00DR value held in the latch matches the timer 00 counter value, the timer output becomes "H" and the counting continues until the T01DR value held in the latch matches the timer 01 counter value. When the T01DR value held in the latch of the 8-bit comparator matches the timer 01 counter value, the values of these registers are transferred again to the latch and the next PWM cycle of counting is performed continuously.

The current count value can be read from this register.

In 16-bit operation, set the upper data and lower data to T01DR and T00DR, respectively. And, write and read T01DR and T00DR in this order.

- PWC timer function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to read PWC measurement results. When PWC measurement is completed, the counter value is transferred to this register and the BF bit is set to "1".

When the 8/16-bit compound timer 00/01 data register is read, the BF bit is set to "0". Transfer to the 8/16-bit compound timer 00/01 data register is not performed with the BF bit containing "1".

As the exception, when the "H" pulse and cycle measurement (T00CR0/T01CR0:F3, F2, F1, F0 = 1001_B) is selected, the "H" pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1", but the cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1". For cycle measurement, therefore, the "H" pulse measurement result must be read before the cycle is completed. Note also that the result of "H" pulse measurement or cycle measurement is lost unless read before the completion of the next "H" pulse.

When reading the 8/16-bit compound timer 00/01 data register, be careful not to clear the BF bit unintentionally.

Writing to the 8/16-bit compound timer 00/01 data register updates the stored measurement data with the write value. Therefore, do not perform a write operation. In 16-bit operation, the upper data and lower data are transferred to T01DR and T00DR, respectively. Read T01DR and T00DR in this order.

- Input capture function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to read input capture results. When a specified edge is detected, the counter value is transferred to the 8/16-bit compound timer 00/01 data register.

Writing a value to the data register updates the measurement data stored there with that value. Therefore, do not write to the 8/16-bit compound timer 00/01 data register. In 16-bit operation, the upper data and lower data are transferred to T01DR and T00DR, respectively. Read T01DR and T00DR in this order.

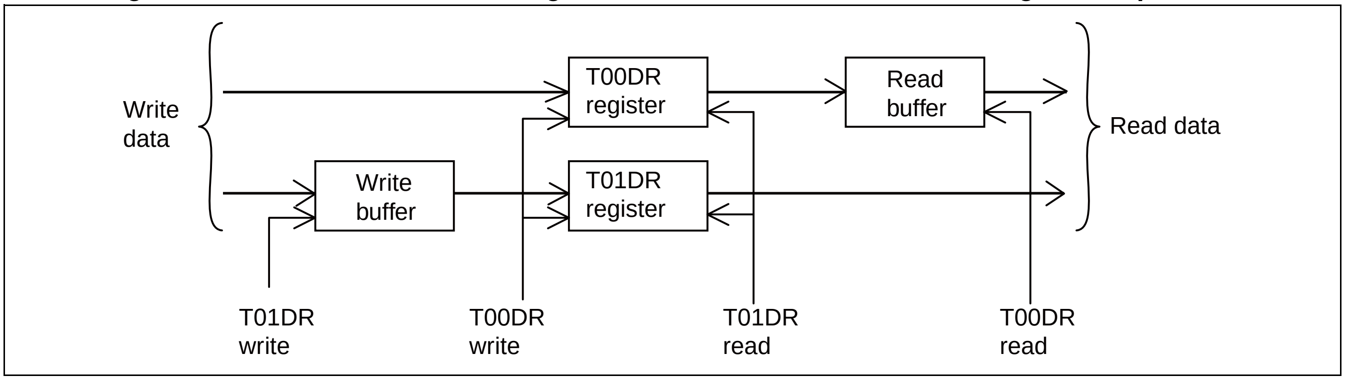
● Read and write operations

Read and write operations of T00DR and T01DR are performed in the following manner during 16-bit operation and PWM timer function (variable-cycle).

- Read from T01DR: Read access from the register also involves storing the T00DR value into the internal read buffer.
- Read from T00DR: Read from the internal read buffer.
- Write to T01DR: Write to the internal write buffer.
- Write to T00DR: Write access to the register also involves storing the value of the internal write buffer into T01DR.

Figure 15.5-6 shows the T00DR and T01DR registers read from and written to during 16-bit operation.

Figure 15.5-6 T00DR and T01DR registers read from and written to during 16-bit operation



15.6 Interrupts of 8/16-bit Compound Timer

The 8/16-bit compound timer generates the following types of interrupts to each of which an interrupt number and interrupt vector are assigned.

- **Timer 00 interrupt**
- **Timer 01 interrupt**

■ Timer 00 Interrupt

Table 15.6-1 explains the timer 00 interrupt and its source.

Table 15.6-1 Timer 00 Interrupt

Item	Description		
Interrupt generating condition	Comparison match in interval timer function or PWM timer function (variable-cycle mode) has been selected	Overflow in PWC timer function or input capture function	Completion of measurement in PWC timer function or edge detection in input capture function
Interrupt flag	T00CR1:IF	T00CR1:IF	T00CR1:IR
Interrupt enable	T00CR1:IE and T00CR0:IFE	T00CR1:IE and T00CR0:IFE	T00CR1:IE

■ Timer 01 Interrupt

Table 15.6-2 explains the timer 01 interrupt and its cause.

Table 15.6-2 Timer 01 Interrupt

Item	Description		
Interrupt generating condition	Comparison match in interval timer function or PWM timer function (variable-cycle mode) has been selected Excluded during 16-bit operation	Overflow in PWC timer function or input capture function Excluded during 16-bit operation	Completion of measurement in PWC timer function or edge detection in input capture function Excluded during 16-bit operation
Interrupt flag	T01CR1:IF	T01CR1:IF	T01CR1:IR
Interrupt enable	T01CR1:IE and T01CR0:IFE	T01CR1:IE and T01CR0:IFE	T01CR1:IE

■ Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer

Table 15.6-3 Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer

Interrupt source	Interrupt request No.	Interrupt level setup register		Vector table address	
		Register	Setting bit	Upper	Lower
Timer 00	IRQ5	ILR1	L05	$FFF0_H$	$FFF1_H$
Timer 01	IRQ6	ILR1	L06	$FFEE_H$	$FFEF_H$
Timer 10*	IRQ22	ILR5	L22	$FFCE_H$	$FFCF_H$
Timer 11	IRQ14	ILR3	L14	$FFDE_H$	$FFDF_H$

*: 8/16-bit compound timer (ch.1) shares the same interrupt request number and vector table as the external interrupt circuit (ch.12 to ch.15).

The request numbers and vector tables of all peripheral functions are listed in "APPENDIX B Table of Interrupt Causes".

15.7 Operating Description of Interval Timer Function (One-shot Mode)

This section describes the operations of the interval timer function (one-shot mode) for the 8/16-bit compound timer.

■ Operation of Interval Timer Function (One-shot Mode)

The compound timer requires the register settings shown in Figure 15.7-1 to serve as the interval timer function.

Figure 15.7-1 Settings of Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	0	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	❍	❍
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval timer (counter compare value)							

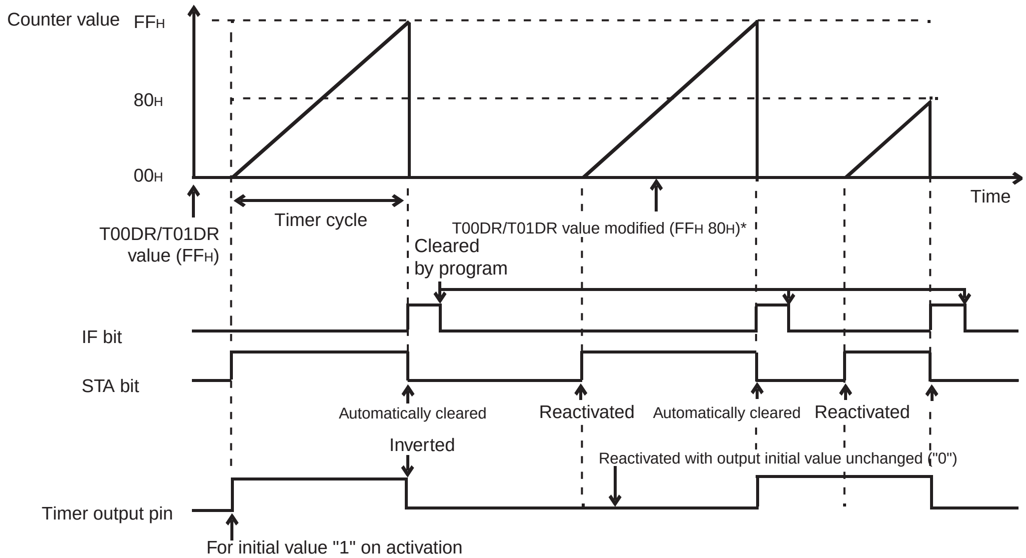
❍: Used bit
×: Unused bit
1: Set "1"
0: Set "0"

In interval timer function (one-shot mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00_H" at the rising edge of a selected count clock signal. When the counter value matches the value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1" and the start bit (T00CR0/T00CR1:STA) is set to "0", and then the count operation stops.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator when the counter starts counting. Writing "00_H" to the 8/16-bit compound timer 00/01 data register is prohibited.

Figure 15.7-2 shows the operation of the interval timer function in the 8-bit operation.

Figure 15.7-2 Operation of Interval Timer Function in 8-bit Mode (Timer 0)



*: If the T00DR/T01DR data register value is modified during operation, the new value is used from the next active cycle.

15.8 Operating Description of Interval Timer Function (Continuous Mode)

This section describes the interval timer function (continuous mode operation) of the 8/16-bit compound timer.

■ Operation of Interval Timer Function (Continuous Mode)

The compound timer requires the register settings shown in Figure 15.8-1 to serve as the interval timer function (continuous mode).

Figure 15.8-1 Settings for Counter Function (8-bit Mode)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	0	1
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	❍	❍
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval time (counter compare value)							

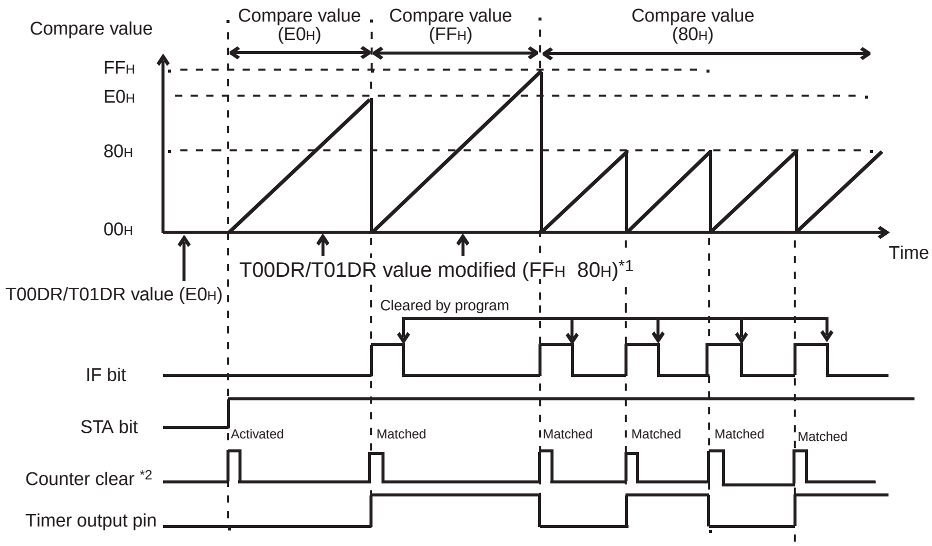
❍: Used bit
×: Unused bit
1: Set "1"
0: Set "0"

In interval timer function (continuous mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00_H" at the rising edge of a selected count clock signal. When the counter value matches the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1", and the counter continues to count by restarting at "00_H". The timer outputs a square wave as a result of this continuous operation.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected. Writing "00_H" to the 8/16-bit compound timer 00/01 data register is disabled during the count operation.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

Figure 15.8-2 Operating Diagram of Interval Timer Function (Continuous Mode)



*1: If the T00DR/T01DR data register value is modified during operation, the new value is used from the next active cycle.

*2: The counter is cleared and the data register settings are loaded into the comparison data latch when a match is detected at each point during activation.

15.9 Operating Description of Interval Timer Function (Free-run Mode)

This section describes the operation of the interval timer function (free-run mode) for the 8/16-bit compound timer.

■ Operation of Interval Timer Function (Free-run Mode)

The compound timer requires the settings shown in Figure 15.9-1 to serve as the interval timer function (free-run mode).

Figure 15.9-1 Settings for Interval Timer Function (Free-run Mode)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	1	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	❍	❍
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval time (counter compare value)							

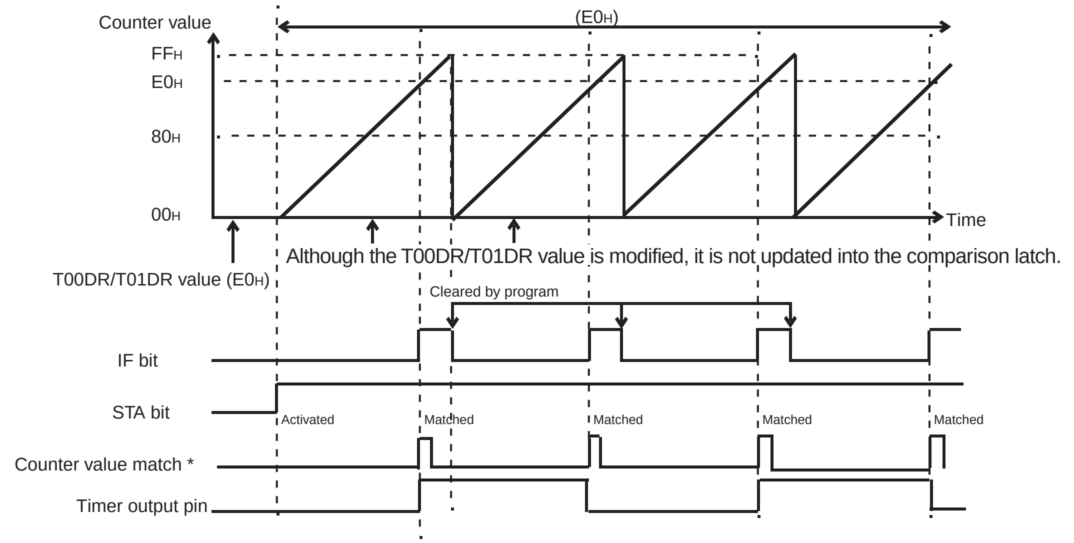
❍: Used bit
×: Unused bit
1: Set "1"
0: Set "0"

In interval timer function (free-run mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00_H" at the rising edge of a selected count clock signal. When the counter value matches the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted and the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The counter continues to count, and when the count value reaches "FF_H", it restarts counting at "00_H" to continue. The timer outputs a square wave as a result of this continuous operation.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected. Writing "00_H" to the 8/16-bit compound timer 00/01 data register is prohibited.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

Figure 15.9-2 Operating Diagram of Interval Timer Function (Free-run Mode)



*: The counter is not cleared and the data register settings are not reloaded into the comparison data latch when a match is detected at each point during activation.

15.10 Operating Description of PWM Timer Function (Fixed-cycle mode)

This section describes the operation of the PWM timer function (fixed-cycle mode) for the 8/16-bit compound timer.

■ Operation of PWM Timer Function (Fixed-cycle Mode)

The compound timer requires the settings shown in Figure 15.10-1 to serve as the PWM timer function (fixed-cycle mode).

Figure 15.10-1 Settings for PWM Timer Function (Fixed-cycle Mode)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	1	1
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	×	×	×	×	×	×
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets "H" pulse width (compare value)							

❍: Used bit
×: Unused bit
1: Set "1"
0: Set "0"

In PWM timer function (fixed-cycle mode), a fixed cycle PWM signal in a variable "H" pulse width is outputted from the timer output pin (TO00/TO01). The cycle is fixed to "FF_H" in 8-bit operation or "$FFFF_H$" in 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR).

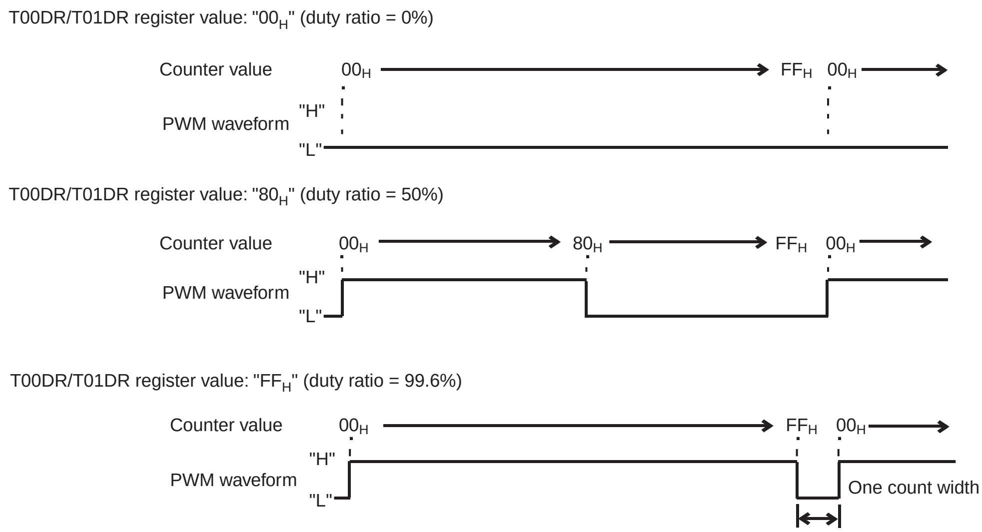
This function has no effect on the interrupt flag (T00CR1/T01CR1:IF). As each cycle always starts with "H" pulse output, the timer output initial value setting bit (T00CR1/T01CR1:SO) is meaningless.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

The "H" pulse is one count clock shorter than the setting value in the output waveform immediately after activation of the timer (write "1" to the STA bit).

Figure 15.10-2 Operating Diagram of PWM Timer Function (Fixed-cycle Mode)



Note: When the PWM function has been selected, the timer output pin holds the level used when the counter stops (T00CR0/T01CR0:STA = 0).

15.11 Operating Description of PWM Timer Function (Variable-cycle Mode)

This section describes the operations of the PWM timer function (variable-cycle mode) for the 8/16-bit compound timer.

■ Operation of PWM Timer Function (Variable-cycle Mode)

The compound timer requires the settings shown in Figure 15.11-1 to serve as the PWM timer function (variable-cycle mode).

Figure 15.11-1 Settings for PWM Timer Function (Variable-cycle Mode)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	1	0	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	×	×
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	×	❍	❍	❍	❍
T00DR	Sets "L" pulse width (compare value)							
T01DR	Sets the cycle of PWM waveform (compare value)							

❍: Used bit
×: Unused bit
1: Set "1"
0: Set "0"

In PWM timer function (variable-cycle mode), both timers 00 and 01 are used when the cycle is specified by the 8/16-bit compound timer 01 data register (T01DR), and the "L" pulse width is specified by the 8/16-bit compound timer 00 data register (T00DR), any cycle and duty PWM signal is generated from the timer output bit (TO00).

For this function, the compound timer cannot serve as a 16-bit counter as the two 8-bit counters are used.

Enabling timer operation (by setting either T00CR1:STA = 1 or T01CR1:STA = 1) sets the mode bit (TMCR0:MOD) to "0". As the first cycle always begins with "L" pulse output, the timer initial value setting bit (T00CR1/T01CR1:SO) is meaningless.

The interrupt flag (T00CR1/T01CR1:IF) is set when each 8-bit counter matches the value in the corresponding 8/16-bit compound timer 00/01 data register (T00DR/T01DR).

The 8/16-bit compound timer 00/01 data register value is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a comparison match with each counter value is detected.

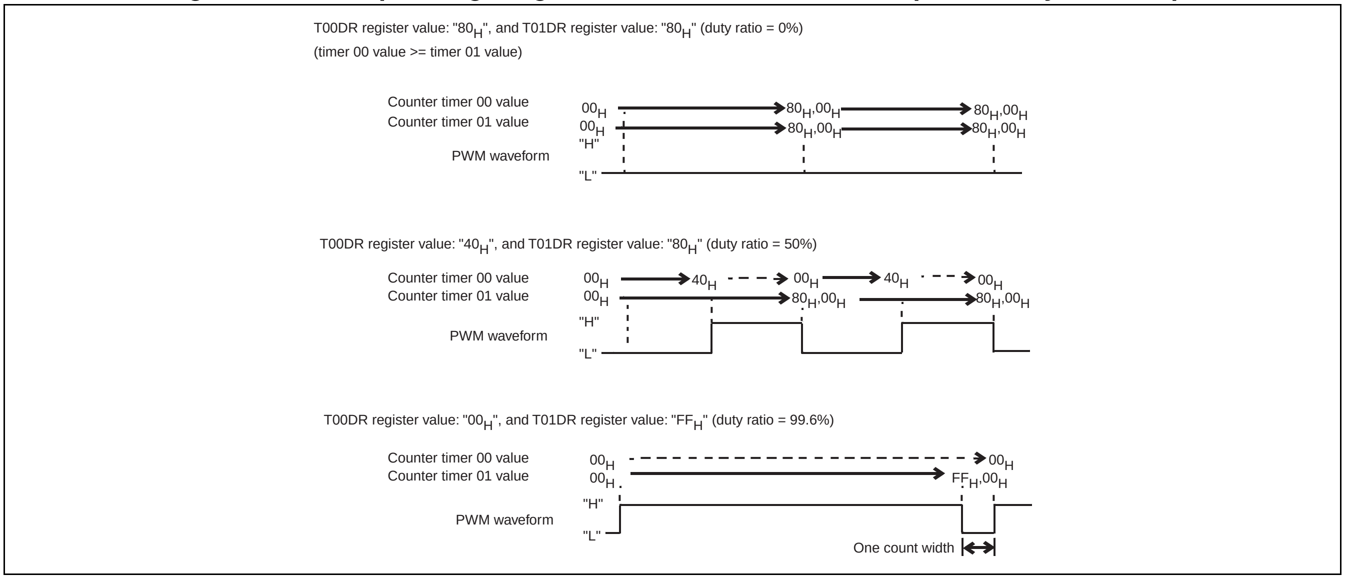
"H" is not outputted when the "L" pulse width setting value is greater than the cycle setting value.

The count clock must be selected for both of timers 00 and 01. Selecting different count clocks, however, is prohibited.

When the timer stops operation, the timer output bit (TMCR0:TO0) holds the last output value.

If the 8/16-bit compound timer 00/01 data register is written over during operation, the written data will be effective from the cycle immediately after the detection of a synchronous match.

Figure 15.11-2 Operating Diagram of PWM Timer Function (Variable-cycle Mode)



15.12 Operating Description of PWC Timer Function

This section describes the operations of the PWC timer function for the 8/16-bit compound timer.

■ Operation of PWC Timer Function

The compound timer requires the settings shown in Figure 15.12-1 to serve as the PWC timer function.

Figure 15.12-1 Settings for PWC Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	❍	❍	❍	❍
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	❍	❍	❍	❍	×
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	❍	❍	❍	❍	❍	❍
T00DR/T01DR	Holds pulse width measurement value							

❍: Used bit
×: Unused bit
1: Set "1"

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured. The edges to start and end counting are selected by timer operation mode setting (T00CR0/T01CR0:F3, F2, F1, F0).

In this operation mode, the counter starts counting from "00_H" upon detection of the specified count start edge of an external input signal. Upon detection of the specified count end edge, the count value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) and buffer full flag (T00CR1/T01CR1:BF) are set to "1". The buffer full flag is set to "0" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is read from.

The 8/16-bit compound timer 00/01 data register holds data with the buffer full flag set to "1". Even when the next edge is detected at this time, the next measurement result is lost as the count value is not transferred to the 8/16-bit compound timer 00/01 data register.

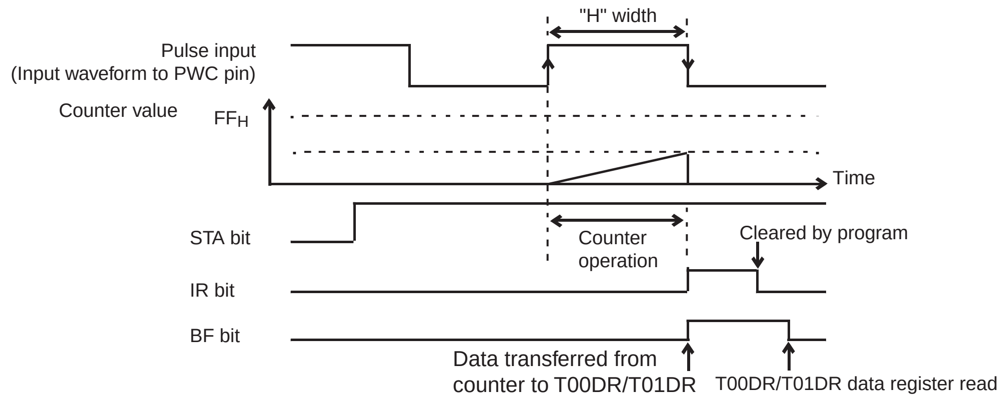
As the exception, when the H-pulse and cycle measurement (T00CR0/T01CR0:F3, F2, F1, F0 = 1001_B) is selected, the H-pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1", but the cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1". For cycle measurement, therefore, the H-pulse measurement result must be read before the cycle is completed. Note also that the result of H-pulse measurement or cycle measurement is lost unless read before the completion of the next H pulse.

To measure the time exceeding the value of the counter, you can use software to count the number of occurrences of a counter overflow. When the counter causes an overflow, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of times the overflow occurs. Note also that an overflow toggles the timer output. The timer output initial value can be set by the timer output initial value bit (T00CR1/T01CR1:SO).

When the timer stops operation, the timer output bit (TMCR0:TO1/TO0) holds the last value.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) must be nullified if an interrupt occurs before the timer is activated (before "1" is written to the STA bit).

Figure 15.12-2 Operating Diagram of PWC Timer (Example of H-pulse Width Measurement)



15.13 Operating Description of Input Capture Function

This section describes the operations of the input capture function for the 8/16-bit compound timer.

■ Operation of Input Capture Function

The compound timer requires the settings shown in Figure 15.13-1 to serve as the input capture function.

Figure 15.13-1 Settings for Input Capture Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	❍	❍	❍	❍
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	❍	×	❍	×	×
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	×	×	❍	❍	❍	❍	❍	❍
T00DR/T01DR	Holds pulse width measurement value							

❍: Used bit
×: Unused bit
1: Set "1"

When the input capture function is selected, the counter value is stored to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) upon detection of an edge of the external signal input. The edge to be detected is selected by timer operation mode setting (T00CR0/T01CR0:F3, F2, F1, F0).

This function is available in either free-run mode or clear mode, which can be selected by timer operation mode setting.

In clear mode, the counter starts counting from "00_H". When the edge is detected, the counter value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the interrupt flag (T00CR1/T01CR1:IR) is set to "1", and the counter continues to count by restarting at "00_H".

When the edge is detected in free-run mode, the counter value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) is set to "1". In this case, the counter continues to count without being cleared.

This function has no effect on the buffer full flag (T00CR1/T01CR1:BF).

To measure the time exceeding the value of the counter, software can be used to count the number of occurrences of a counter overflow. When the counter causes an overflow, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of times the overflow occurs.

The capture value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) must be nullified if an interrupt occurs before the timer is activated (before "1" is written 2to the STA bit).

When the timing at which the 8/16-bit compound timer captures a counter value is the detection of either edge of the external input signal (T00CR0/T01CR0:F3-F0=1100_B or 1111_B), the operations in falling edge detection vary according to the level of the external input signal as explained below.

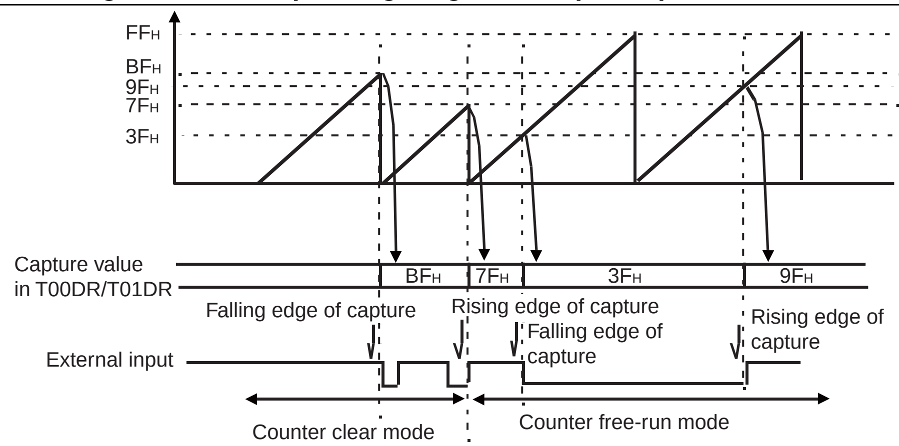
- External input signal level: H

  In both free-run mode and clear mode, the first falling edge is ignored, no counter value is transferred to the data register (T00DR/T01DR), and the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) is not set. In addition, in clear mode, the counter is not cleared either.

- External input signal level: L

  The 8/16-bit compound timer starts edge detection from the first rising edge.

Figure 15.13-2 Operating Diagram of Input Capture Function



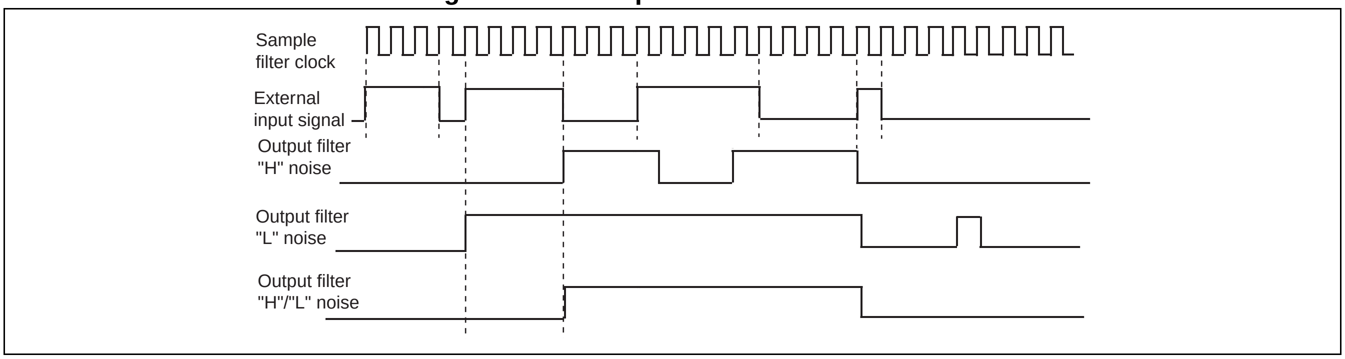
MB95150/M Series

15.14 Operating Description of Noise Filter

This section describes the operations of the noise filter for the 8/16-bit compound timer.

When the input capture or PWC timer function has been selected, a noise filter can be used to eliminate the pulse noise of the signal from the external input pin (EC0/EC1). "H"-pulse noise, "L"-pulse noise, or "H"/"L"-pulse noise elimination can be selected depending on the register setting (TMCR0:FE11, FE10, FE01, FE00). The maximum pulse width from which to eliminate noise is three machine clock cycles. When the filter function is active, the signal input is subject to a delay of four machine clock cycles.

Figure 15.14-1 Operation of Noise Filter



15.15 States in Each Mode during Operation

This section describes how the 8/16-bit compound timer behaves when the microcontroller enters watch mode or stop mode or when a suspend (T00CR1/T01CR1:HO = 1) request is issued during operation.

■ When Interval Timer, Input Capture, or PWC Function Has Been Selected

Figure 15.15-1 shows how the counter value changes when transition to watch mode or stop mode or a suspend request occurs during operation of the 8/16-bit compound timer.

The counter stops operation while holding the value when transition to stop mode or watch mode occurs. When the stop mode or watch mode is canceled by an interrupt, the counter resumes operation with the last value held. So the first interval time and external clock count are incorrect. After releasing from stop mode or watch mode, be sure to initialize the counter value.

Figure 15.15-1 Operations of Counter in Standby Mode or in Pause (Not Serving as PWM Timer)

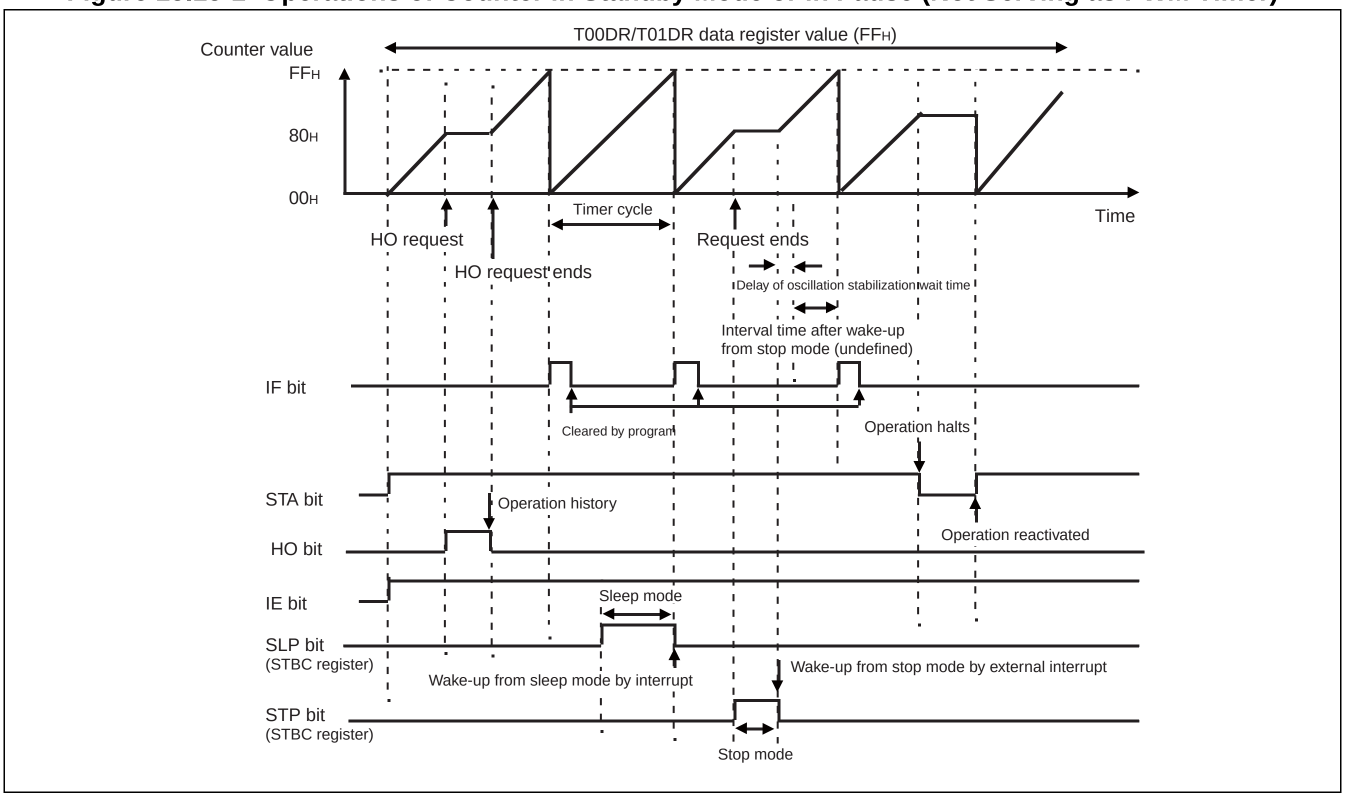
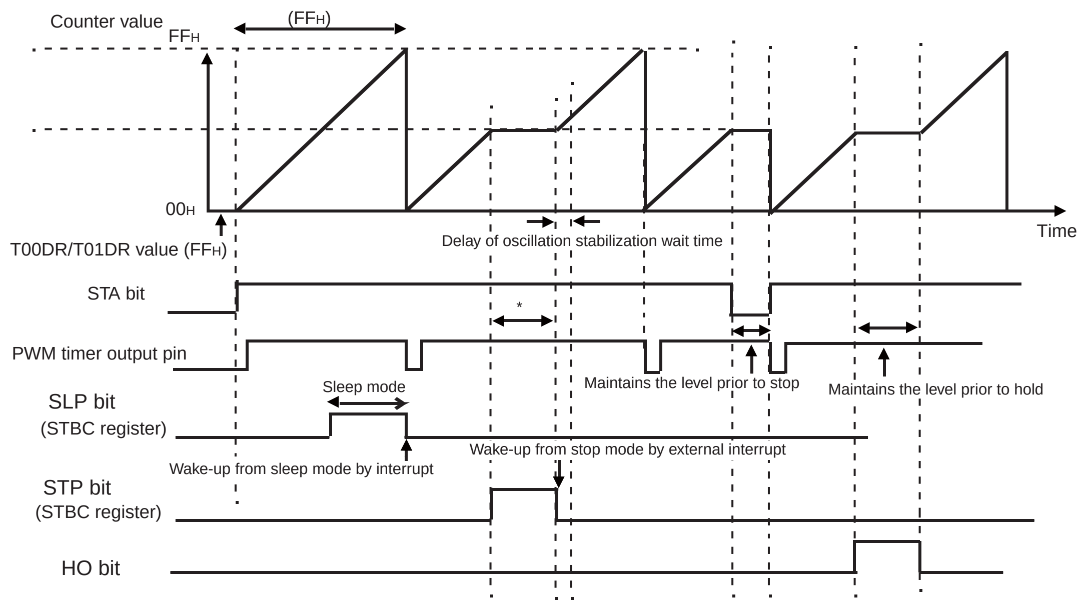


Figure 15.15-2 Operations of Counter in Standby Mode or in Pause (Serving as PWM Timer)



*: The PWM timer output maintains the value held before it enters the stop mode.

15.16 Notes on Using 8/16-bit Compound Timer

This section explains the precautions to be taken when using the 8/16-bit compound timer.

■ Notes on Using 8/16-bit Compound Timer

When changing the timer function by using the timer operation mode select bits (T00CR0/T01CR0:F3, F2, F1, F0), the timer operation must be stopped (T00CR1/T01CR1:STA = 0) before clearing the interrupt flag (T00CR1/T01CR1:IF, IR), interrupt enable bits (T00CR1/T01CR1:IE, T00CR0/T01CR0:IFE) and buffer full flag (T00CR1/T01CR1:BF).

When the PWC or input capture function has been selected, an interrupt may occur even before the timer is activated (STA = 0). Therefore, nullify the value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) obtained before the activation.

In the case of using the input capture function, when the timing at which the 8/16-bit compound timer captures a counter value is the detection of either edge of the external input signal (T00CR0/T01CR0:F3-F0=1100_B or 1111_B), the operations in falling edge detection vary according to the level of the external input signal as explained below.

- External input signal level: H

  In both free-run mode and clear mode, the first falling edge is ignored, no counter value is transferred to the data register (T00DR/T01DR), and the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) is not set. In addition, in clear mode, the counter is not cleared either.

- External input signal level: L

  The 8/16-bit compound timer starts edge detection from the first rising edge.

CHAPTER 16
8/16-BIT PPG

This chapter describes the functions and operations of the 8/16-bit PPG.

16.1 Overview of 8/16-bit PPG

The 8/16-bit PPG is an 8-bit reload timer module that uses pulse output control based on timer operation to perform PPG output. The 8/16-bit PPG also operates in cascade (8 bits + 8 bits) as 16-bit PPG.

■ Overview of 8/16-bit PPG

The following section summarizes the 8/16-bit PPG functions.

● 8-bit PPG output independent operation mode

In this mode, the unit can operate as 2 8-bit PPG (PPG timer 00 and PPG timer 01).

● 8-bit prescaler + 8-bit PPG output operation mode

The rising and falling edge detection pulses from the PPG timer 01 output can be inputted to the down-counter of the PPG timer 00 to enable variable-cycle 8-bit PPG output.

● 16-bit PPG output operation mode

The unit can also operate in cascade (PPG timer 01 (upper 8 bits) + PPG timer 00 (lower 8 bits)) as 16-bit PPG output.

● PPG output operation

In this operation, a variable-cycle pulse waveform is outputted in any duty ratio.

The unit can also be used as a D/A converter in conjunction with an external circuit.

● Output inversion mode

This mode can invert the PPG output value.

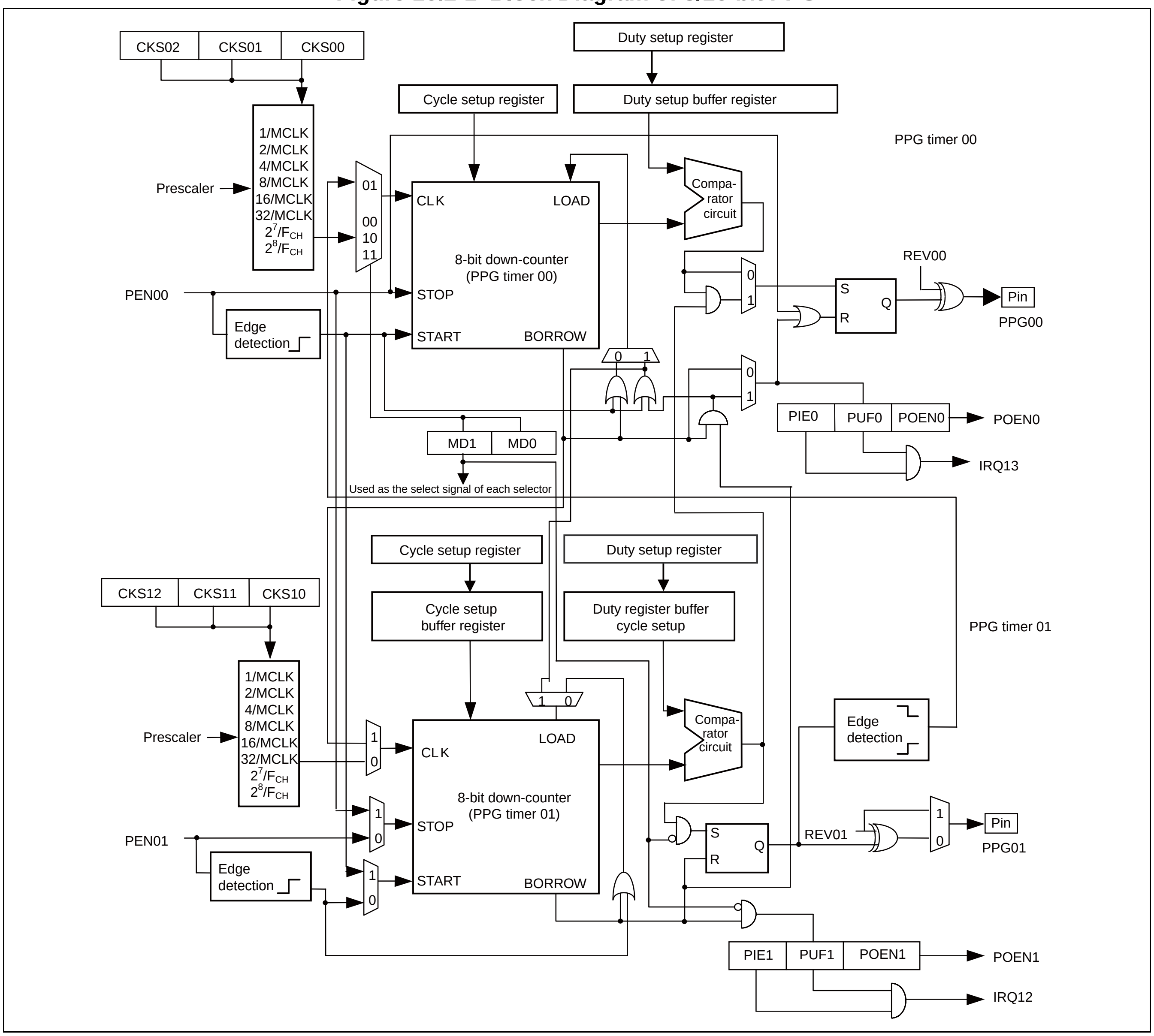
16.2 Configuration of 8/16-bit PPG

This section shows the block diagram of 8/16-bit PPG.

■ Block Diagram of 8/16-bit PPG

Figure 16.2-1 shows the block diagram of the 8/16-bit PPG.

Figure 16.2-1 Block Diagram of 8/16-bit PPG



● Counter clock selector

The clock for the countdown of 8-bit down counter is selected from eight types of internal count clocks.

● 8-bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" level until the value of 8-bit down counter is corresponding to the value of 8/16-bit PPG duty setup buffer register from the value of 8/16-bit set buffer register of PPG cycle.

Afterwards, after keep "L" level the output until the counter value is corresponding to "1", it keeps counting 8-bit down counter from the value of 8/16-bit PPG cycle setup buffer register.

● 8/16-bit PPG timer 01 control register (PC01)

The operation condition on the PPG timer 01 side of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 00 control register (PC00)

The operation mode of 8/16-bit PPG timer and the operation condition on the PPG timer 00 side are set.

● 8/16-bit PPG timer 01/00 cycle setup buffer register ch.0 (PPS01), ch.0 (PPS00)

The compare value for the cycle of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 01/00 duty setup buffer register ch.0 (PDS01), ch.0 (PDS00)

The compare value for "H" width of 8/16-bit PPG timer is set.

● 8/16-bit PPG start register

The start or the stop of 8/16-bit PPG timer is set.

● 8/16-bit PPG output inversion register

An initial level also includes the output of 8/16-bit PPG timer and it is reversed.

■ Input Clock

The 8/16-bit PPG uses the output clock from the prescaler as its input clock (count clock).

16.3 Channels of 8/16-bit PPG

This section describes the channels of the 8/16-bit PPG.

■ Channels of 8/16-bit PPG

MB95150/M series has two channels of the 8/16-bit PPG. There are 8-bit PPG timer 00 and 8-bit PPG timer 01 in 1 channel. They can be used respectively as two 8-bit PPGs. Also, they can be used as a 16-bit PPG.

Table 16.3-1 and Table 16.3-2 show the channels and their corresponding pins and registers.

Table 16.3-1 Pins of 8/16-bit PPG

Channel	Pin name	Pin function
0	PPG00	PPG timer 00 (8-bit PPG (00), 16-bit PPG)
	PPG01	PPG timer 01 (8-bit PPG (01), 8-bit prescaler)
1	PPG10	PPG timer 00 (8-bit PPG (10), 16-bit PPG)
	PPG11	PPG timer 01 (8-bit PPG (11), 8-bit prescaler)

Table 16.3-2 Registers of 8/16-bit PPG

Channel	Register name	Corresponding register (as written in this manual)
0	PC01	8/16-bit PPG timer 01 control register
	PC00	8/16-bit PPG timer 00 control register
	PPS01	8/16-bit PPG timer 01 cycle setup buffer register
	PPS00	8/16-bit PPG timer 00 cycle setup buffer register
	PDS01	8/16-bit PPG timer 01 duty setup buffer register
	PDS00	8/16-bit PPG timer 00 duty setup buffer register
1	PC11	8/16-bit PPG timer 01 control register
	PC10	8/16-bit PPG timer 00 control register
	PPS11	8/16-bit PPG timer 01 cycle setup buffer register
	PPS10	8/16-bit PPG timer 00 cycle setup buffer register
	PDS11	8/16-bit PPG timer 01 duty setup buffer register
	PDS10	8/16-bit PPG timer 00 duty setup buffer register
Both channels	PPGS	8/16-bit PPG start register
	REVC	8/16-bit PPG output inversion register

The following sections describe only the 8/16-bit PPG in ch.0 side.

16.4 Pins of 8/16-bit PPG

This section describes the pins of the 8/16-bit PPG.

■ Pins of 8/16-bit PPG

● PPG00 pin and PPG01 pin

These pins function both as general-purpose I/O ports and 8/16-bit PPG outputs.

PPG00, PPG01: A PPG waveform is outputted to these pins. The PPG waveform can be outputted by enabling the output by the 8/16-bit PPG timer 01/00 control registers (PC00: POEN0 = 1, PC01: POEN1 = 1).

■ Block Diagram of Pins Related to 8/16-bit PPG

Figure 16.4-1 Block Diagram of Pins (PPG00, PPG01) Related to 8/16-bit PPG

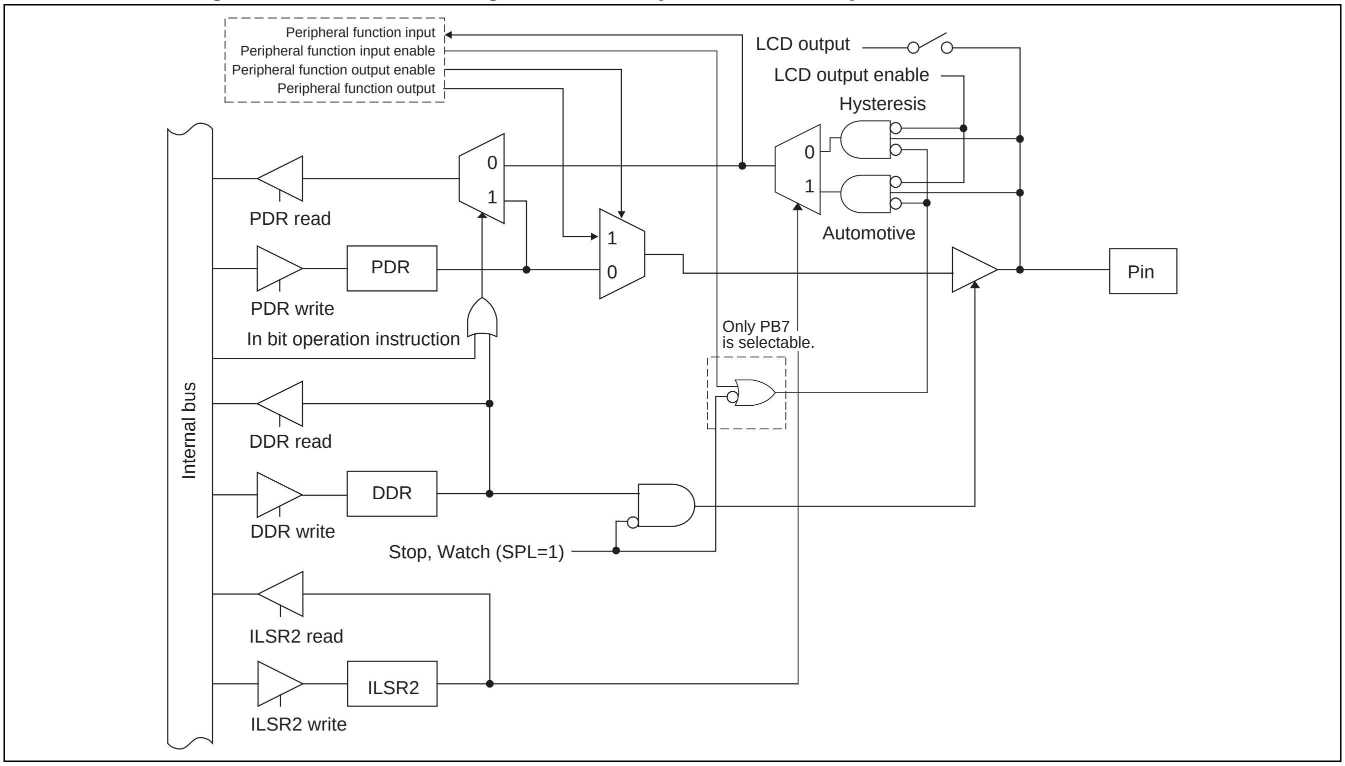
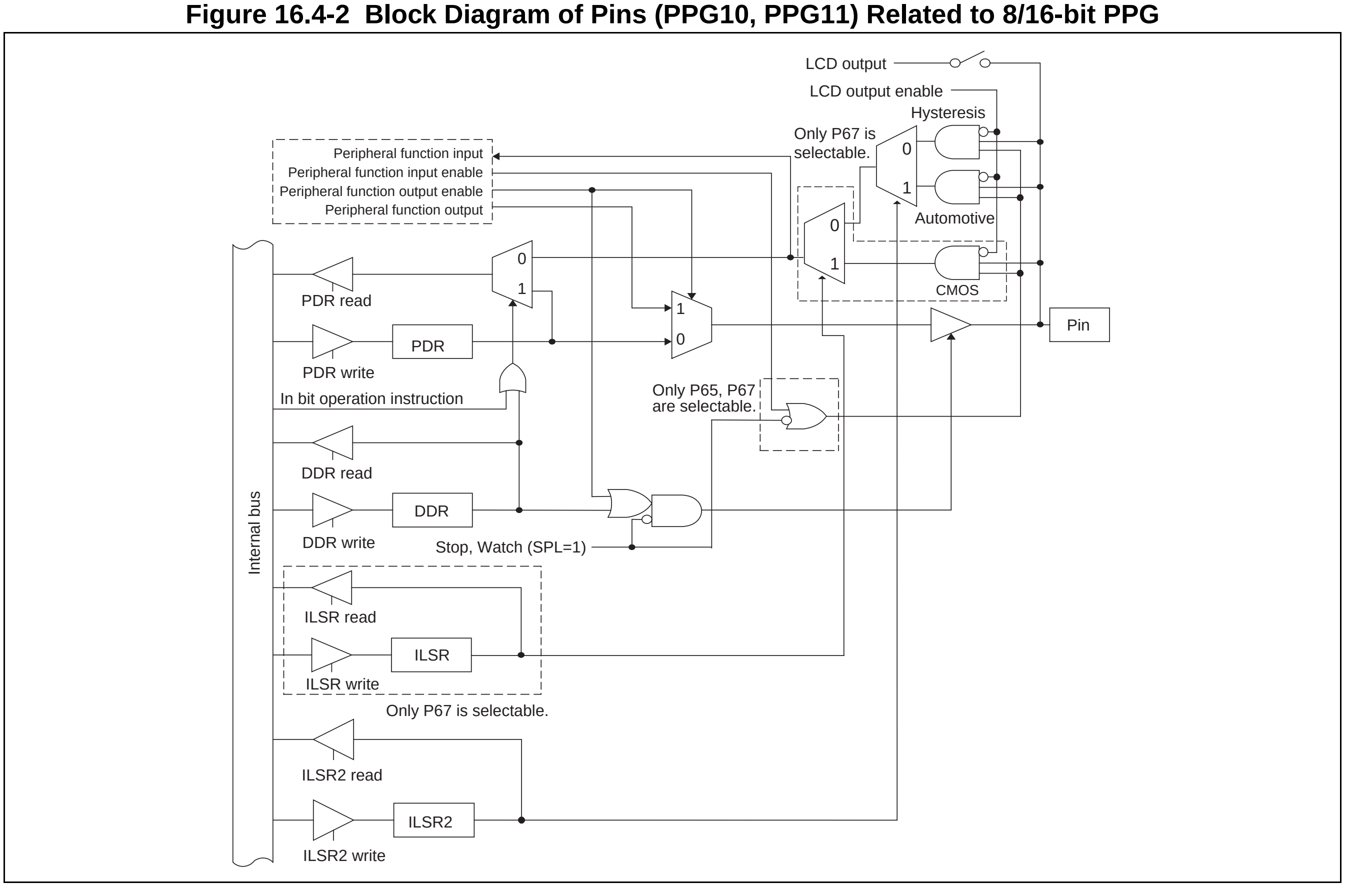


Figure 16.4-2 Block Diagram of Pins (PPG10, PPG11) Related to 8/16-bit PPG



16.5 Registers of 8/16-bit PPG

This section describes the registers of the 8/16-bit PPG.

■ Registers of 8/16-bit PPG

Figure 16.5-1 shows the registers of the 8/16-bit PPG.

Figure 16.5-1 Registers of 8/16-bit PPG

8/16-bit PPG timer 01 control register (PC01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC01 $003A_H$	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10	00000000_B
	R0/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 control register (PC00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC00 $003B_H$	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00	00000000_B
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 01 cycle setup buffer register (PPS01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PPS01 $0F9C_H$	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 cycle setup buffer register (PPS00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PPS00 $0F9D_H$	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 01 duty setup buffer register (PDS01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDS01 $0F9E_H$	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 duty setup buffer register (PDS00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDS00 $0F9F_H$	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG start register (PPGS)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FA4_H$	-	-	-	-	PEN11	PEN10	PEN01	PEN00	00000000_B
	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	

8/16-bit PPG output inversion register (REVC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FA5_H$	-	-	-	-	REV11	REV10	REV01	REV00	00000000_B
	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)
R(RM1), W: Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0/WX: Undefined bit (Read value is "0", writing has no effect on operation)

16.5.1 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)

The 8/16-bit PPG timer 01 control register ch.0 (PC01) sets the operating conditions for PPG timer 01.

■ 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)

Figure 16.5-2 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC01 003A$_H$ PC11 003C$_H$	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10	00000000$_B$
	R0/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

CKS12	CKS11	CKS10	Operating clock select bits
0	0	0	1/MCLK
0	0	1	2/MCLK
0	1	0	4/MCLK
0	1	1	8/MCLK
1	0	0	16/MCLK
1	0	1	32/MCLK
1	1	0	$2^7/F_{CH}$
1	1	1	$2^8/F_{CH}$

POEN1	Output enable bit
0	Output disabled (general-purpose port)
1	Output enabled

PUF1	Counter borrow detection flag bit for PPG cycle down-counter	
	Read	Write
0	Counter borrow undetected	Flag cleared
1	Counter borrow detected	No effect on operation

PIE1	Interrupt request enable bit
0	Interrupt disabled
1	Interrupt enabled

MCLK : Machine clock frequency
F_{CH} : Main oscillation frequency
R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
(shaded box) : Initial value

Table 16.5-1 8/16-bit PPG Timer 01 Control Register (PC01)

Bit name		Function
bit7, bit6	-: Undefined bits	These bits are undefined. • Writing to the bits is meaningless. • Read always returns "0".
bit5	PIE1: Interrupt request enable bit	This bit controls interrupts of PPG timer 01. **Setting the bit to "0"**: disables interrupts of PPG timer 01. **Setting the bit to "1"**: enables interrupts of PPG timer 01. The bit outputs an interrupt request (IRQ) when the counter borrow detection bit (PUF1) and the PIE1 bit are both set to "1".
bit4	PUF1: Counter borrow detection flag bit for PPG cycle down-counter	This bit serves as the counter borrow detection flag for the PPG cycle down-counter of the PPG timer 01. • This bit is set to "1" when a counter borrow occurs during 8-bit PPG mode or 8-bit prescaler mode. • In 16-bit PPG mode, this bit is not set to "1" even when a counter borrow occurs. • Writing "1" to the bit is meaningless. • Writing "0" clears the bit. • "1" is read in read-modify-write (RMW) instruction. **When the bit is set to "0"**: a counter borrow is undetected. **When the bit is set to "1"**: a counter borrow is detected.
bit3	POEN1: Output enable bit	This bit enables or disables the output of PPG timer 01 pin. **When the bit is set to "0"**: the PPG timer 01 pin is used as a general-purpose port. **When the bit is set to "1"**: the PPG timer 01 pin is used as the PPG output pin. Setting this bit to "1" during 16-bit PPG operation mode sets the PPG timer 01 pin as an output. (The setting value of REV01 is outputted. "L" output is supplied when REV01 is "0".)
bit2 to bit0	CKS12, CKS11, CKS10: Operating clock select bits	These bits select the operating clock for 8-bit down-counter of the PPG timer 01. • The operating clock is generated from the prescaler. Refer to "CHAPTER 6 CLOCK CONTROLLER". • In 16-bit PPG operation mode, the setting of this bit has no effect on the operation. **"000_B":** 1/MCLK **"001_B":** 2/MCLK **"010_B":** 4/MCLK **"011_B":** 8/MCLK **"100_B":** 16/MCLK **"101_B":** 32/MCLK **"110_B":** $2^7/F_{CH}$ **"111_B":** $2^8/F_{CH}$ Note: Use of a sub clock (in dual clock product) stops the time-base timer operation. Therefore, selecting "110_B" or "111_B" is prohibited.

16.5.2 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)

The 8/16-bit PPG timer 00 control register ch.0 (PC00) sets the operating conditions and the operation mode for PPG timer 00.

■ 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)

Figure 16.5-3 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC00 003BH PC10 003DH	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00	00000000B
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

CKS02	CKS01	CKS00	Operating clock select bits
0	0	0	1/MCLK
0	0	1	2/MCLK
0	1	0	4/MCLK
0	1	1	8/MCLK
1	0	0	16/MCLK
1	0	1	32/MCLK
1	1	0	$2^7/F_{CH}$
1	1	1	$2^8/F_{CH}$

POEN0	Output enable bit
0	Output disabled (general-purpose port)
1	Output enabled

PUF0	Counter borrow detection flag bit for PPG cycle down-counter	
	Read	Write
0	Counter borrow undetected	Flag cleared
1	Counter borrow detected	No effect on operation

PIE0	Interrupt request enable bit
0	Interrupt disabled
1	Interrupt enabled

MD1	MD0	Operation mode select bits
0	0	8-bit PPG independent mode
0	1	8-bit prescaler + 8-bit PPG mode
1	0	16-bit PPG mode
1	1	

MCLK : Machine clock frequency
F_{CH} : Main oscillation frequency
R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
[shaded box] : Initial value

Table 16.5-2 8/16-bit PPG0 Control Register (PC0)

Bit name		Function
bit7, bit6	MD1, MD0: Operation mode select bits	These bits select the PPG operation mode. Do not modify the bit settings during counting. **When set to "00_B":** 8-bit PPG independent mode **When set to "01_B":** 8-bit prescaler + 8-bit PPG mode **When set to "$1x_B$":** 16-bit PPG mode
bit5	PIE0: Interrupt request enable bit	This bit controls interrupts of PPG timer 00. • Set this bit in 16-bit PPG operation mode. **Setting the bit to "0":** disables interrupts of PPG timer 00. **Setting the bit to "1":** enables interrupts of PPG timer 00. • An interrupt request (IRQ) is outputted when the counter borrow detection bit (PUF0) and PIE0 bit are both set to "1".
bit4	PUF0: Counter borrow detection flag bit for PPG cycle down-counter	This is the counter borrow detection flag for the PPG cycle down-counter of PPG timer 00. • Only this bit is effective in 16-bit PPG operation mode (PC1:PUF1 is not operable). Note: Always effective in 8-bit mode • Writing "1" to this bit is meaningless. • Writing "0" clears the bit. • "1" is read in read-modify-write (RMW) instruction. **When set to "0":** Counter borrow of PPG timer 00 undetected **When set to "1":** Counter borrow of PPG timer 00 detected
bit3	POEN0: Output enable bit	This bit enables or disables the output of PPG timer 00 pin. **When set to "0":** PPG timer 00 pin is used as a general-purpose port. **When set to "1":** PPG timer 00 pin is used as the PPG output pin. As the output is supplied from the PPG timer 00 pin in 16-bit PPG operation mode, this bit is used to control the operation.
bit2 to bit0	CKS02, CKS01, CKS00: Operating clock select bits	These bits select the operating clock for PPG down-counter PPG timer 00. • The operating clock is generated from the prescaler. Refer to "CHAPTER 6 CLOCK CONTROLLER". • The rising and falling edge detection pulses from the PPG timer 01 output are used as the count clock for PPG timer 00 when the 8-bit prescaler + 8-bit PPG mode has been selected. Therefore, the setting of this bit has no effect on the operation. • Set this bit in 16-bit PPG operation mode. **"000_B":** 1/MCLK **"001_B":** 2/MCLK **"010_B":** 4/MCLK **"011_B":** 8/MCLK **"100_B":** 16/MCLK **"101_B":** 32/MCLK **"110_B":** $2^7/F_{CH}$ **"111_B":** $2^8/F_{CH}$ Note: Use of a sub clock (in dual clock product) stops the time-base timer operation. Therefore, selecting "110_B" or "111_B" is prohibited.

16.5.3 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

The 8/16-bit PPG timer 00/01 cycle setup buffer register (PPS01), (PPS00) sets the PPG output cycle.

■ 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

Figure 16.5-4 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

PPS01	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	11111111_B
$0F9C_H$ PPS01	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
$0FA0_H$ PPS11									

PPS00	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	11111111_B
$0F9D_H$ PPS00	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
$0FA1_H$ PPS10									

R/W : Readable/Writable (Read value is the same as write value)

This register is used to set the PPG output cycle.

- In 16-bit PPG mode, PPS01 serves as the upper 8 bits, while PPS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load.
- 8-bit mode: Cycle = max. 255 (FF_H) × Input clock cycle
- 16-bit mode: Cycle = max. 65535 ($FFFF_H$) × Input clock cycle
- Initialized at reset.
- Do not set the cycle to "00_H" or "01_H" when using the unit in 8-bit PPG independent mode, or in 8-bit prescaler mode + 8-bit PPG mode.
- Do not set the cycle to "0000_H" or "0001_H" when using the unit in 16-bit PPG mode.
- If the cycle settings are modified during the operation, the modified settings will be effective from the next PPG cycle.

16.5.4 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

The 8/16-bit PPG timer 00/01 duty setup buffer register (PDS01), (PDS00) sets the duty of the PPG output.

■ 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

Figure 16.5-5 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

PDS01	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	11111111_B
$0F9E_H$ PDS01	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
$0FA2_H$ PDS11									

PDS00	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0	11111111_B
$0F9F_H$ PDS00	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
$0FA3_H$ PDS10									

R/W : Readable/Writable (Read value is the same as write value)

This register is used to set the duty of the PPG output ("H" pulse width when normal polarity).

- In 16-bit PPG mode, PDS01 serves as the upper 8 bits while PDS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load. By writing to PDS00, PDS01 is updated.
- Initialized at reset.
- To set the duty to 0%, select "00_H".
- To set the duty to 100%, set it to the same value as the 8/16-bit PPG timer 00/01 cycle setup register (PPS).
- When the 8/16-bit PPG timer 00/01 duty setup register (PDS) is set to a larger value than the setting value of the 8/16-bit PPG timer 00/01 cycle setup buffer register (PPS), the PPG output becomes "L" output in the normal polarity (when the output level inversion bit of 8/16-bit PPG output inversion register is "0").
- If the duty settings are modified during operation, the modified value will be effective from the next PPG cycle.

16.5.5 8/16-bit PPG Start Register (PPGS)

The 8/16-bit PPG start register (PPGS) starts or stops the down-counter. The operation enable bit of each channel is assigned to the PPGS register, allowing simultaneous activation of the PPG channels.

■ 8/16-bit PPG Start Register (PPGS)

Figure 16.5-6 8/16-bit PPG Start Register (PPGS)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA4H	- *	- *	- *	- *	PEN11	PEN10	PEN01	PEN00	00000000B
	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	

PEN00	PPG timer 00 (ch.0) down-counter operation enable bit
0	Stops operation
1	Enables operation

PEN01	PPG timer 01 (ch.0) down-counter operation enable bit
0	Stops operation
1	Enables operation

PEN10	PPG timer 00 (ch.1) down-counter operation enable bit
0	Stops operation
1	Enables operation

PEN11	PPG timer 01 (ch.1) down-counter operation enable bit
0	Stops operation
1	Enables operation

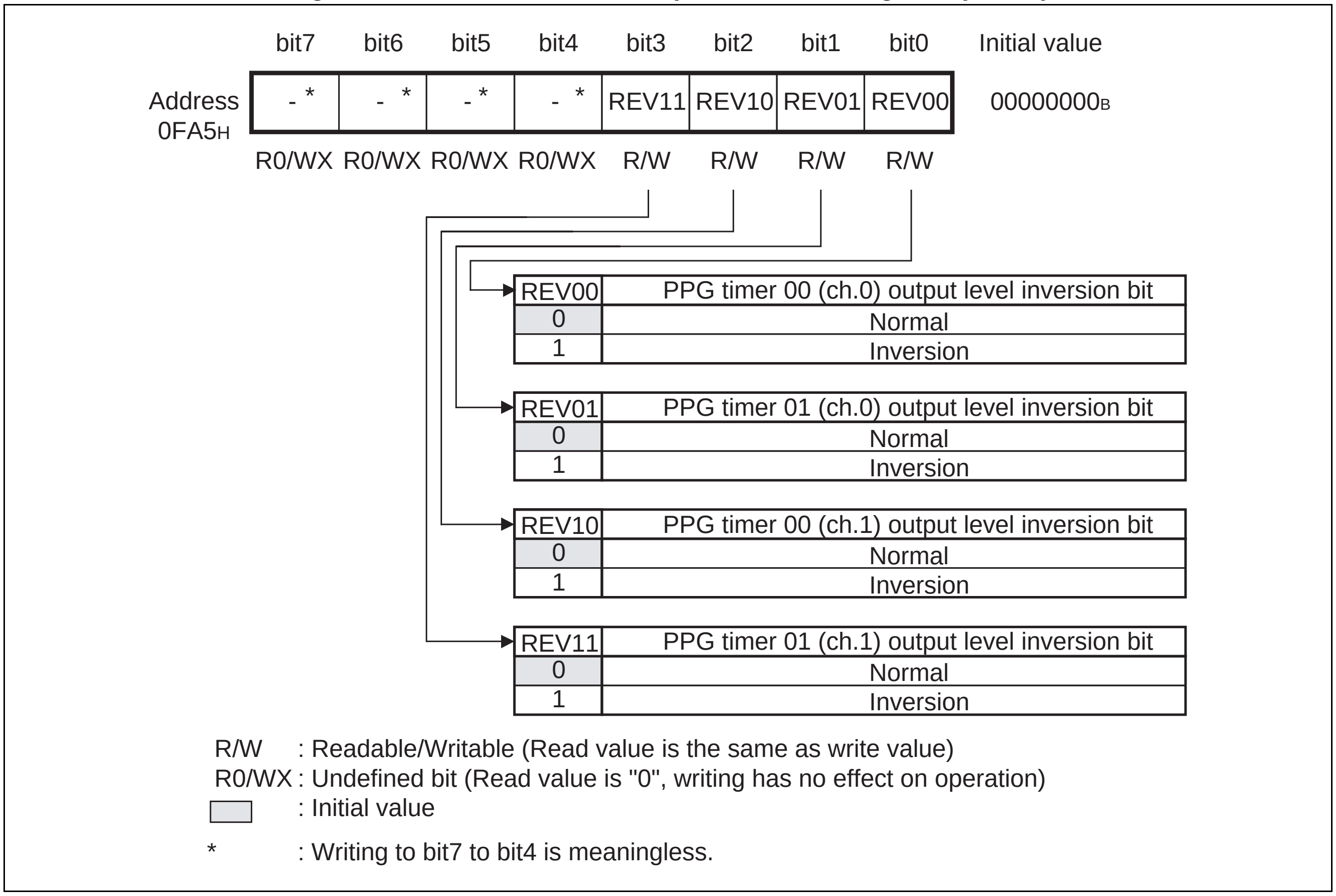
R/W : Readable/Writable (Read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
(shaded) : Initial value
* : Writing to bit7 to bit4 is meaningless.

16.5.6 8/16-bit PPG Output Inversion Register (REVC)

The 8/16-bit PPG output inversion register (REVC) inverts the PPG output including the initial level.

■ 8/16-bit PPG Output Inversion Register (REVC)

Figure 16.5-7 8/16-bit PPG Output Inversion Register (REVC)



16.6 Interrupts of 8/16-bit PPG

The 8/16-bit PPG outputs an interrupt request when a counter borrow is detected.

■ Interrupts of 8/16-bit PPG

Table 16.6-1 shows the interrupt control bits and interrupt sources of the 8/16-bit PPG.

Table 16.6-1 Interrupt Control Bits and Interrupt Sources of 8/16-bit PPG

Item	Description	
	PPG timer 01 (8-bit PPG, 8-bit prescaler)	PPG timer 00 (8-bit PPG, 16-bit PPG)
Interrupt request flag bit	PUF1 bit in PC01	PUF0 bit in PC00
Interrupt request enable bit	PIE1 bit in PC01	PIE0 bit in PC00
Interrupt source	Counter borrow of PPG cycle down-counter	

When a counter borrow occurs on the down-counter, the 8/16-bit PPG sets the counter borrow detection flag bit (PUF) in the 8/16-bit PPG timer 00/01 control register (PC) to "1". When the interrupt request enable bit is enabled (PIE = 1), an interrupt request is outputted to the interrupt controller.

In 16-bit PPG mode, the 8/16-bit PPG timer 00 control register (PC00) is available.

■ Registers and Vector Table Related to Interrupts of 8/16-bit PPG

Table 16.6-2 Registers and Vector Table Related to Interrupts of 8/16-bit PPG

Interrupt source	Interrupt request No.	Interrupt level setup register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.1 (lower)	IRQ9	ILR2	L09	$FFE8_H$	$FFE9_H$
ch.1 (upper)	IRQ10	ILR2	L10	$FFE6_H$	$FFE7_H$
ch.0 (lower)	IRQ12	ILR3	L12	$FFE2_H$	$FFE3_H$
ch.0 (upper)	IRQ13	ILR3	L13	$FFE0_H$	$FFE1_H$

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

16.7 Operating Description of 8/16-bit PPG

This section describes the operations of the 8/16-bit PPG.

■ Setup Procedure Example

The setup procedure of the 8/16-bit PPG is described below.

● Initial setting

1) Set the port output (DDRB, DDR6)
2) Set the interrupt level (ILR2, ILR3)
3) Select the operating clock, enable the output and interrupt (PC01)
4) Select the operating clock, enable the output and interrupt, select the operation mode (PC00)
5) Set the cycle (PPS)
6) Set the duty (PDS)
7) Set the output inversion (REVC)
8) Start PPG (PPGS)

● Interrupt processing

1) Process any interrupt
2) Clear the interrupt request flag (PC01: PUF1, PC00: PUF0)
3) Start PPG (PPGS)

16.7.1 8-bit PPG Independent Mode

In this mode, the unit operates as two channels (PPG timer 00 and PPG timer 01) of the 8-bit PPG.

■ Setting 8-bit Independent Mode

The unit requires the register settings shown in Figure 16.7-1 to operate in 8-bit independent mode.

Figure 16.7-1 8-bit Independent Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0	◎	◎	◎	◎	◎	◎
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	◎	◎
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	◎	◎

◎ : Used bit
0 : Set "0"
* : The bit status depends on the number of channels provided.

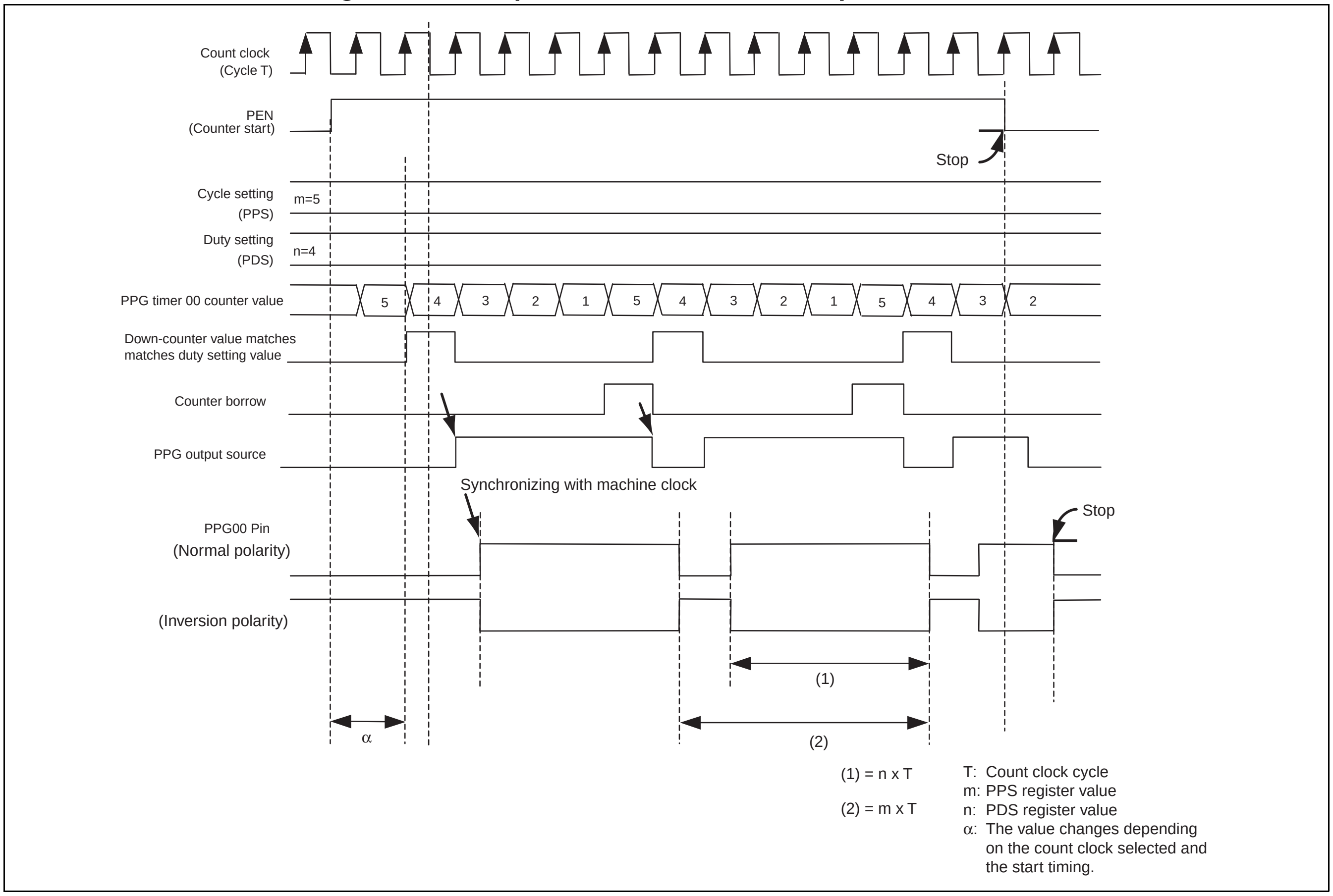
■ Operation of 8-bit PPG Independent Mode

- This mode is selected when the operation mode select bits (MD1, MD0) in the 8/16-bit PPG timer 00 control register (PC00) are set to "00_B".
- When the corresponding bit (PEN) in the 8/16-bit PPG start register (PPGS) is set to "1", the value in the 8/16-bit PPG cycle setup buffer register (PPS) is loaded to start down-count operation. When the count value reaches "1", the value in the cycle setup register is reloaded to repeat the counting.
- "H" is output to the PPG output synchronizing with the count clock. When the down-counter value matches the value in the 8/16-bit PPG timer 00/01 duty setup buffer register (PDS). After "H" which is the value of duty setting is output, "L" is output to the PPG output.

If, however, the PPG output inversion bit is set to "1", the PPG output is set and reset inversely from the above process.

Figure 16.7-2 shows the operation of the 8-bit PPG independent mode.

Figure 16.7-2 Operation of 8-bit PPG Independent Mode



Example for setting the duty to 50%

When PDS is set to "02_H" with PPS set to "04_H", the PPG output is set at a duty ratio of 50% (PPS setting value /2 set to PDS).

16.7.2 8-bit Prescaler + 8-bit PPG Mode

In this mode, the rising and falling edge detection pulses from the PPG timer 01 output can be used as the count clock of the PPG timer 00 down-counter to allow variable-cycle 8-bit PPG output from PPG timer 00.

■ Setting 8-bit Prescaler + 8-bit PPG Mode

The unit requires the register settings shown in Figure 16.7-3 to operate in 8-bit prescaler + 8-bit PPG mode.

Figure 16.7-3 Setting 8-bit Prescaler + 8-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	1	◎	◎	◎	×	×	×
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	◎	◎
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	◎	◎

◎ : Used bit
0 : Set "0"
1 : Set "1"
× : Setting nullified
* : The bit status varies depending of the number of channels implemented

■ Operation of 8-bit Prescaler + 8-bit PPG Mode

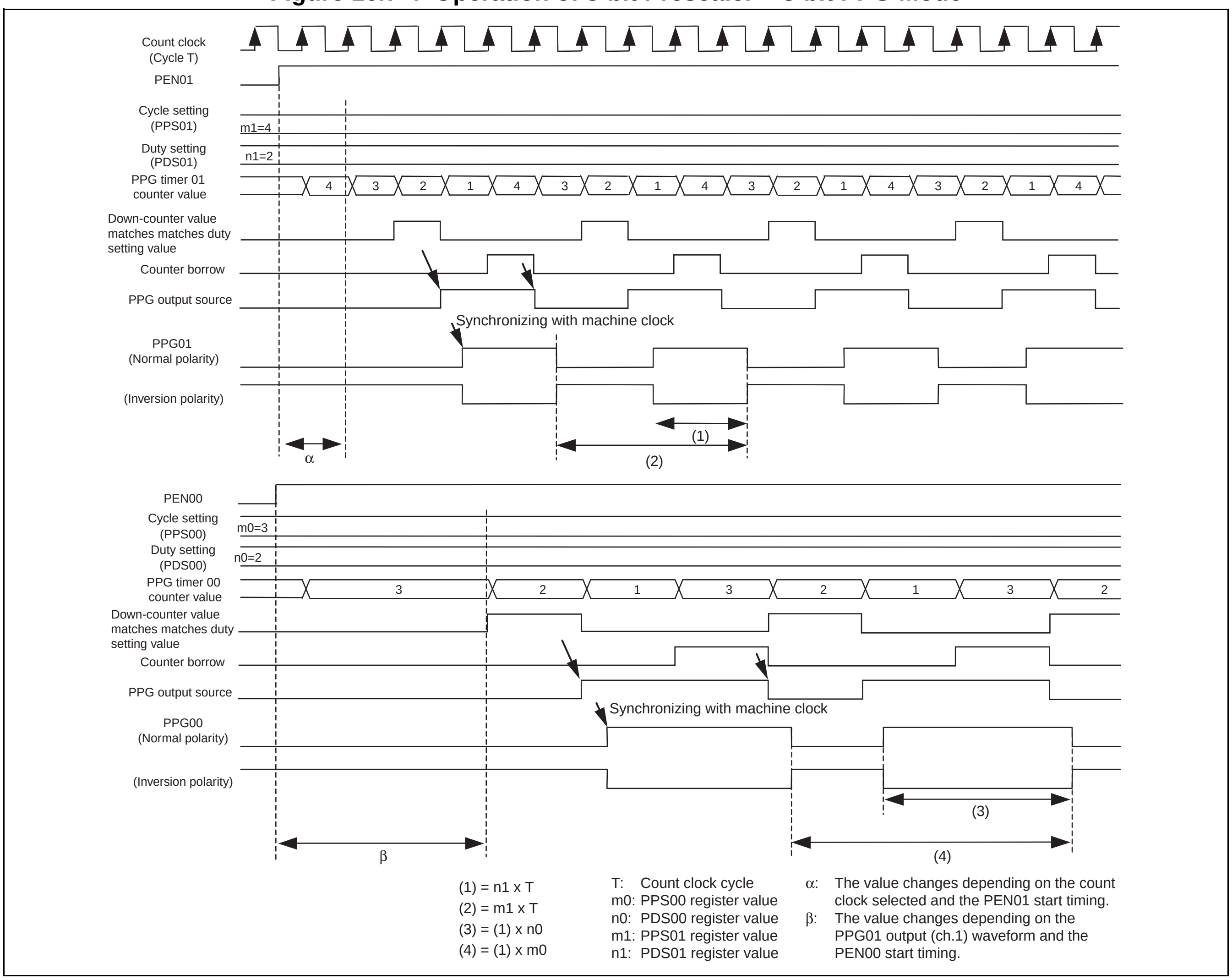
- This mode is selected by setting the operation mode select bits (MD1, MD0) of the 8/16-bit PPG timer 00 control register (PC00) to "01_B". This allows PPG timer 01 to be used as an 8-bit prescaler and PPG timer 00 to be used as an 8-bit PPG.
- When the PPG timer 01 (ch.0) down counter operation enable bit (PEN01) is set to "1", the 8-bit prescaler (PPG timer 01) loads the value in the 8/16-bit PPG timer 01 cycle setup buffer register (PPS01) and starts down-count operation. When the value of the down-counter matches the value in the 8/16-bit PPG timer 01 duty setup buffer register (PDS01), the PPG01 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG01 output is set to "L". If the output inversion signal (REV01) is "0", the polarity will remain the same. If it is "1", the

polarity will be inverted and the signal will be outputted to the PPG pin.

- When the PPG operation enable bit (PEN00) is set to "1", the 8-bit PPG (PPG timer 00) loads the value in the 8/16-bit PPG timer 00 cycle setup buffer register (PPS00) and starts down-count operation (count clock = rising and falling edge detection pulses of PPG01 output after PPG timer 01 operation is enabled). When the count value reaches "1", the value in the 8/16-bit PPG timer 00 cycle setup buffer register is reloaded to repeat the counting. When the value of the down-counter matches the value in the 8/16-bit PPG timer 00 duty setup buffer register (PDS00), the PPG00 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 output is reset to "L". If the output inversion signal (REV00) is "0", the polarity will remain the same. If it is "1", the polarity will be inverted and the signal will be outputted to the PPG00 pin.
- Set that the duty of the 8-bit prescaler (PPG timer 01) output to 50%.
- When PPG timer 00 is started with the 8-bit prescaler (PPG timer 01) being stopped, PPG timer 00 does not count.
- When the duty of the 8-bit prescaler (PPG timer 01) is set to 0% or 100%, PPG timer 00 does not perform counting as the 8-bit prescaler (PPG timer 01) output does not toggle.

Figure 16.7-4 shows the operation of 8-bit prescaler + 8-bit PPG mode.

Figure 16.7-4 Operation of 8-bit Prescaler + 8-bit PPG Mode



16.7.3 16-bit PPG Mode

In this mode, the unit can operate as a 16-bit PPG when PPG timer 01 and PPG timer 00 are assigned to the upper and lower bits respectively.

■ Setting 16-bit PPG Mode

The unit requires the register settings shown in Figure 16.7-5 to operate in 16-bit PPG mode.

Figure 16.7-5 Setting 16-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0/1	◎	◎	◎	◎	◎	◎
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle (Upper 8 bits) for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle (Lower 8 bits) for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty (Upper 8 bits) for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty (Lower 8 bits) for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	×	◎
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	×	◎

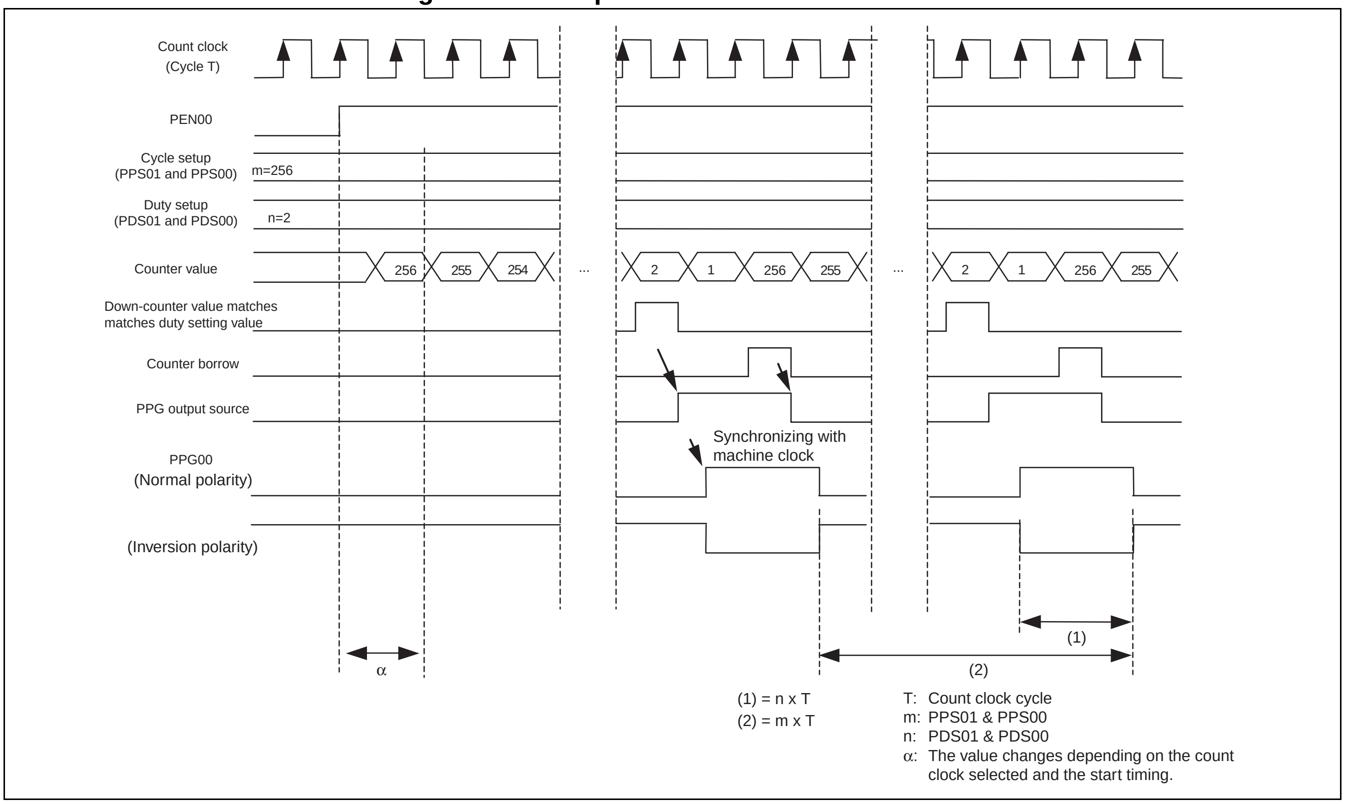
◎ : Used bit
0 : Set "0"
1 : Set "1"
× : Setting nullified
* : The bit status changes depending on the number of channels implemented.

■ Operation of 16-bit PPG Mode

- This mode is selected by setting the operation mode select bits (MD1, MD0) of the PPG timer 00 control register (PC00) to "10_B" or "11_B".
- When the PPG operation enable bit (PEN00) is set to "1" in 16-bit PPG mode, the 8-bit down-counters (PPG timer 00) and 8-bit down-counter (PPG timer 01) load the values in the 8/16-bit PPG timer 00/01 cycle setup buffer registers (PPS01 for PPG timer 01 and PPS00 for PPG timer 00) and start down-count operation. When the count value reaches "1", the values in the cycle setup register are reloaded and the counters repeat the counting.
- When the values of the down-counters match the values in the 8/16-bit PPG timer duty setup buffer registers (both the value in PDS01 for PPG timer 01 and the value in PDS00 for PPG timer 00), the PPG00 pin is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 pin is set to "L". If the output inversion signal (REV00) is "0", the signal will be outputted to the PPG00 with the polarity unchanged. If it is set to "1", the polarity will be inverted and the signal will be outputted to the PPG00 pin. (ch.0 only. ch.1 will be set to the initial value <"L" if REV01 is "0", or "H" if it is "1">.)

Figure 16.7-6 shows the operation of 16-bit PPG mode.

Figure 16.7-6 Operation of 16-bit PPG Mode



16.8 Notes on Using 8/16-bit PPG

The following precautions must be followed when using the 8/16-bit PPG.

■ Notes on Using 8/16-bit PPG

● Operational precaution

Depending on the timing between the activation of PPG and count clock, an error may occur in the first cycle of the PPG output immediately after the activation. The error varies depending on the count clock selected. The output, however, is performed properly in the succeeding cycles.

● Precaution regarding interrupts

A PPG interrupt is generated when the interrupt enable bit (PIE1/PIE0) is set to "1" and the interrupt request flag bit (PUF1/PUF0) in the 8/16-bit PPG timer 01/00 control register (PC01/PC00) is also set to "1". Always clear the interrupt request flag bit (PUF1/PUF0) to "0" in the interrupt routine.

16.9 Sample Programs for 8/16-bit PPG Timer

We provide sample programs that can be used to operate the 8/16-bit PPG timer.

■ Sample Programs for 8/16-bit PPG Timer

For information about the sample programs for the 8/16-bit PPG timer, refer to "■ Sample Programs" in PREFACE.

■ Setup Methods without Sample Program

● How to enable/stop PPG operation

The PPG operation enable bit (PPGS:PEN00 or PPGS:PEN10) is used for PPG timer 00.

Control	PPG operation enable bit (PEN00 or PEN10)
When stopping PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

PPG operation must be enabled before the PPG is activated.

The PPG operation enable bit (PPGS:PEN01 or PPGS:PEN11) is used for PPG timer 01.

Control	PPG operation enable bit (PEN01 or PEN11)
When stopping PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

PPG operation must be enabled before the PPG is activated.

● How to set the PPG operation mode

The operation mode select bits (PC00:MD[1:0]) are used.

● How to select the operating clock

ch.1 is selected by the operating clock select bits (PC01:CKS12/CKS11/CKS10).

ch.0 is selected by the operating clock select bits (PC00:CKS02/CKS01/CKS00).

● How to enable/disable the PPG output pin

The output enable bit (PC00:POEN0 or PC01:POEN1) is used.

Control	Output enable bit (POEN0 or POEN1)
When enabling PPG output	Set the bit to "1"
When disabling PPG output	Set the bit to "0"

- How to invert the PPG output

The output level inversion bit (REVC:REV00 or REVC:REV10) is used for PPG timer 00.

Control	Output level inversion bit (REV00 or REV10)
When inverting PPG output	Set the bit to "1"

The output level inversion bit (REVC:REV01 or REVC:REV11) is used for PPG timer 01.

Control	Output level inversion bit (REV01 or REV11)
When inverting PPG output	Set the bit to "1"

- Interrupt-related register

The interrupt level is set by the interrupt setup register shown in the following table.

Interrupt source	Interrupt level setup register	Interrupt vector
ch.1 (lower)	Interrupt level register (ILR2) Address:0007B$_H$	#09 Address:0FFE8$_H$
ch.1 (upper)	Interrupt level register (ILR2) Address:0007B$_H$	#10 Address:0FFE6$_H$
ch.0 (lower)	Interrupt level register (ILR3) Address:0007C$_H$	#12 Address:0FFE2$_H$
ch.0 (upper)	Interrupt level register (ILR3) Address:0007C$_H$	#13 Address:0FFE0$_H$

- How to enable/disable/clear interrupts

Interrupt request enable flag, Interrupt request flag

The interrupt request enable bit (PC00:PIE0 or PC01:PIE1) is used to enable or disable interrupts.

Operation	Interrupt request enable bit (PIE0 or PIE1)
When disabling interrupt requests	Set the bit to "0"
When enabling interrupt requests	Set the bit to "1"

The interrupt request flag (PC00:PUF0 or PC01:PUF1) is used to clear interrupt requests.

Operation	Interrupt request flag (PUF0 or PUF1)
When clearing interrupt requests	Write "0"

CHAPTER 17

16-BIT PPG TIMER

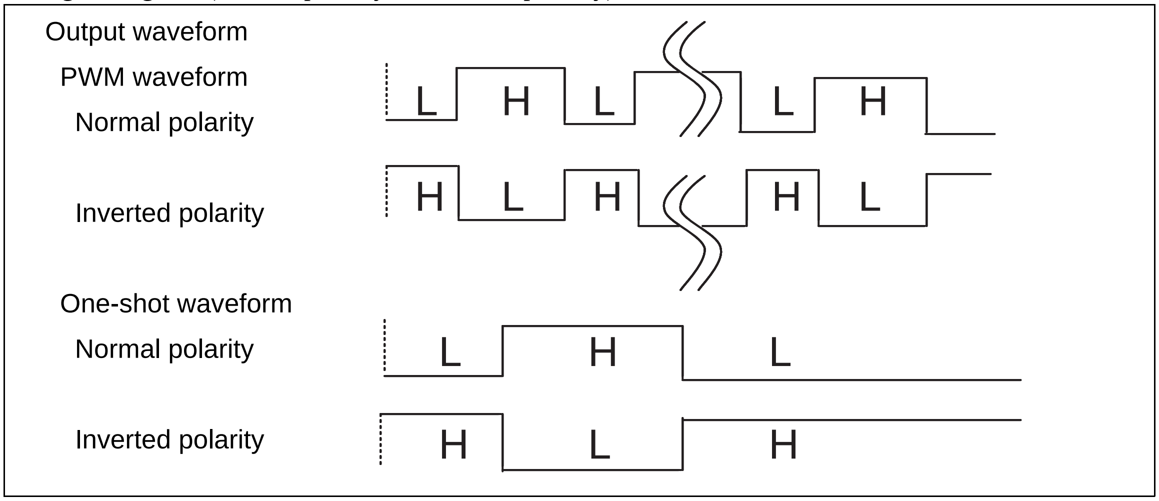
This chapter describes the functions and operations of the 16-bit PPG timer.

17.1 Overview of 16-bit PPG Timer

The 16-bit PPG timer can generate a PWM (Pulse Width Modulation) output or one-shot (square wave) output, and the period and duty of the output waveform can be changed by software freely. The timer can also generate an interrupt when a start trigger occurs or on the rising or falling edge of the output waveform.

■ 16-bit PPG Timer

16-bit PPG timer can output the PWM output and the one shot. The output wave form can be reversed by setting the register (Normal polarity ↔ Inverted polarity).



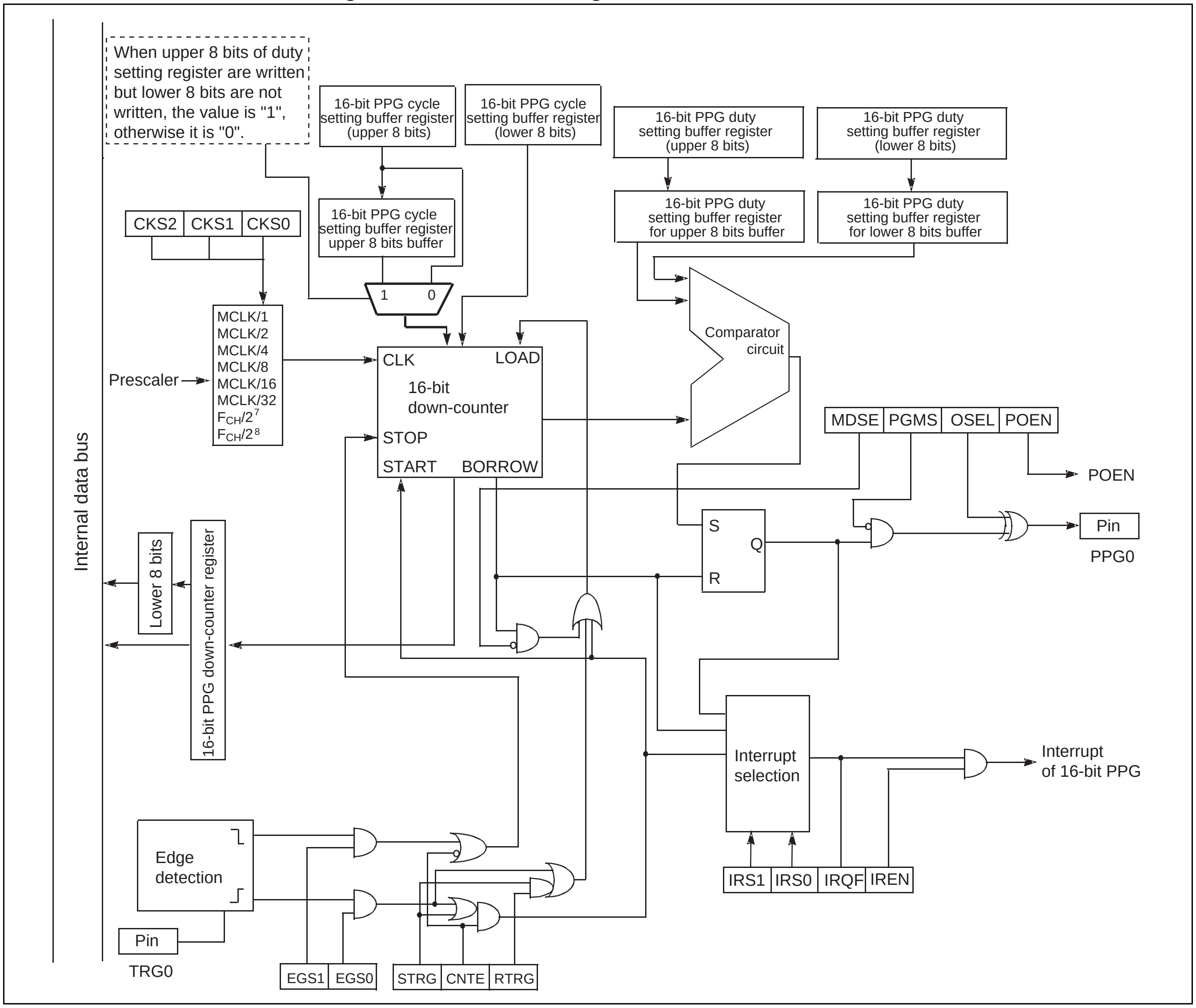
- The count operation clock can be selected from eight different clock sources (MCLK/1, MCLK/2, MCLK/4, MCLK/8, MCLK/16, MCLK/32, $F_{CH}/2^7$, or $F_{CH}/2^8$). (MCLK: Machine clock, F_{CH}: Main Clock)
- Interrupt can be selectively triggered by the following four conditions:
  - Occurrence of a start trigger in the PPG timer
  - Occurrence of a counter borrow in the 16-bit down-counter (cycle match).
  - Rising edge of PPG in normal polarity or falling edge of PPG in inverted polarity
  - Counter borrow, rising edge of PPG in normal polarity, or falling edge of PPG in inverted polarity

17.2 Configuration of 16-bit PPG Timer

Shown below is the block diagram of the 16-bit PPG timer.

■ Block Diagram of 16-bit PPG Timer

Figure 17.2-1 Block Diagram of 16-bit PPG Timer



● Count clock selector

The clock for the countdown of 16-bit down-counter is selected from eight types of internal count clocks.

● 16 bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" until the value of 16-bit down-counter is corresponding to the value of 8/16-bit PPG duty setting buffer register from the value of 16-bit PPG cycle setting buffer register.

Afterwards, after keep "L" the output until the counter value is corresponding to "1", it keeps counting 8-bit down counter from the value of 16-bit PPG cycle setting buffer register.

● 16-bit PPG down-counter register (PDCRH0, PDCRL0)

The value of 16-bit down-counter of 16-bit PPG timer is read.

● 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0)

The compare value for the cycle of 16-bit PPG timer is set.

● 16-bit PPG duty setting buffer register (PDUTH0, PDUTL0)

The compare value for "H" width of 16-bit PPG timer is set.

● 16-bit PPG status control register (PCNTH0, PCNTL0)

The operation mode and the operation condition of 16-bit PPG timer are set.

■ Input Clock

The 16-bit PPG timer uses the output clock from the prescaler as its input clock (count clock).

MB95150/M Series

17.3 Channels of 16-bit PPG Timer

This section describes the channels of the 16-bit PPG timer.

■ Channels of 16-bit PPG Timer

MB95150/M series has one16-bit PPG timer.

Table 17.3-1 and Table 17.3-2 show the correspondence among the channel, pin and register.

Table 17.3-1 Pins of 16-bit PPG Timer

Channel	Pin name	Pin function
0	PPG0	PPG0 output
	TRG0	Trigger 0 input

Table 17.3-2 Registers of 16-bit PPG Timer

Channel	Register name	Corresponding register (name in this manual)
0	PDCRH0	16-bit PPG down counter register (upper)
	PDCRL0	16-bit PPG down counter register (lower)
	PCSRH0	16-bit PPG cycle setting buffer register (upper)
	PCSRL0	16-bit PPG cycle setting buffer register (lower)
	PDUTH0	16-bit PPG duty setting buffer register (upper)
	PDUTL0	16-bit PPG duty setting buffer register (lower)
	PCNTH0	16-bit PPG status control register (upper)
	PCNTL0	16-bit PPG status control register (lower)

17.4 Pins of 16-bit PPG Timer

This section describes the pins of the 16-bit PPG timer.

■ Pins of 16-bit PPG Timer

The pin related to the 16-bit PPG timer is namely the PPG0 pin and TRG0 pin.

● PPG0 pin

Each pin serves as a general-purpose I/O port as well as a 16-bit PPG timer output.

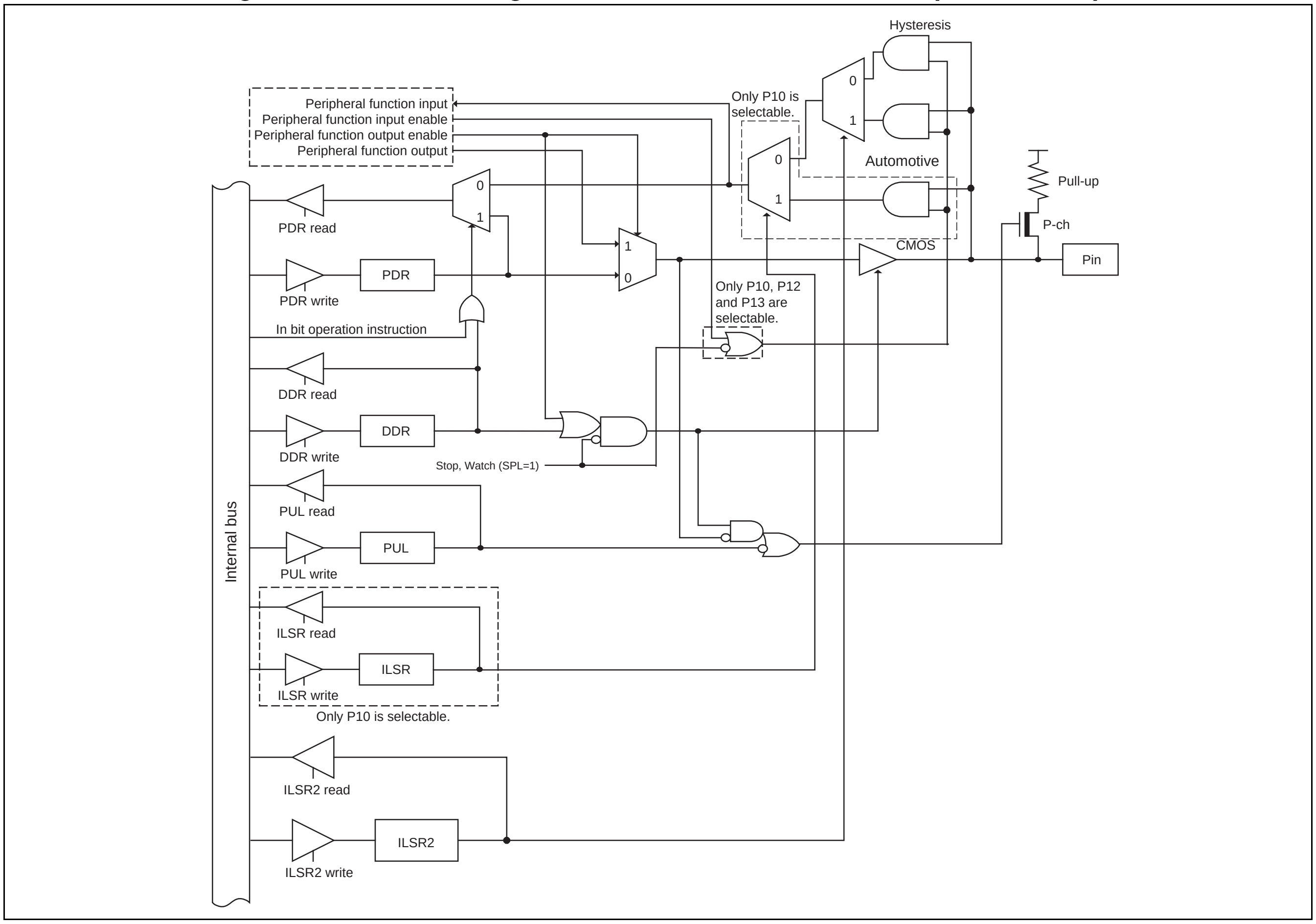
PPG0: A PPG waveform is outputted to these pins. The PPG waveform can be outputted by using the 16-bit PPG status control register to enable output (PCNTL0: POEN=1).

● TRG0 pin

TRG0: Used to start 16-bit PPG timer by hardware trigger.

■ Block Diagrams of Pins Related to 16-bit PPG

Figure 17.4-1 Block Diagram of Pin Related to 16-bit PPG (PPG0, TRG0)



17.5 Registers of 16-bit PPG Timer

This section describes the registers of the 16-bit PPG timer.

■ Registers of 16-bit PPG Timer

Figure 17.5-1 Registers of 16-bit PPG Timer

16-bit PPG down counter register (upper): PDCRH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PDCRH0 $0FAA_H$	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down counter register (lower): PDCRL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDCRL0 $0FAB_H$	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG cycle setting buffer register (upper): PCSRH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PCSRH0 $0FAC_H$	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG cycle setting buffer register (lower): PCSRL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCSRL0 $0FAD_H$	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (upper): PDUTH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PDUTH0 $0FAE_H$	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (lower): PDUTL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDUTL0 $0FAF_H$	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG status control register (upper): PCNTH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PCNTH0 0042_H	CNTE	STRG	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	00000000_B
	R/W	R0, W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG status control register (lower): PCNTL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCNTL0 0043_H	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	00000000_B
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)
R/WX: Read only (Readable, writing has no effect on operation)
R(RM1), W: Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R0, W: Write only (Writable, the read value is "0")

17.5.1 16- bit PPG Down Counter Registers (PDCRH0, PDCRL0)

The 16-bit PPG down counter registers (PDCRH0, PDCRL0) form a 16-bit register which is used to read the count value from the 16-bit PPG down-counter.

■ 16-bit PPG Down Counter Registers (PDCRH0, PDCRL0)

Figure 17.5-2 16-bit PPG Down Counter Registers (PDCRH0, PDCRL0)

16-bit PPG down counter register (upper) PDCRH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PDCRH0 $0FAA_H$	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down counter register (lower) PDCRL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDCRL0 $0FAB_H$	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX: Read only (Readable, writing has no effect on operation)

These registers form a 16-bit register which is used to read the count value from the 16-bit down-counter. The initial values of the register are all "0".

Always use one of the following procedures to read from this register.

- Use the "MOVW" instruction (use a 16-bit access instruction to read the PDCRH0 register address)
- Use the "MOV" instruction and read PDCRH0 first and PDCRL0 second (reading PDCRH0 automatically copies the lower 8 bits of the down-counter to PDCRL0)

These registers are read only and writing has no effect on the operation.

Note:

If you use the "MOV" instruction and read PDCRL0 before PDCRH0, PDCRL0 will return the value from the previous valid read operation. Therefore, the value of the 16-bit down-counter will not be read correctly.

17.5.2 16-bit PPG Cycle Setting Buffer Registers (PCSRH0, PCSRL0)

The 16-bit PPG cycle setting buffer registers are used to set the cycle for the output pulses generated by the PPG.

■ 16-bit PPG Cycle Setting Buffer Registers (PCSRH0, PCSRL0)

Figure 17.5-3 16-bit PPG Cycle Setting Buffer Registers (PCSRH0, PCSRL0)

16-bit PPG cycle setting buffer register (upper) PCSRH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PCSRH0 $0FAC_H$	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG cycle setting buffer register (lower) PCSRL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCSRL0 $0FAD_H$	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)

These registers form a 16-bit register which sets the period for the output pulses generated by the PPG. The values set in these registers are loaded to the down-counter.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PCSRH0 register address)
- Use the "MOV" instruction and write to PCSRH0 first and PCSRL0 second
  If a down-counter load occurs after writing data to PCSRH0 (but before writing data to PCSRL0), the previous valid PCSRH0/PCSRL0 value will be loaded to the down-counter. If the PCSRH0/PCSRL0 value is modified during counting, the modified value will become effective from the next load of the down-counter.
- Do not set PCSRH0 and PCSRL0 to "00_H", or PCSRH0 and PCSRL0 to "01_H".

Note:

If the down-counter load occurs after the "MOV" instruction is used to write data to PCSRL0 before PCSRH0, the previous valid PCSRH0 value and newly written PCSRL0 value are loaded to the down-counter. It should be noted that as a result, the correct period cannot be set.

17.5.3 16-bit PPG Duty Setting Buffer Registers (PDUTH0, PDUTL0)

The 16-bit PPG duty setting buffer registers control the duty ratio for the output pulses generated by the PPG.

■ 16-bit PPG Duty Setting Buffer Registers (PDUTH0, PDUTL0)

Figure 17.5-4 16-bit PPG Duty Setting Buffer Registers (PDUTH0, PDUTL0)

16-bit PPG duty setting buffer register (upper) PDUTH0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
PDUTH0 $0FAE_H$	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (lower) PDUTL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDUTL0 $0FAF_H$	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)

These registers form a 16-bit register which controls the duty ratio for the output pulses generated by the PPG. Transfer of the data from the 16-bit PPG duty setting buffer registers to the duty setting registers is performed at the same timing as the down-counter read.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PDUTH0 register address)
- Use the "MOV" instruction and write to PDUTH0 first and PDUTL0 second
  If a down-counter load occurs after writing data to PDUTH0 (but before writing data to PDUTL0), the value of the 16-bit PPG duty setting buffer registers is not transferred to the duty setting registers.

The relation between the value of the 16-bit PPG duty setting registers and output pulse is as follows:

- When the same value is set in both the 16-bit PPG cycle setting buffer registers and duty setting registers, the "H" level will always be outputted if normal polarity is set, or the "L" level will always be outputted if inverted polarity is set.
- When the duty setting registers are set to "0000_H", the "L" level will always be outputted if normal polarity is set, or the "H" level will always be outputted if inverted polarity is set.
- When the value set in the duty setting registers is greater than the value in the 16-bit PPG cycle setting buffer registers, the "L" level will always be outputted if normal polarity is set, and the "H" level will always be outputted if inverted polarity is set.

17.5.4 16-bit PPG Status Control Register (PCNTH0, PCNTL0)

The 16-bit PPG status control register is used to enable and disable the 16-bit PPG timer and also to set the operating status for the software trigger, retrigger control interrupt, and output polarity. This register can also check the operation status.

■ 16-bit PPG Status Control Register, Upper Byte (PCNTH0)

Figure 17.5-5 16-bit PPG Status Control Register, Upper Byte (PCNTH0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCNTH0 0042_H	CNTE	STRG	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	00000000_B
	R/W	R0, W	R/W	R/W	R/W	R/W	R/W	R/W	

PGMS	PPG0 output mask enable bit
0	Disables PPG0 output mask
1	Enables PPG0 output mask

CKS2	CKS1	CKS0	Counter clock select bit
0	0	0	MCLK/1
0	0	1	MCLK/2
0	1	0	MCLK/4
0	1	1	MCLK/8
1	0	0	MCLK/16
1	0	1	MCLK/32
1	1	0	$F_{CH}/2^7$
1	1	1	$F_{CH}/2^8$

MCLK: Machine clock, F_{CH} : Main clock

RTRG	Software retrigger enable bit
0	Disables software retrigger
1	Enables software retrigger

MDSE	Mode select bit
0	PWM mode
1	One-shot mode

STRG	Software trigger bit	
	Write	Read
0	No effect on operation	Always reads "0"
1	Generates software trigger	

CNTE	Timer enable bit
0	Stops PPG timer
1	Enables PPG timer

R/W : Readable/Writable
(Read value is the same as write value)
R0, W: Write only (Writable, the read value is "0".)
▭ : Initial value

Table 17.5-1 16-bit PPG Status Control Register, Upper Byte (PCNTH0)

Bit name		Function
bit7	CNTE: Timer enable bit	This bit is used to enable/stop PPG timer operation. **When the bit is set to** "0", the PPG operation halts immediately and the PPG0 output goes to the initial level ("L" output if OSEL is "0"; "H" output if OSEL is "1"). **When the bit is set to** "1", PPG operation is enabled and the PPG goes to standby to wait for a trigger.
bit6	STRG: Software trigger bit	This bit is used to start the PPG timer by software. **When the bit is set to "1"**, setting the CNTE bit to "1" starts the PPG timer. Reading this bit always returns "0".
bit5	MDSE: Mode select bit	This bit is used to set the PPG operation mode. **When the bit is set to** "0", the PPG operates in PWM mode. **When the bit is set to** "1", the PPG operates in one-shot mode. Note: Modifying this bit is prohibited during operation.
bit4	RTRG: Software retrigger enable bit	This bit is used to enable or disable the software retrigger function of the PPG during operation. **When the bit is set to** "0", the software retrigger function is "disabled". **When the bit is set to** "1", the software retrigger function is "enabled".
bit3 to bit1	CKS2 to CKS0: Count clock select bits	These bits select the operating clock for the 16-bit PPG timer. The count clock signal is generated by the prescaler. Refer to "6.12 Operating Explanation of Prescaler". Note: As the time-base timer (TBT) is halted in sub clock mode, $F_{CH}/2^7$ and $F_{CH}/2^8$ cannot be selected in this case.
bit0	PGMS: PPG output mask enable bit	This bit is used to mask the PPG0 output to a specific level regardless of the mode setting (MDSE: bit5), period setting (PCSRH0, PCSRL0), and duty setting (PDUTH0, PDUTL0). **When the bit is set to** "0", the PPG0 output mask function is disabled. **When the bit is set to** "1", the PPG0 output mask function is enabled. When the PPG0 output polarity setting is set to "normal" (OSEL bit in PCNTL0 register = 0), the output is always masked to "L". When the polarity setting is se to "inverted" (OSEL bit in PCNTL0 register = 1), the PPG0 output is always masked to "H".

MB95150/M Series

■ 16-bit PPG Status Control Register, Lower Byte (PCNTL0)

Figure 17.5-6 16-bit PPG Status Control Register, Lower Byte (PCNTL0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCNTL0 0043_H	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	00000000_B
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

OSEL	Output inversion bit
0	Normal polarity
1	Inverted polarity

POEN	Output enable bit
0	General-purpose I/O port
1	PPG0 output pin

IRS1	IRS0	Interrupt type select bit
0	0	Trigger, software trigger, and retrigger by TRG0 input
0	1	Counter borrow
1	0	Rising edge of PPG output in normal polarity or falling edge of PPG output in inverted polarity (Duty match)
1	1	Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity

IRQF	PPG interrupt flag bit	
	Read	Write
0	No PPG interrupt	Clears this bit
1	PPG interrupt generated	No effect on operation

IREN	PPG interrupt request enable flag
0	Disables interrupt request
1	Enables interrupt request

EGS0	Hardware trigger enable bit0
0	The falling edge of TRG0 has no effect on operation
1	The operation is started by the falling edge of TRG0

EGS1	Hardware trigger enable bit1
0	The rising edge of TRG0 has no effect on operation
1	The operation is started by the rising edge of TRG0

R/W: Readable/Writable (Read value is the same as write value)
R(RM1), W: Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
(shaded) : Initial value

Table 17.5-2 16-bit PPG Status Control Register, Lower Byte (PCNTL0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>EGS1:
Hardware trigger enable bit1</td><td>This bit determines whether to allow or disallow the falling edge of TRG0 input to stop operation.
When the bit is set to "0", the falling edge of TRG0 has no effect on operation.
When the bit is set to "1", the operation is stopped by the falling edge of TRG0.</td></tr>
<tr><td>bit6</td><td>EGS0:
Hardware trigger enable bit0</td><td>This bit determines whether to allow or disallow the rising edge of TRG0 input to start operation.
When the bit is set to "0", the rising edge of TRG0 has no effect on operation.
When the bit is set to "1", the operation is started by the rising edge of TRG0.</td></tr>
<tr><td>bit5</td><td>IREN:
PPG interrupt request enable bit</td><td>This bit enables or disables PPG interrupt request to the interrupt controller.
When the bit is set to "0", an interrupt request is disabled.
When the bit is set to "1", an interrupt request is enabled.</td></tr>
<tr><td>bit4</td><td>IRQF:
PPG interrupt flag bit</td><td>This bit is set to "1" when a PPG interrupt occurs.
When the bit is set to "0", clears the bit.
When the bit is set to "1", has no effect on operation.
"1" is always read in read-modify-write (RMW) instruction.</td></tr>
<tr><td>bit3,
bit2</td><td>IRS1, IRS0:
Interrupt type select bits</td><td>These bits select the interrupt type for the PPG timer.
<table>
<tr><th>IRS1</th><th>IRS0</th><th>Type of interrupt</th></tr>
<tr><td>0</td><td>0</td><td>Trigger by TRG0 input, software trigger, or retrigger</td></tr>
<tr><td>0</td><td>1</td><td>Counter borrow</td></tr>
<tr><td>1</td><td>0</td><td>Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr>
<tr><td>1</td><td>1</td><td>Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr>
</table></td></tr>
<tr><td>bit1</td><td>POEN:
Output enable bit</td><td>This bit enables or disables output from the PPG0 output pin.
When the bit is set to "0", the pin serves as a general-purpose port.
When the bit is set to "1", the pin serves as the PPG timer output pin.</td></tr>
<tr><td>bit0</td><td>OSEL:
Output inversion bit</td><td>This bit selects the polarity of PPG0 output pin.
When the bit is set to "0", the PPG output goes to "H" when "L" is output in the internal start and the 16-bit down-counter value matches the duty setting register value, and goes to "L" when a down-counter borrow occurs (Normal polarity).
When the bit is set to "1", the PPG output is inverted (Inverted polarity).</td></tr>
</table>

17.6 Interrupts of 16-bit PPG Timer

The 16-bit PPG timer can generate interrupt requests in the following cases:
- **When a trigger or counter borrow occurs**
- **When a rising edge of PPG is generated in normal polarity**
- **When a falling edge of PPG is generated in inverted polarity**

The interrupt operation is controlled by IRS1 (bit3) and IRS0 (bit2) in the PCNTL register.

■ Interrupts of 16-bit PPG Timer

Table 17.6-1 shows interrupt control bits and interrupt sources of the 16-bit PPG timer.

Table 17.6-1 Interrupt Control Bits and Interrupt Sources of 16-bit PPG Timer

Item	Description
Interrupt flag bit	PCNTL0:IRQF
Interrupt request enable bit	PCNTL0:IREN
Interrupt type select bits	PCNTL0:IRS1, IRS0
Interrupt sources	PCNTL0:IRS1, IRS0=00_B Hardware trigger by TRG0 Pin input of 16-bit down-counter, software trigger and retrigger
	PCNTL0:IRS1, IRS0=01_B Counter borrow of 16-bit down-counter
	PCNTL0:IRS1, IRS0=10_B Rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity
	PCNTL0:IRS1, IRS0=11_B Counter borrow of 16-bit down-counter, rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity

When IRQF (bit4) in the 16-bit PPG status control register (PCNTL0) is set to "1" and interrupt requests are enabled (PCNTL0:IREN: bit5 = 1) in the 16-bit PPG timer, an interrupt request is generated and outputted to the controller.

■ Registers and Vector Table Related to Interrupts of 16-bit PPG Timer

Table 17.6-2 Registers and Vector Table Related to Interrupts of 16-bit PPG Timer

Interrupt source	Interrupt request No.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.0	IRQ15	ILR3	L15	$FFDC_H$	$FFDD_H$

ch. : Channel

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

17.7 Explanation of 16-bit PPG Timer Operations and Setup Procedure Example

The 16-bit PPG timer can operate in PWM mode or one-shot mode. In addition, a retrigger function can be used in the 16-bit PPG timer.

■ PWM Mode (MDSE of PCNTH Register: bit5 = 0)

In PWM operation mode, the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are loaded and the 16-bit down-counter starts down-count operation when a software trigger is inputted or a hardware trigger by TRG0 pin input is inputted. When the count value reaches "1", the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are reloaded to repeat the down-count operation.

The initial state of the PPG0 output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H" synchronizing with count clock. The output changes back to "L" when the "H" was output until the value of duty setting. (The output levels will be reversed if OSEL is set to "1".)

When the retrigger function is disabled (RTRG = 0), software triggers (STRG = 1) are ignored during the operation of the down-counter.

When the down-counter is not running, the maximum time between a valid trigger input occurring and the down-counter starting is as follows.

Software trigger: 1count clock cycle + 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 1 count clock cycle + 3 machine clock cycles

The minimum time is as follows.

Software trigger: 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 3 machine clock cycles

When the down-counter is running, the maximum time between a valid retrigger input occurring and the down-counter restarting is as follows.

Software trigger: 1 count clock cycle + 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 1 count clock cycle + 3 machine clock cycles

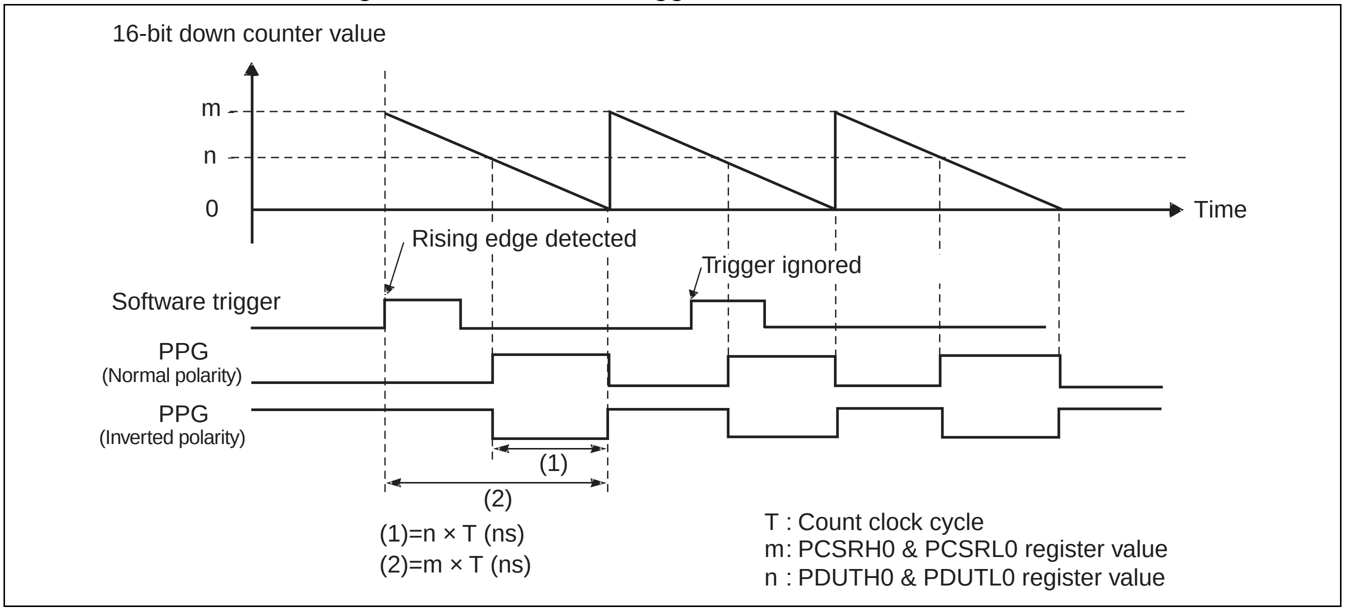
The minimum time is as follows.

Software trigger: 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 3 machine clock cycles

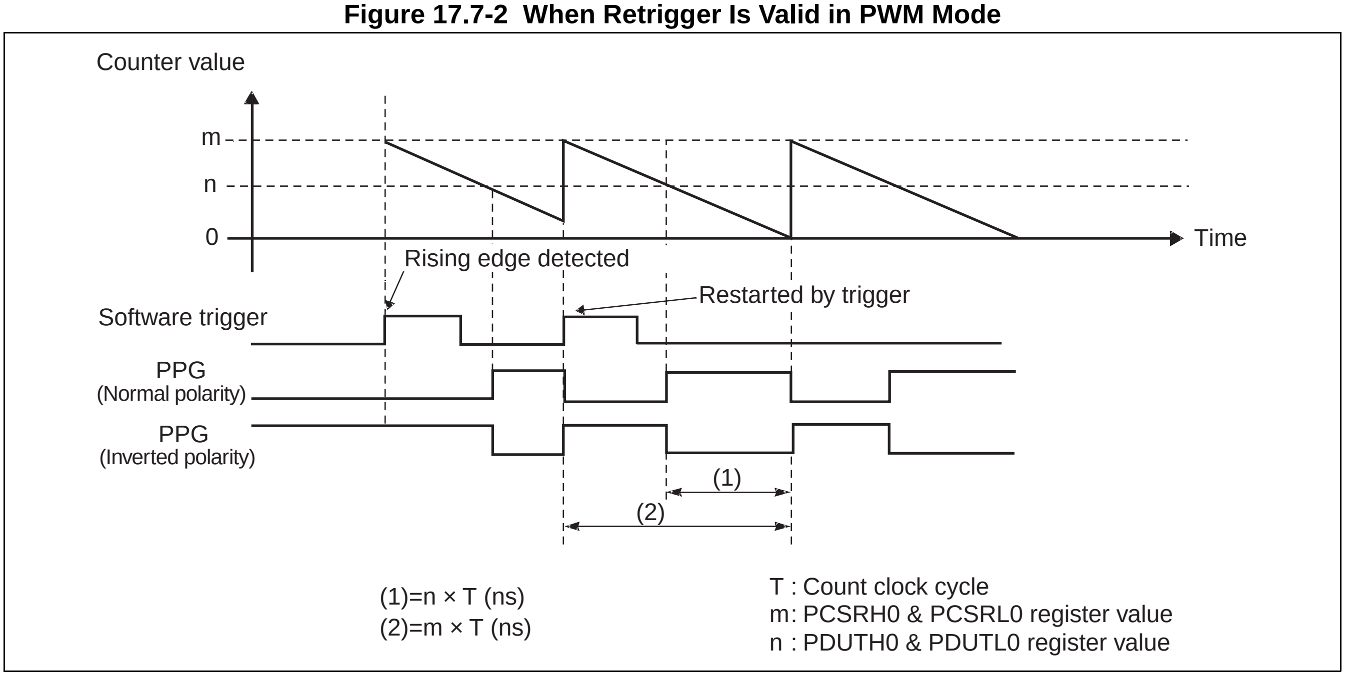
● Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 17.7-1 When Retrigger Is Invalid in PWM Mode



● Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 17.7-2 When Retrigger Is Valid in PWM Mode



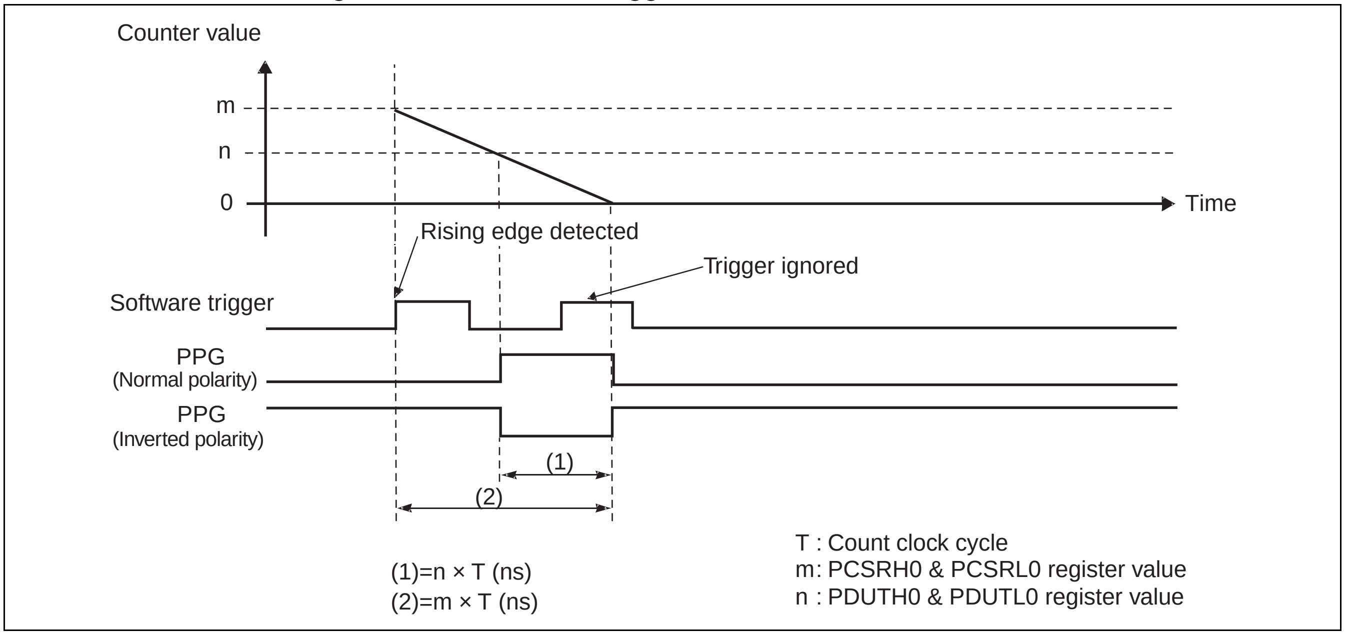
■ One-shot Mode (MDSE of PCNTH0 Register: bit5 = 1)

One-shot operation mode can be used to output a single pulse with a specified width when a valid trigger input occurs. When retriggering is enabled and a valid trigger is detected during the counter operation, the down counter value is reloaded.

The initial state of the PPG0 output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H". The output changes back to "L" when the counter reaches "1". (The output levels will be reversed if OSEL is set to "1".)

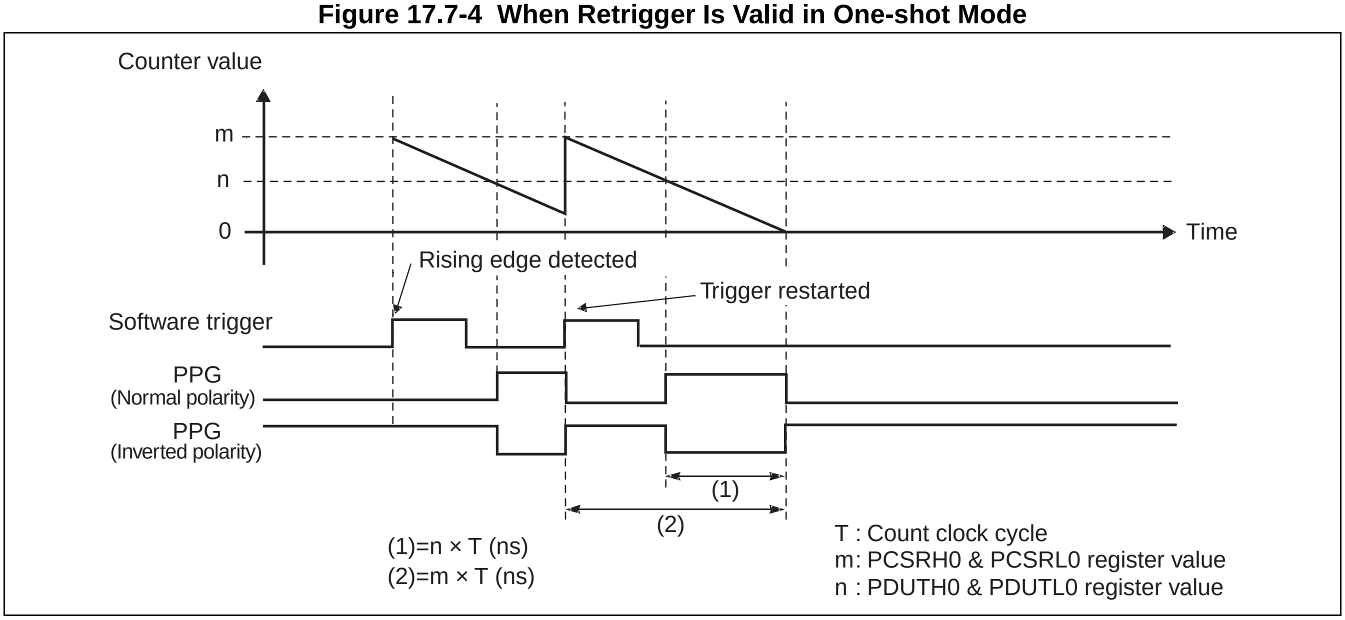
● Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 17.7-3 When Retrigger Is Invalid in One-shot Mode



● Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 17.7-4 When Retrigger Is Valid in One-shot Mode



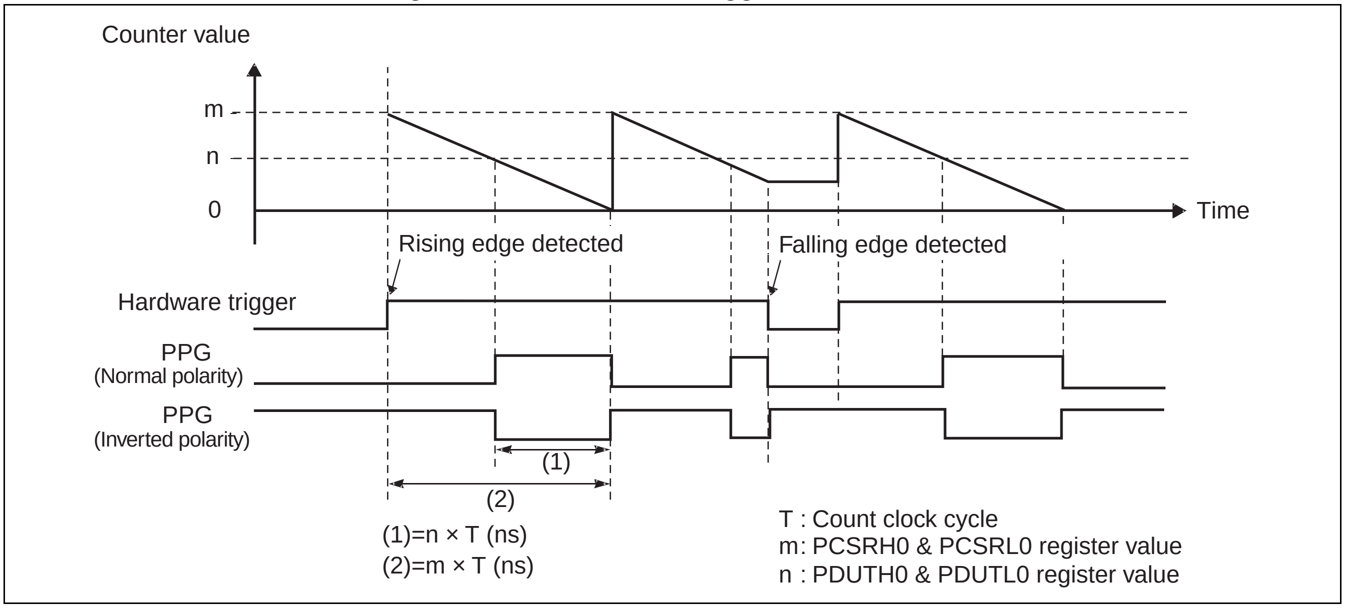
■ Hardware Trigger

"Hardware trigger" refers to PPG activation by signal input to the TRG0 input pin. When EGS1 and EGS0 are set to "11_B" and the hardware trigger is used with TRG0 input, PPG starts operation on a rising edge and halts the operation upon the detection of a falling edge.

Moreover, the PPG timer begins operation of the following rising edge from the beginning.

The operation can be retriggered by a valid TRG0 input hardware trigger regardless of the retrigger setting of the RTRG bit when the TRG0 input hardware trigger has been selected.

Figure 17.7-5 Hardware Trigger in PWM Mode



■ Setup Procedure Example

The 16-bit PPG timer is set up in the following procedure:

● Initial setting

1) Set the interrupt level (ILR3, ILR4)
2) Enable the hardware trigger and interrupts, select the interrupt type, and enable output (PCNTL0)
3) Select the count clock and the mode, and enable timer operation (PCNTH0)
4) Set the cycle (PCSRL0, PCSRH0)
5) Set the duty (PDUTH0, PDUTL0)
6) Start the PPG by the software trigger (PCNTH0:STRG = 1)

● Interrupt processing

1) Process any interrupt
2) Clear the interrupt request flag (PCNTL0:IRQF)

17.8 Notes on Using 16-bit PPG Timer

Shown below are the precautions that must be followed when using the 16-bit PPG timer.

■ Notes on Using 16-bit PPG Timer

● Precautions when setting the program

Do not use the retrigger if the same values are set for the cycle and duty. If used, the PPG0 output will go to the "L" level for one count clock cycle after the retrigger, and then go back to the "H" level when normal polarity has been selected.

If the microcontroller enters a standby mode, the TRG0 pin setting may change and cause the device to malfunction. Therefore, disable the timer enable bit (PCNTH0:CNTE = 0) or disable the hardware trigger enable bit (PCNTL0:EGS1, EGS0 = 00_B).

When the cycle and duty are set to the same value, an interrupt is generated only once by duty match. Moreover, if the duty is set to a value greater than the value of the period, no interrupt will be generated by duty match.

Do not disable the timer enable bit (PCNTH0: CNTE = 0) and software trigger (PCNTH0: STRG =1) at the same time when retrigger by the software is enabled (PCNTH0: RTRG =1) and the retrigger is selected as an interrupt type (PCNTL0: IRS1, IRS0 = 00_B) during count operation. If it occurs, interrupt flag bit may set by retrigger although timer stops.

17.9 Sample Programs for 16-bit PPG Timer

We provide sample programs that can be used to operate the 16-bit PPG timer.

■ Sample Programs for 16-bit PPG Timer

For information about the sample programs for the 16-bit PPG timer, refer to "■ Sample Programs" in PREFACE.

■ Setup Methods without Sample Program

● How to set the PPG operation mode

The operation mode select bit (PCNTH0:MDSE) is used.

Operation mode	Operation mode select bit (MDSE)
PWM mode	Set the bit to "0"
One-shot mode	Set the bit to "1"

● How to select the operating clock

The operating clock select bits (PCNTH0:CKS2/CKS1/CKS0) are used to select the clock.

● How to enable/disable the PPG output pin

The output enable bit (PCNTL0:POEN) is used.

Operation	Output enable bit (POEN)
When enabling PPG0 output	Set the bit to "1"
When disabling PPG0 output	Set the bit to "0"

● How to enable/disable PPG operation

The timer enable bit (PCNTH0:CNTE) is used.

Operation	Timer enable bit (CNTE)
When disabling PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

Enable PPG operation before starting the PPG.

● How to start PPG operation by software

The software trigger bit (PCNTH0:STRG) is used.

Operation	Software trigger bit (STRG)
When starting PPG operation by software	Set the bit to "1"

● How to enable/disable the retrigger function of the software trigger

The retrigger enable bit (PCNTH0:RTRG) is used.

Operation	Retrigger enable bit (RTRG)
When enabling retrigger function	Set the bit to "1"
When disabling retrigger function	Set the bit to "0"

● How to start/stop operation on a rising edge of trigger input

The hardware trigger enable bit (PCNTL0:EGS0) is used.

Operation	Hardware trigger enable bit (EGS0)
When starting operation on rising edge	Set the bit to "1"
When stopping operation on rising edge	Set the bit to "0"

● How to start/stop operation on a falling edge of trigger input

The hardware trigger enable bit (PCNTL0:EGS1) is used.

Operation	Hardware trigger enable bit (EGS1)
When starting operation on falling edge	Set the bit to "1"
When stopping operation on falling edge	Set the bit to "0"

● How to invert PPG output

The output inversion bit (PCNTL0:OSEL) is used.

Operation	Output inversion bit (OSEL)
When inverting PPG output	Set the bit to "1"

● How to set the PPG output to the "H" or "L" level

The PPG output mask enable bit (PCNTH0:PGMS) and the output inversion bit (PCNTL0:OSEL) are used.

Operation	PPG output mask enable bit (PGMS)	Output inversion bit (OSEL)
When setting output to "H" level	Set the bit to "1"	Set the bit to "1"
When setting output to "L" level	Set the bit to "1"	Set the bit to "0"

● How to select the interrupt source

The interrupt select bits (PCNTL0:IRS1/IRS0) are used to select the interrupt source.

Interrupt source	Interrupt select bits (IRS1/IRS0)
Trigger by TRG0 input, software trigger, or retrigger	Set the bits to "00_B"
Counter borrow	Set the bits to "01_B"
Rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity	Set the bits to "10_B"
Counter borrow, rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity	Set the bits to "11_B"

● Interrupt-related registers

The interrupt level is set by the level setting registers shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR3) Address: $0007C_H$	#15 Address: $0FFDC_H$

● How to enable/disable/clear interrupts

The interrupt request enable bit (PCNTL0:IREN) is used to enable interrupts.

Operation	Interrupt request enable bit (IREN)
When disabling interrupt request	Set the bit to "0"
When enabling interrupt request	Set the bit to "1"

The interrupt request flag (PCNTL0:IRQF) is used to clear interrupt requests.

Operation	Interrupt request flag (IRQF)
When clearing interrupt request	Write "0" to the bit

CHAPTER 18

EXTERNAL INTERRUPT CIRCUIT

This chapter describes the functions and operations of the external interrupt circuit.

18.1 Overview of External Interrupt Circuit

The external interrupt circuit detects edges on the signal that is inputted to the external interrupt pin and generates interrupt requests to the CPU.

■ Functions of External Interrupt Circuit

The external interrupt circuit has the functions to detect any edge of a signal that is inputted to an external interrupt pin and generate an interrupt request to the CPU. This interrupt allows the unit to recover from a standby mode and return to its normal operation.

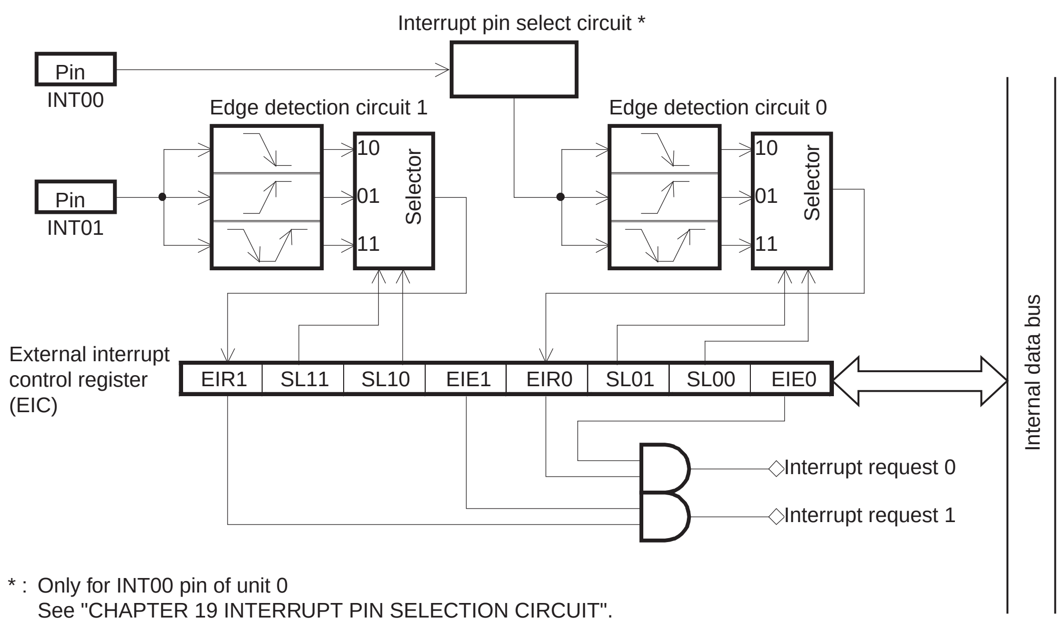
18.2 Configuration of External Interrupt Circuit

The external interrupt circuit consists of the following blocks:
- **Edge detection circuit**
- **External interrupt control register**

■ Block Diagram of External Interrupt Circuit

Figure 18.2-1 shows the block diagram of the external interrupt circuit.

Figure 18.2-1 Block Diagram of External Interrupt Circuit (Unit0)



● Edge detection circuit

When the polarity of the edge detected on a signal inputted to an external interrupt circuit pin (INT) matches the polarity of the edge selected in the interrupt control register (EIC), the corresponding external interrupt request flag bit (EIR) is set to "1".

● External interrupt control register (EIC)

This register is used to select the valid edge, enable or disable interrupt requests, check for interrupt requests, etc.

18.3 Channels of External Interrupt Circuit

This section describes the channels of the external interrupt circuit.

■ Channels of External Interrupt Circuit

In MB95150/M series, each unit has four channels of the external interrupt circuit.

Table 18.3-1 and Table 18.3-2 show the correspondence among the channel, pin and register.

Table 18.3-1 Pins of External Interrupt Circuit

Unit	Pin name	Pin function
0	INT00	External interrupt input ch.0
	INT01	External interrupt input ch.1
1	INT02	External interrupt input ch.2
	INT03	External interrupt input ch.3
2	INT04	External interrupt input ch.4
	INT05	External interrupt input ch.5
3	INT06	External interrupt input ch.6
	INT07	External interrupt input ch.7

Table 18.3-2 Registers of External Interrupt Circuit

Unit	Register name	Corresponding register (Name in this manual)
0	EIC00	EIC: External Interrupt Control register
1	EIC10	
2	EIC20	
3	EIC30	

The following sections only describe the unit 0 side of the external interrupt circuit.

The other units are the same as the unit 0 side of the external interrupt circuit.

18.4 Pins of External Interrupt Circuit

This section shows the pins related to the external interrupt circuit and the block diagram of such pins.

■ Pins Related to External Interrupt Circuit

The pins related to the external interrupt circuit are the INT00 to INT07 pins.

● INT00 to INT07 pins

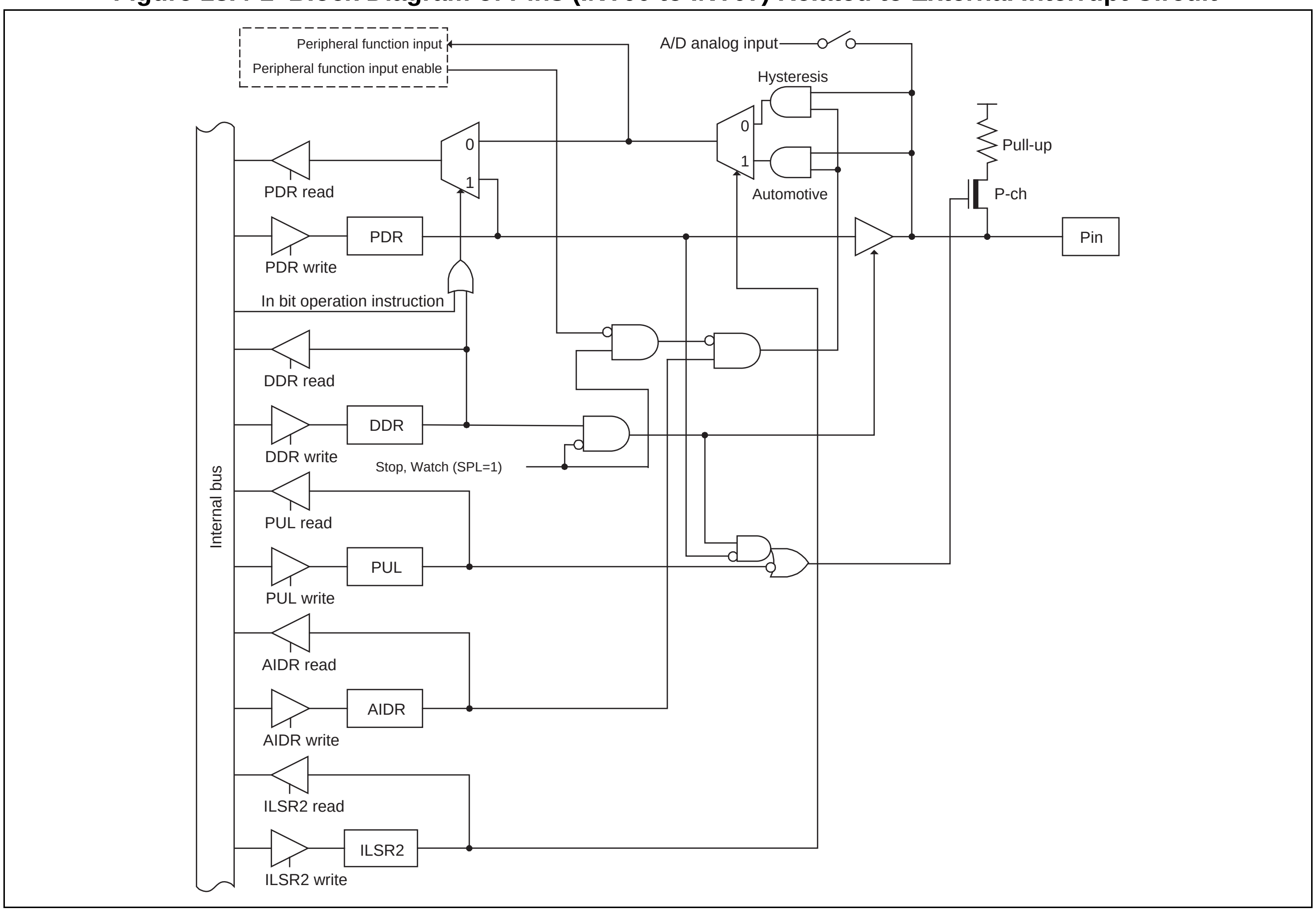
These pins serve both as external interrupt inputs and as general-purpose I/O ports.

INT00 to INT07: When the corresponding pin of the INT00 to INT07 pins is set as an input port by the port direction register (DDR) and the corresponding external interrupt input is enabled by the external interrupt control register (EIC), that pin functions as an external interrupt input pin (INT00 = INT07).

The state of pins can be read from the port data register (PDR) whenever input port is set as a pin function. However, the value of PDR is read when read-modify-write (RMW) instruction is used.

■ Block Diagram of Pins Related to External Interrupt Circuit

Figure 18.4-1 Block Diagram of Pins (INT00 to INT07) Related to External Interrupt Circuit



18.5 Registers of External Interrupt Circuit

This section describes the registers of the external interrupt circuit.

■ List of Registers of External Interrupt Circuit

Figure 18.5-1 shows the registers of the external interrupt circuit.

Figure 18.5-1 Registers of External Interrupt Circuit

External interrupt control register (EIC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC00 0048_H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC10 0049_H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC20 $004A_H$	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC30 $004B_H$	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)
R(RM1), W: Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

18.5.1 External Interrupt Control Register (EIC00)

The external interrupt control register (EIC00) is used to select the edge polarity for the external interrupt input and control interrupts.

■ External Interrupt Control Register (EIC00)

Figure 18.5-2 External Interrupt Control Register (EIC00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC00 0048_H EIC10 0049_H EIC20 $004A_H$ EIC30 $004B_H$	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

EIE0	Interrupt request enable bit 0
0	Disables output of interrupt request
1	Enables output of interrupt request

SL01	SL00	Edge polarity select bits 0
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR0	External interrupt request flag bit 0	
	Read	Write
0	Specified edge not inputted	Clears this bit
1	Specified edge inputted	No change, no effect on others

EIE1	Interrupt request enable bit 1
0	Disables output of interrupt request
1	Enables output of interrupt request

SL11	SL10	Edge polarity select bits 1
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR1	External interrupt request flag bit 1	
	Read	Write
0	Specified edge not inputted	Clears this bit
1	Specified edge inputted	No change, no effect on others

R/W : Readable/Writable (Read value is the same as write value)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
(shaded box) : Initial value

Table 18.5-1 Functional Description of Each Bit of External Interrupt Control Register (EIC00)

Bit name		Function
bit7	EIR1: External interrupt request flag bit 1	This flag is set to "1" when the edge selected by the edge polarity select bits (SL11, SL10) is inputted to the external interrupt pin INT01. • When this bit and the interrupt request enable bit 1 (EIE1) are set to "1", an interrupt request is outputted. • Writing "0" clears the bit. Writing "1" has no effect. • "1" is read in read-modify-write (RMW) instructions.
bit6, bit5	SL11, SL10: Edge polarity select bits 1	These bits select the polarity of the interrupt source edge of the pulse inputted to the external interrupt pin INT01. • Edge detection is not performed and no interrupt is generated when these bits are set to "00_B". • Rising edges are detected when these bits are "01_B", falling edges when "10_B", and both edges when "11_B".
bit4	EIE1: Interrupt request enable bit 1	This bit is used to enable and disable output of interrupt requests to the interrupt controller. When this bit and the external interrupt request flag bit 1 (EIR1) are "1", an interrupt request is outputted. • When using an external interrupt pin, write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input. • The status of the external interrupt pin can be read directly from the port data register, regardless of the status of the interrupt request enable bit.
bit3	EIR0: External interrupt request flag bit 0	This flag is set to "1" when the edge selected by the edge polarity select bits (SL01, SL00) is inputted to the external interrupt pin INT00. • When this bit and the interrupt request enable bit 0 (EIE0) are set to "1", an interrupt request is outputted. • Writing "0" clears the bit. Writing "1" has no effect. • "1" is read in read-modify-write (RMW) instructions.
bit2, bit1	SL01, SL00: Edge polarity select bits 0	These bits are used to select the polarity of the interrupt source edge of the pulse inputted to the external interrupt pin INT00. • Edge detection is not performed and no interrupt request is generated when these bits are "00_B". • Rising edges are detected when the bits are "01_B", falling edges when "10_B", and both edges when "11_B".
bit0	EIE0: Interrupt request enable bit 0	This bit enables or disables the output of interrupt requests to the interrupt controller. An interrupt request is outputted when this bit and the external interrupt request flag bit 0 (EIR0) are "1". • When using an external interrupt pin, write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input. • The status of the external interrupt pin can be read directly from the port data register (PFR), regardless of the status of the interrupt request enable bit.

18.6 Interrupts of External Interrupt Circuit

The interrupt sources for the external interrupt circuit include detection of the specified edge of the signal inputted to an external interrupt pin.

■ Interrupt During Operation of External Interrupt Circuit

When the specified edge of external interrupt input is detected, the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1". In this case, an interrupt request will be generated to the interrupt controller, if the corresponding interrupt request enable bit is enabled (EIC: EIE0, EIE1=1). Write "0" to the corresponding external interrupt request flag big to clear the interrupt request in the interrupt process routine.

■ Registers and Vector Table Related to Interrupts of External Interrupt Circuit

Table 18.6-1 Registers and Vector Table Related to Interrupts of External Interrupt Circuit

Interrupt source	Interrupt request No.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.0	IRQ0	ILR0	L00	$FFFA_H$	$FFFB_H$
ch.4					
ch.1	IRQ1	ILR0	L01	$FFF8_H$	$FFF9_H$
ch.5					
ch.2	IRQ2	ILR0	L02	$FFF6_H$	$FFF7_H$
ch.6					
ch.3	IRQ3	ILR0	L03	$FFF4_H$	$FFF5_H$
ch.7					

ch.: Channel

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

18.7 Explanation of External Interrupt Circuit Operations and Setup Procedure Example

This section describes the operation of the external interrupt circuit.

■ Operation of External Interrupt Circuit

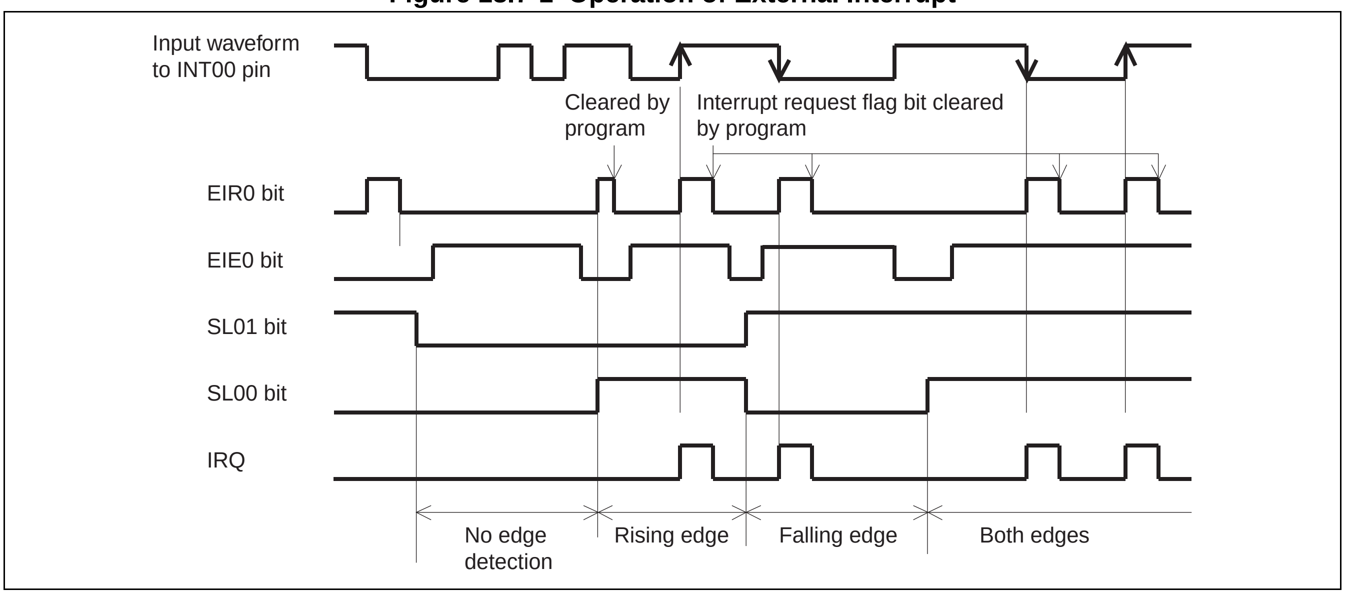
When the polarity of an edge of a signal inputted from one of the external interrupt pins (INT00, INT01) matches the polarity of the edge selected by the external interrupt control register (EIC: SL00, SL01, SL10, SL11), the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1" and the interrupt request is generated.

Always set the interrupt enable bit to "0" when not using an external interrupt to recover from a standby mode.

When setting the edge polarity select bit (SL), set the interrupt request enable bit (EIE) to "0" to prevent the interrupt request from being generated accidentally. Also clear the interrupt request flag bit (EIR) to "0" after changing the edge polarity.

Figure 18.7-1 shows the operation for setting the INT00 pin as an external interrupt input.

Figure 18.7-1 Operation of External Interrupt



■ Setup Procedure Example

The external interrupt circuit is set up in the following procedure:

● Initial setup

1) Set the interrupt level. (ILR0)
2) Select the edge polarity. (EIC:SL01, SL00)
3) Enable interrupt requests. (EIC:EIE0 = 1)

● Interrupt processing

1) Clear the interrupt request flag. (EIC:EIR0 = 0)
2) Process any interrupt.

Note:

The external interrupt input is also used as an I/O port. Therefore, when it is used as the external interrupt input, the corresponding bit in the port direction register (DDR) must be set to "0" (input).

18.8 Notes on Using External Interrupt Circuit

This section describes the precautions that must be followed when using the external interrupt circuit.

■ Notes on Using External Interrupt Circuit

- Set the interrupt request enable bit (EIE) to "0" (disabling interrupt requests) when setting the edge polarity select bit (SL). Also clear the external interrupt request flag bit (EIR) to "0" after setting the edge polarity.
- The operation cannot recover from the interrupt processing routine if the external interrupt request flag bit is "1" and the interrupt request enable bit is enabled. Always clear the external interrupt request flag bit in the interrupt processing routine.

18.9 Sample Programs for External Interrupt Circuit

We provide sample programs that can be used to operate the external interrupt circuit.

■ Sample Programs for External Interrupt Circuit

For information about the sample programs for the external interrupt circuit, refer to "■ Sample Programs" in PREFACE.

■ Setup Methods without Sample Program

● Detection levels and setup methods

Four detection levels are available: no edge detection, rising edge, falling edge, both edges

The detection level bits (EIC: SL01, SL00 or EIC:SL11, SL10) are used.

Operation mode	Detection level bits (EIC: SL01, SL00 or EIC:SL11, SL10)
No edge detection	Set to "00_B"
Detecting rising edges	Set to "01_B"
Detecting falling edges	Set to "10_B"
Detecting both edges	Set to "11_B"

● How to use the external interrupt pin

Set the corresponding data direction register (DDR0) to "0".

Operation	Direction bit (P00 to P07)	Setting
Using INT00 pin for external interrupt	DDR0:P00	Set the register to "0"
Using INT01 pin for external interrupt	DDR0:P01	Set the register to "0"
Using INT02 pin for external interrupt	DDR0:P02	Set the register to "0"
Using INT03 pin for external interrupt	DDR0:P03	Set the register to "0"
Using INT04 pin for external interrupt	DDR0:P04	Set the register to "0"
Using INT05 pin for external interrupt	DDR0:P05	Set the register to "0"
Using INT06 pin for external interrupt	DDR0:P06	Set the register to "0"
Using INT07 pin for external interrupt	DDR0:P07	Set the register to "0"

- Interrupt-related registers

The interrupt level is set by the interrupt level setting registers shown in the following table.

Channel no.	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR0) Address: 00079_H	#0 Address: $0FFFA_H$
ch.1	Interrupt level register (ILR0) Address: 00079_H	#1 Address: $0FFF8_H$
ch.2	Interrupt level register (ILR0) Address: 00079_H	#2 Address: $0FFF6_H$
ch.3	Interrupt level register (ILR0) Address: 00079_H	#3 Address: $0FFF4_H$
ch.4	Interrupt level register (ILR0) Address: 00079_H	#0 Address: $0FFFA_H$
ch.5	Interrupt level register (ILR0) Address: 00079_H	#1 Address: $0FFF8_H$
ch.6	Interrupt level register (ILR0) Address: 00079_H	#2 Address: $0FFF6_H$
ch.7	Interrupt level register (ILR0) Address: 00079_H	#3 Address: $0FFF4_H$

- How to enable/disable/clear interrupts

Interrupts are enabled by the interrupt enable bit (EIC00:EIE0 or EIC00:EIE1).

Operation	Interrupt enable bit (EIE0 or EIE1)
When disabling interrupt request	Set the bit to "0"
When enabling interrupt request	Set the bit to "1"

Interrupt requests are cleared by the interrupt request bit (EIC00:EIR0 or EIC00:EIR1).

Operation	Interrupt request bit (EIR0 or EIR1)
When clearing interrupt request	Write "0"

CHAPTER 19
INTERRUPT PIN SELECTION CIRCUIT

This chapter describes the functions and operations of the interrupt pin selection circuit.

Code: CM26-00110-2E

19.1 Overview of Interrupt Pin Selection Circuit

The interrupt pin selection circuit selects pins to be used as interrupt input pins from among various peripheral input pins.

■ Interrupt Pin Selection Circuit

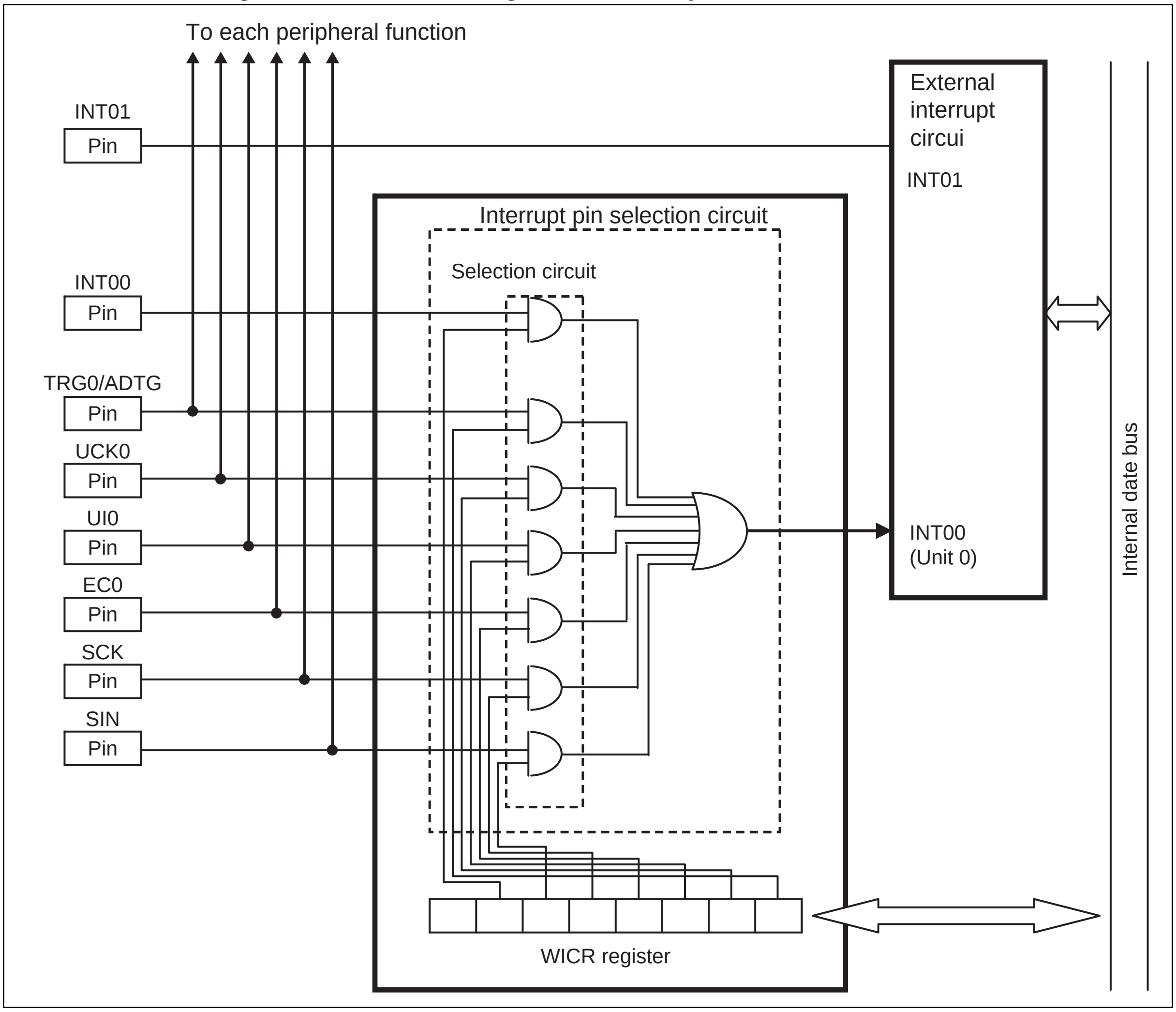
The interrupt pin selection circuit is used to select interrupt input pins from amongst various peripheral inputs (TRG0/ADTG, UCK0, UI0, EC0, SCK, SIN, INT00). The input signal from each peripheral function pin is selected by this circuit and the signal is used as the INT00 (channel 0) input of external interrupt. This enables the input signals to the peripheral function pins to also serve as external interrupt pins.

19.2 Configuration of Interrupt Pin Selection Circuit

Figure 19.2-1 shows the block diagram of the interrupt pin selection circuit.

■ Block Diagram of Interrupt Pin Selection Circuit

Figure 19.2-1 Block Diagram of Interrupt Pin Selection Circuit



- WICR register (interrupt pin selection circuit control register)

  This register is used to determine which of the available peripheral input pins should be outputted to the interrupt circuit and which interrupt pins they should serve as.

- Selection circuit

  This circuit outputs the input from the pin selected by the WICR register to the INT00 input of the external interrupt circuit (ch.0).

19.3 Pins of Interrupt Pin Selection Circuit

This section describes the pins of the interrupt pin selection circuit.

■ Pins Related to Interrupt Pin Selection Circuit

The peripheral function pins related to the interrupt pin selection circuit are the TRG0/ADTG, UCK0, UI0, EC0, SCK, SIN, and INT00 pins. These inputs (except INT00) are also connected to their respective peripheral units in parallel and can be used for both functions simultaneously. Table 19.3-1 lists the correlation between the peripheral functions and peripheral input pins.

Table 19.3-1 Correlation Between Peripheral Functions and Peripheral Input Pins

Peripheral input pin name	Peripheral functions name
INT00	Interrupt pin selection circuit
TRG0/ADTG	Interrupt pin selection circuit 16-bit PPG timer (trigger input) 8/10-bit A/D converter (trigger input)
UCK0	Interrupt pin selection circuit UART/SIO (clock input/output)
UI0	Interrupt pin selection circuit UART/SIO (data input)
EC0	Interrupt pin selection circuit 8/16-bit compound timer (event input)
SCK	Interrupt pin selection circuit LIN-UART (clock input/output)
SIN	Interrupt pin selection circuit LIN-UART (data input)

19.4 Registers of Interrupt Pin Selection Circuit

Figure 19.4-1 shows the registers related to the interrupt pin selection circuit.

■ Registers Related to Interrupt Pin Selection Circuit

Figure 19.4-1 Registers Related to Interrupt Pin Selection Circuit

Interrupt pin selection circuit control register (WICR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FEF_H$	–	INT00	SIN	SCK	EC0	UI0	UCK0	TRG0	01000000_B
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

19.4.1 Interrupt Pin Selection Circuit Control Register (WICR)

This register is used to determine which of the available peripheral input pins should be outputted to the interrupt circuit and which interrupt pins they should serve as.

■ Interrupt Pin Selection Circuit Control Register (WICR)

Figure 19.4-2 Interrupt Pin Selection Circuit Control Register (WICR)

Interrupt pin selection circuit control register (WICR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FEFH	-	INT00	SI	SCK	EC0	UI0	UCK0	TRG0	01000000B
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TRG0	TRG0 interrupt pin select bit
0	Deselects TRG0 as interrupt input pin
1	Selects TRG0 as interrupt input pin

UCK0	UCK0 interrupt pin select bit
0	Deselects UCK0 as interrupt input pin
1	Selects UCK0 as interrupt input pin

UI0	UI0 interrupt pin select bit
0	Deselects UI0 as interrupt input pin
1	Selects UI0 as interrupt input pin

EC0	ECO interrupt pin select bit
0	Deselects ECO as interrupt input pin
1	Selects ECO as interrupt input pin

SCK	SCK interrupt pin select bit
0	Deselects SCK as interrupt input pin
1	Selects SCK as interrupt input pin

SIN	SIN interrupt pin select bit
0	Deselects SIN as interrupt input pin
1	Selects SIN as interrupt input pin

INT00	INT00 interrupt pin select bit
0	Deselects INT00 as interrupt input pin
1	Selects INT00 as interrupt input pin

R/W : Readable/writable (Read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
(shaded) : Initial value

Table 19.4-1 Functional Description of Each Bit of Interrupt Pin Selection Circuit Control Register (WICR) (1 / 2)

Bit name		Function
bit7	Undefined bit	This bit is undefined. • The read value is always "0". • Writing has no effect on the operation.
bit6	INT00: IINT00 interrupt pin select bit	This bit is used to determine whether to select the INT00 pin as an interrupt input pin. **Setting the bit to "0":** Deselects the INT00 pin as an interrupt input pin and the circuit treats the INT00 pin input as being fixed at "0". **Setting the bit to "1":** Selects the INT00 pin as an interrupt input pin and the circuit passes the INT00 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the INT00 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.
bit5	SIN: SIN interrupt pin select bit	This bit is used to determine whether to select the SIN pin as an interrupt input pin. **Setting the bit to "0":** Deselects the SIN pin as an interrupt input pin and the circuit treats the SIN pin input as being fixed at "0". **Setting the bit to "1":** Selects the SIN pin as an interrupt input pin and the circuit passes the SIN pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SIN pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.
bit4	SCK: SCK interrupt pin select bit	This bit is used to determine whether to select the SCK pin as an interrupt input pin. **Setting the bit to "0":** Deselects the SCK pin as an interrupt input pin and the circuit treats the SCK pin input as being fixed at "0". **Setting the bit to "1":** Selects the SCK pin as an interrupt input pin and the circuit passes the SCK pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SCK pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.
bit3	EC0: EC0 interrupt pin select bits	This bit is used to determine whether to select the EC0 pin as an interrupt input pin. **Setting the bit to "0":** Deselects the EC0 pin as an interrupt input pin and the circuit treats the EC0 pin input as being fixed at "0". **Setting the bit to "1":** Selects the EC0 pin as an interrupt input pin and the circuit passes the EC0 pin input to INT000 (ch.0) of the external interrupt circuit. In this case, the input signal to the EC0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.
bit2	UI0: UI0 interrupt pin select bits	This bit is used to determine whether to select the UI0 pin as an interrupt input pin. **Setting the bit to "0":** Deselects theUI0 pin as an interrupt input pin and the circuit treats the UI0 pin input as being fixed at "0". **Setting the bit to "1":** Selects the UI0 pin as an interrupt input pin and the circuit passes the UI0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the UI0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.
bit1	UCK0: UCK0 interrupt pin select bit	This bit is used to determine whether to select the UCK0 pin as an interrupt input pin. **Setting the bit to "0":** Deselects theUCK0 pin as an interrupt input pin and the circuit treats the UCK0 pin input as being fixed at "0". **Setting the bit to "1":** Selects the UCK0 pin as an interrupt input pin and the circuit passes the UCK0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the UCK0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.

Table 19.4-1 Functional Description of Each Bit of Interrupt Pin Selection Circuit Control Register (WICR) (2 / 2)

Bit name		Function
bit0	TRG0: TRG0 interrupt pin select bit	This bit is used to determine whether to select the TRG0 pin as an interrupt input pin. **Setting the bit to "0":** Deselects the TRG0 pin as an interrupt input pin and the circuit treats the TRG0 pin input as being fixed at "0". **Setting the bit to "1":** Selects the TRG0 pin as an interrupt input pin and the circuit passes the TRG0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SCK pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.

When these bits are set to "1" and the operation of INT00 (ch.0) of the external interrupt circuit is enabled in MCU standby mode, the selected pins are enabled to perform input operation. The MCU wakes up from the standby mode when a valid edge pulse is inputted to the pins. For information about the standby modes, refer to "6.8 Operations in Low-power Consumption Modes (Standby Modes)".

Note:

The input signals to the peripheral pins do not generate an external interrupt even when "1" is written to these bits if the INT00 (ch.0) of the external interrupt circuit is disabled.

Do not modify the values of these bits while the INT00 (ch.0) of the external interrupt circuit is enabled. If modified, the external interrupt circuit may detect a valid edge, depending on the pin input level.
If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch.0) of the external interrupt circuit is enabled (the values other than "00_B" are set to SL01, SL00 bits in EIC00 register of external interrupt circuit and the interrupt is enabled by writing "1" to the EIE0 bit when selecting the valid edge), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.

19.5 Operating Description of Interrupt Pin Selection Circuit

The interrupt pins are selected by setting WICR (interrupt pin selection circuit control register).

■ Operation of Interrupt Pin Selection Circuit

The WICR (interrupt pin selection circuit control register) setting is used to select the input pins to be inputted to INT00 of the external interrupt circuit (ch.0). Shown below is the setup procedure for the interrupt pin selection circuit and external interrupt circuit (ch.0), which must be followed when selecting the TRG0 pin as an interrupt pin.

1) Write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input.
2) Select the TRG0 pin as an interrupt input pin in WICR (interrupt pin selection circuit control register).
   (Write "01_H" to the WICR register. At this point, after writing "0" in the EIE0 bit of the EIC00 register of the external interrupt circuit, the operation of the external interrupt circuit is disabled).
3) Enable the operation of INT00 of the external interrupt circuit (ch.0).
   (Set the SL01 and SL00 bits of the EIC00 register to any value other than "00_B" in the external interrupt circuit to select the valid edge. Also write "1" to the EIE0 bit to enable interrupts).
4) The subsequent interrupt operation is the same as for the external interrupt circuit.

When a reset is released, WICR (interrupt pin selection circuit control register) is initialized to "40_H" and the INT00 bit is selected as the only available interrupt pin. Update the value of this register before enabling the operation of the external interrupt circuit, when using any pins other than the INT00 pin as external interrupt pins.

Note:

If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously, an input to INT00 (ch.0) of the external interrupt circuit is treated as "H" if any of the selected input signals is "H". (It becomes "OR" of the signals inputted to the selected pins.)

19.6 Notes on Using Interrupt Pin Selection Circuit

This section explains the precautions to be taken when using the interrupt pin selection circuit.

■ Notes on Using Interrupt Pin Selection Circuit

- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch.0) of the external interrupt circuit is enabled (the values other than "00_B" are set to SL01, SL00 bits in EIC00 register of external interrupt circuit and the interrupt is enabled by writing "1" to the EIE0 bit when selecting the valid edge), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.
- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously, an input to INT00 (ch.0) of the external interrupt circuit is treated as "H" level if any of the selected input signals is "H" level (it becomes "OR" of the signals inputted to the selected pins).

CHAPTER 20
UART/SIO

This chapter describes the functions and operations of UART/SIO.

20.1 Overview of UART/SIO

The UART/SIO is a general-purpose serial data communication interface. Serial data transfers of variable-length data can be made with a synchronous or asynchronous clock. The transfer format is NRZ. The transfer rate can be set with the dedicated baud rate generator or external clock (in clock synchronous mode).

■ Functions of UART/SIO

The UART/SIO is capable of serial data transmission/reception (serial input/output) to and from another CPU or peripheral device.

- Equipped with a full-duplex double buffer that allows 2-way full-duplex communication.
- The synchronous or asynchronous transfer mode can be selected.
- The optimum baud rate can be selected with the dedicated baud rate generator.
- The data length is variable; it can be set to 5 bits to 8 bits when no parity is used or to 6 bits to 9 bits when parity is used. (Refer to Table 20.1-1).
- The serial data direction (endian) can be selected.
- The data transfer format is NRZ (Non-Return-to-Zero).
- Two operation modes (operation modes 0 and 1) are available.
  Operation mode 0 operates as asynchronous clock mode (UART).
  Operation mode 1 operates as clock synchronous mode (SIO).

Table 20.1-1 Operation Modes of UART/SIO

Operation mode	Data length		Synchronous mode	Stop bit length
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	–	Synchronous	–
	6	–		
	7	–		
	8	–		

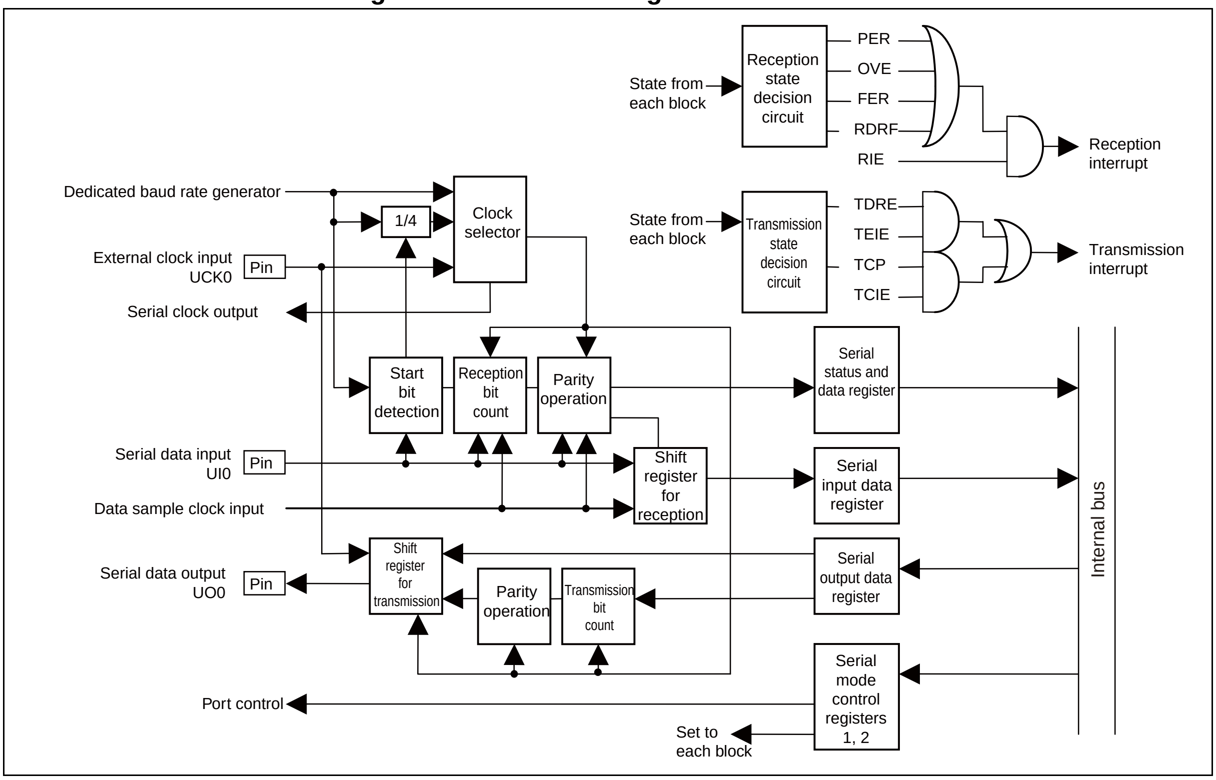
20.2 Configuration of UART/SIO

The UART/SIO consists of the following blocks:

- **UART/SIO serial mode control register 1 (SMC10)**
- **UART/SIO serial mode control register 2 (SMC20)**
- **UART/SIO serial status and data register (SSR0)**
- **UART/SIO serial input data register (RDR0)**
- **UART/SIO serial output data register (TDR0)**

■ Block Diagram of UART/SIO

Figure 20.2-1 Block Diagram of UART/SIO



● UART/SIO serial mode control register 1 (SMC10)

This register controls UART/SIO operation mode. The register is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/asynchronous), data length, and serial clock.

● UART/SIO serial mode control register 2 (SMC20)

This register controls UART/SIO operation mode.It is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

● UART/SIO serial status and data register (SSR0)

This register indicates the transmission/reception status and error status of UART/SIO.

● UART/SIO serial input data register (RDR0)

This register holds the receive data. The serial input is converted and then stored in this register.

● UART/SIO serial output data register (TDR0)

This register sets the transmit data. Data written to this register is serial-converted and then outputted.

■ Input Clock

The UART/SIO uses the output clock (internal clock) from the dedicated baud rate generator or the input signal (external clock) from the UCK0 pin as its input clock (serial clock).

20.3 Channels of UART/SIO

This section describes the channels of UART/SIO.

■ Channels of UART/SIO

This series contains one channel of the UART/SIO.

Table 20.3-1 and Table 20.3-2show the correspondence of sthe channel, pin, and register.

Table 20.3-1 Pins of UART/SIO

Channel	Pin name	Pin function
0	UCK0	Clock input/output
	UO0	Data output
	UI0	Data input

Table 20.3-2 Registers of UART/SIO

Channel	Register name	Corresponding register (Representation in this manual)
0	SMC10	UART/SIO serial mode control register 1
	SMC20	UART/SIO serial mode control register 2
	SSR0	UART/SIO serial status and data register
	TDR0	UART/SIO serial output data register
	RDR0	UART/SIO serial input data register

20.4 Pins of UART/SIO

This section describes the pins related to the UART/SIO.

■ Pins Related to UART/SIO

The pins associated with UART/SIO are the clock input and output pin (UCK0), serial data output pin (UO0) and serial data input pin (UI0).

UCK0:

Clock input/output pin for UART/SIO.

When the clock output is enabled (SMC20:SCKE=1), it serves as a UART/SIO clock output pin (UCK0) regardless of the value of the corresponding port direction register. At this time, do not select the external clock (set SMC10:CKS = 0).

When it is to be used as a UART/SIO clock input pin, disable the clock output (SMC20:SCKE = 0) and make sure that it is set as input port by the corresponding port direction register. At this time, be sure to select the external clock (set SMC10:CKS = 0).

UO0:

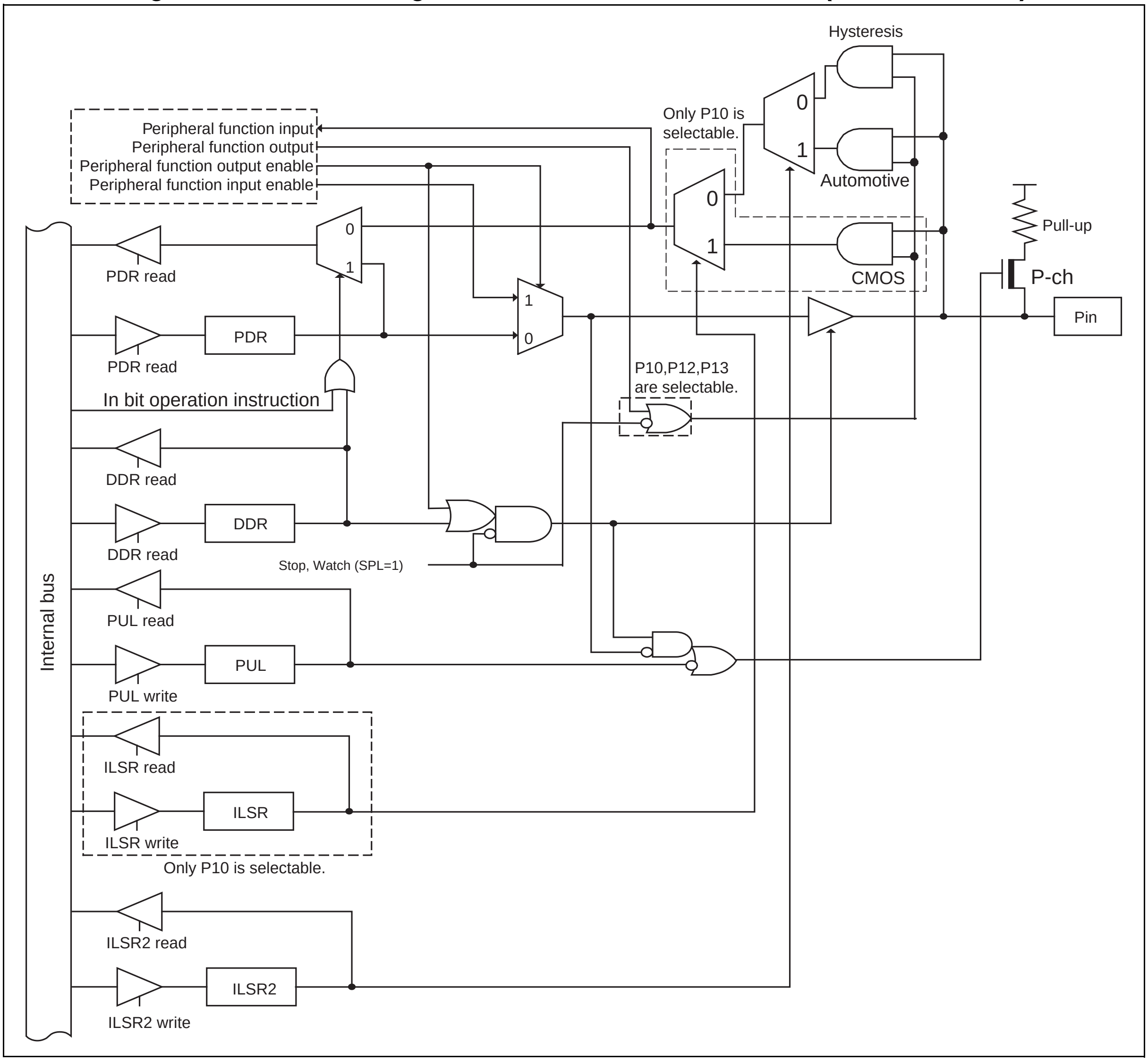
Serial data output pin for UART/SIO. When the serial data output is enabled (SMC20:TXOE = 1), it serves as a UART/SIO serial data output pin (UO0) regardless of the value of the corresponding port direction register.

UI0:

Serial data input pin for UART/SIO. When it is to be used as a UART/SIO serial data input pin, make sure that it is set as input port by the corresponding port direction register.

■ Block Diagram of Pins Related to UART/SIO

Figure 20.4-1 Block Diagram of Pins Related to UART/SIO (UI0, UO0, UCK0)



20.5 Registers of UART/SIO

The registers related to UART/SIO are UART/SIO serial mode control register 1 (SMC10), UART/SIO serial mode control register 2 (SMC20), UART/SIO serial status and data register (SSR0), UART/SIO serial output data register (TDR0), and UART/SIO serial input data register (RDR0).

■ Registers Related to UART/SIO

Figure 20.5-1 Registers Related to UART/SIO

UART/SIO serial mode control register 1 (SMC10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0056_H	BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial mode control register 2 (SMC20)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0057_H	SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE	00100000_B
	R/W	R/W	R1/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial status and data register (SSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0058_H	–	–	PER	OVE	FER	RDRF	TCPL	TDRE	00000001_B
	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R(RM1),W	R/WX	

UART/SIO serial output data register (TDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0059_H	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial input data register (RDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005A_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
R1/W : Readable/writable (Read value is always "1")

20.5.1 UART/SIO Serial Mode Control Register 1 (SMC10)

UART/SIO serial mode control register 1(SMC10) controls the UART/SIO operation mode. The register is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/ asynchronous), data length, and serial clock.

■ UART/SIO Serial Mode Control Register 1 (SMC10)

Figure 20.5-2 UART/SIO Serial Mode Control Register 1 (SMC10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0056H	BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

MD	Operation mode selection bit
0	Clock asynchronous mode (UART)
1	Clock synchronous mode (SIO)

CKS	Clock selection bit
0	Dedicated baud rate generator
1	External clock (cannot be used in clock asynchronous mode)

CBL1	CBL0	Character bit length control bits
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

SBL	Stop bit length control bit
0	1-bit length
1	2-bit length

TDP	Parity polarity bit
0	Even parity
1	Odd parity

PEN	Parity control bit
0	No parity
1	With parity

BDS	Serial data direction control bit
0	Transmit/receive data from LSB side sequentially
1	Transmit/receive data from MSB side sequentially

R/W : Readable/writable (Read value is the same as write value)
(shaded) : Initial value

Table 20.5-1 Functional Description of Each Bit of UART/SIO Serial Mode Control Register 1 (SMC10)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>BDS:
Serial data direction control bit</td><td>This bit sets the serial data direction (endian).
Setting the bit to "0": the bit specifies transmission or reception to be performed sequentially starting from the LSB side in the serial data register.
Setting the bit to "1": the bit specifies transmission or reception to be performed sequentially starting from the MSB side in the serial data register.</td></tr>
<tr><td>bit6</td><td>PEN:
Parity control bit</td><td>This bit enables or disables parity in clock asynchronous mode.
Setting the bit to "0": no parity
Setting the bit to "1": with parity</td></tr>
<tr><td>bit5</td><td>TDP:
Parity polarity bit</td><td>This bit controls even/odd parity.
Setting the bit to"0": specifies even parity
Setting the bit to"1": specifies odd parity</td></tr>
<tr><td>bit4</td><td>SBL:
Stop bit length control bit</td><td>This bit controls the length of the stop bit in clock asynchronous mode.
Setting the bit to "0": sets the stop bit length to "1".
Setting the bit to "1": sets the stop bit length to "2".
Note: The setting of this bit is only valid for transmission operation in asynchronous mode.For receiving operation, reception data register full flag is set to "1" after detecting stop bit (1-bit) and completing the reception regardless of this bit.</td></tr>
<tr><td>bit3,
bit2</td><td>CBL1, CBL0:
Character bit length control bit</td><td>These bits select the character bit length as shown in the following table:
<table><tr><th>CBL1</th><th>CBL0</th><th>Character bit length</th></tr><tr><td>0</td><td>0</td><td>5</td></tr><tr><td>0</td><td>1</td><td>6</td></tr><tr><td>1</td><td>0</td><td>7</td></tr><tr><td>1</td><td>1</td><td>8</td></tr></table>The above setting is valid in both asynchronous and synchronous modes.</td></tr>
<tr><td>bit1</td><td>CKS:
Clock selection bit</td><td>This bit selects the external clock or dedicated baud rate generator.
Setting the bit to "0": selects the dedicated baud rate generator.
Setting the bit to "1": selects the external clock.
Note: Setting this bit to "1" forcibly disables the output of the UCK0 pin.
The external clock cannot be used in clock asynchronous mode (UART).</td></tr>
<tr><td>bit0</td><td>MD:
Operation mode selection bit</td><td>This bit selects clock asynchronous mode (UART) or clock synchronous mode (SIO).
Setting the bit to "0": selects clock asynchronous mode (UART).
Setting the bit to "1": selects clock synchronous mode (SIO).</td></tr>
</table>

Note:

When modifying the UART/SIO serial mode control register 1 (SMC10), do not perform the modification during data transmission or reception.

20.5.2 UART/SIO Serial Mode Control Register 2 (SMC20)

UART/SIO serial mode control register 2 (SMC20) controls the UART/SIO operation mode. The register is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

■ UART/SIO Serial Mode Control Register 2 (SMC20)

Figure 20.5-3 UART/SIO Serial Mode Control Register 2 (SMC20)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0057H	SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE	00100000B
	R/W	R/W	R1/W	R/W	R/W	R/W	R/W	R/W	

TEIE	Transmission data register empty interrupt enable bit
0	Disables transmission data register empty interrupt
1	Enables transmission data register empty interrupt

TCIE	Transmission completion interrupt enable bit
0	Disables transmission completion interrupt
1	Enables transmission completion interrupt

RIE	Reception interrupt enable bit
0	Disables reception interrupt
1	Enables reception interrupt

TXE	Transmission operation enable bit
0	Disables transmission operation
1	Enables transmission operation

RXE	Reception operation enable bit
0	Disables reception operation
1	Enables reception operation

RERC	Reception error flag clear bit
0	Clears each error flag
1	No effect on the operation

TXOE	Serial data output enable bit
0	Disables serial data output (usable as port)
1	Enables serial data output

SCKE	Serial clock output enable bit
0	Disables serial clock output (usable as port)
1	Enables serial clock output

R/W : Readable/writable (Read value is the same as write value)
R1/W : Readable/writable (Read value is always "1")
(shaded) : Initial value

Table 20.5-2 Functional Description of Each Bit of UART/SIO Serial Mode Control Register 2 (SMC20)

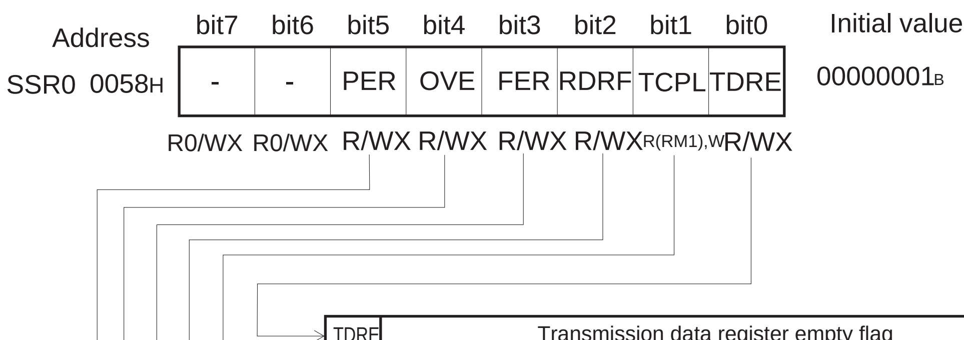
Bit name		Function
bit7	SCKE: Serial clock output enable bit	This bit controls the input/output of the serial clock (UCK0) pin in clock synchronous mode. **Setting the bit to "0":** allows the pin to be used as a general-purpose port. **Setting the bit to "1":** enables clock output. Note: When CKS is 1, the internal clock signal is not outputted even with this bit set to "1". If this bit is set to "1" with SMC10:MD set to "0" (asynchronous mode), the output from the port will always be "H".
bit6	TXOE: Serial data output enable bit	This bit controls the output of the serial data (UO0 pin). **Setting the bit to "0":** allows the pin to be used as a general-purpose port. **Setting the bit to "1":** enables serial data output.
bit5	RERC: Receive error flag clear bit	**Setting the bit to "0":** clears the error flags (PER, OVE, FER) of the SSR0 register. **Setting the bit to "1":** has no effect on operation. Reading this bit always returns "1".
bit4	RXE: Reception operation enable bit	**Setting the bit to "0":** disables the reception of serial data. **Setting the bit to "1":** enables the reception of serial data. If this bit is set to "0" during reception, the reception operation will be immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the serial input data register. Note: Setting this bit to "0" initializes reception operation.It has no effect on the receive data register full (RDRF) bit or an error flag (PER, OVE, FER).
bit3	TXE: Transmission operation enable bit	**Setting the bit to "0":** disables the transmission of serial data. **Setting the bit to "1":** enables the transmission of serial data. If this bit is set to "0" during transmission, the transmission operation will be immediately disabled and initialization will be performed.The transmission completion flag (TCPL) will be set to "1" and the transmission data register empty (TDRE) bit will also be set to "1".
bit2	RIE: Reception interrupt enable bit	**Setting the bit to "0"**: disables reception interrupt. **Setting the bit to "1":** enables reception interrupt. A reception interrupt occurs immediately after either the receive data register full (RDRF) bit or an error flag (PER, OVE, FER) is set to "1" with this bit set to "1" (enabled).
bit1	TCIE: Transmission completion interrupt enable bit	**Setting the bit to "0":** disables interrupts by the transmission completion flag. **Setting the bit to "1":** enables interrupts by the transmission completion flag. A transmission interrupt occurs immediately after the transmission completion flag (TCPL) bit is set to "1" with this bit set to "1" (enabled).
bit0	TEIE: Transmission data register empty interrupt enable bit	**Setting the bit to "0":** disables interrupts by the transmission data register empty. **Setting the bit to "1":** enables interrupts by the transmission data register empty. A transmission interrupt occurs immediately after the transmission data register empty (TDRE) bit is set to "1" with this bit set to "1" (enabled).

20.5.3 UART/SIO Serial Status and Data Register (SSR0)

The UART/SIO serial status and data register (SSR0) indicates the transmission/reception status and error status of the UART/SIO.

■ UART/SIO Serial Status and Data Register (SSR0)

Figure 20.5-4 UART/SIO Serial Status and Data Register (SSR0)



TDRE	Transmission data register empty flag
0	Transmit data present
1	Transmit data absent

TCPL	Transmission completion flag
0	Cleared by writing "0"
1	Serial transmission complete

RDRF	Reception data register full flag
0	Receive data absent
1	Receive data present

FER	Framing error flag
0	Framing error absent
1	Framing error present

OVE	Overrun error flag
0	Overrun error absent
1	Overrun error present

PER	Parity error flag
0	Parity error absent
1	Parity error present

R(RM1), W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no ef fect on operation)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
(shaded) : Initial value

Table 20.5-3 Functional Description of Each Bit of UART/SIO Serial Status and Data Register (SSR0)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. • Reading always returns "0". • Writing to the bits has no effect on operation.
bit5	PER: Parity error flag	Detect a parity error in received data. • The flag is set when a parity error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit4	OVE: Overrun error flag	Detect an overrun error in received data. • The flag is set when an overrun error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit3	FER: Framing error flag	Detect a framing error in received data. • The bit is set when a framing error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit2	RDRF: Receive data register full flag	This flag indicates the status of the UART/SIO serial input data register. • The bit is set to "1" when receive data is loaded to the serial input data register. • The bit is cleared to "0" when data is read from the serial input data register.
bit1	TCPL: Transmission completion flag	This flag indicates the data transmission status. • The bit is set to "1" upon completion of serial transmission. Note, however, that the bit is not set to "1" even upon completion of transmission when the serial output data register contains data to be transmitted in succession. • Writing "0" to this bit clears its flag. • If events to set and clear the flag occur at the same time, it is set preferentially. • Writing "1" to this bit has no effect on operation.
bit0	TDRE: Transmission data register empty flag	This flag indicates the status of the UART/SIO serial output data register. • The bit is set to "0" when transmit data is written to the serial output register. • The bit is set to "1" when data is loaded to the transmission shift register and transmission starts.

20.5.4 UART/SIO Serial Input Data Register (RDR0)

The UART/SIO serial input data register (RDR0) is used to input (receive) serial data.

■ UART/SIO Serial Input Data Register (RDR0)

Figure 20.5-5 shows the bit configuration of the UART/SIO serial input data register (RDR0).

Figure 20.5-5 UART/SIO Serial Input Data Register (RDR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RDR0	$005A_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX: Read only (Readable, writing has no effect on operation)

This register stores received data.The serial data signals sent to the serial data input pin (UI0 pin) is converted by the shift register and stored in this register.

When received data is set correctly in this register, the receive data register full (RDRF) bit is set to "1". At this time, an interrupt occurs if reception interrupt requests have been enabled. If an RDRF bit check by the program or using an interruption shows that received data is stored in this register, the reading of the content for this register clears the RDRF flag to "0".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are set to "0".

20.5.5 UART/SIO Serial Output Data Register (TDR0)

The UART/SIO serial output data register (TDR0) is used to output (transmit) serial data.

■ UART/SIO Serial Output Data Register (TDR0)

Figure 20.5-6 shows the bit configuration of the UART/SIO serial output data register (TDR0).

Figure 20.5-6 UART/SIO Serial Output Data Register (TDR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TDR0	0059_H	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

This register holds data to be transmitted. The register accepts a write when the transmission data register empty (TDRE) bit contains "1". An attempt to write to the bit is ignored when the bit contains "0".

When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of serial mode control register 2 is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible. Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR0, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". The transmission data is transferred to the shift register for the transmission, it is converted into the serial data, and it is transmitted from the serial data output pin.

When transmit data is written to the UART/SIO serial output data register (TDR0), the transmission data register empty bit (TDRE) is set to "0". Upon completion of transfer of transmit data to the transmission shift register, the transmission data register empty bit (TDRE) is set to "1", allowing the next piece of transmit data to be written. At this time, an interrupt occurs if transmission data register empty interrupts have been enabled. Write the next piece of transmit data when transmit data register empty occurs or the transmit data register empty (TDRE) bit is set to "1".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are ignored.

Note:

The data in this register cannot be updated when TDRE in UART/SIO serial status and data register is "0".
When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of serial mode control register 2 is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible. Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR0, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". And, to change data, please write it after making TDRE "1" once by writing TXE =0.

20.6 Interrupts of UART/SIO

The UART/SIO has six interrupt-related bits: error flag bits (PER, OVE, FER), receive data register full bit (RDRF), transmission data register empty bit (TDRE), and transmission completion flag (TCPL).

■ Interrupts of UART/SIO

Table 20.6-1 lists the UART/SIO interrupt control bits and interrupt sources.

Table 20.6-1 UART/SIO Interrupt Control Bits and Interrupt Sources

Item	Description					
Interrupt request flag bit	SSR0: TDRE	SSR0: TCPL	SSR0: RDRE	SSR0: PER	SSR0: OVE	SSR0: FER
Interrupt request enable bit	SMC20: TEIE	SMC20: TCIE	SMC20: RIE	SMC20: RIE	SMC20: RIE	SMC20: RIE
Interrupt source	Transmission data register empty	Transmission completion	Reception data register full	Parity error	Overrun error	Framing error

■ Transmit Interrupts

When transmit data is written to the serial output data register (TDR0), the data is transferred to the transmission shift register. When the next piece of data can be written, the TDRE bit is set to "1".At this time, an interrupt request to the interrupt controller occurs when transmit data register empty interrupt enable bit has been enabled (SMC20:TEIE = 1). The TCPL bit is set to "1" upon completion of transmission of all pieces of transmit data.At this time, an interrupt request to the interrupt controller occurs when transmission completion interrupt enable bit has been enabled (SMC20:TCIE = 1).

■ Reception Interrupt

If the data is inputted successfully up to the stop bit, the RDRF bit is set to 1. If an overrun, parity, or framing error occurs, the corresponding error flag bit (PER, OVE, or FER) is set to "1".

These bits are set when a stop bit is detected. If reception interrupt enable bit has been enabled (SMC20:RIE = 1), an interrupt request to the interrupt controller will be generated.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

■ Registers and Vector Table Related to UART/SIO Interrupts

Table 20.6-2 Registers and Vector Table Related to UART/SIO Interrupts

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
ch.0	IRQ4	ILR1	L04	$FFF2_H$	$FFF3_H$

20.7 Explanation of UART/SIO Operations and Setup Procedure Example

The UART/SIO has a serial communication function (operation modes 0, 1).

■ Operation of UART/SIO

● Operation mode

Two operation modes are available in the UART/SIO.Clock synchronous mode (SIO) or clock asynchronous mode (UART) can be selected (see Table 20.7-1).

Table 20.7-1 Operation Modes of UART/SIO

Operation mode	Data length		Synchronous Mode	Stop bit length
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	–	Synchronous	–
	6	–		
	7	–		
	8	–		

■ Setup Procedure Example

The UART/SIO is set up in the following procedure.

● Initial setting

1) Set the port for input. (DDR1)
2) Set the interrupt level. (ILR1)
3) Set the prescaler. (PSSR0)
4) Set the baud rate. (BRSR0)
5) Select the clock. (SMC10:CKS)
6) Set the operation mode. (SMC10:MD)
7) Enable/disable the serial clock output. (SMC20:SCKE)
8) Enable reception. (SMC20:RXE = 1)
9) Enable interrupts. (SMC20:RIE = 1)

● Interrupt processing

Read receive data. (RDR0)

20.7.1 Operating Description of Operation Mode 0

Operation mode 0 operates as clock asynchronous mode (UART).

■ Operating Description of UART/SIO Operation Mode 0

Clock asynchronous mode (UART) is selected when the MD bit in the UART/SIO serial mode control register 1 (SMC10) is set to "0".

● Baud rate

The serial clock is selected by the CKS bit in the SMC10 register. Be sure to select the dedicated baud rate generator at this time.

The baud rate is equivalent to the output clock frequency of the dedicated baud rate generator, divided by four. The UART can perform communication within the range from -2% to +2% of the selected baud rate.

The baud rate generated by the dedicated baud rate generator is obtained from the equation illustrated below. (For information about the dedicated baud rate generator, refer to "CHAPTER 21 UART/SIO DEDICATED BAUD RATE GENERATOR".

Figure 20.7-1 Baud Rate Calculation when Using Dedicated Baud Rate Generator

$$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{4 \times \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{bmatrix} \times \begin{bmatrix} 2 \\ : \\ 255 \end{bmatrix}} \text{ [bps]}$$

UART prescaler selection register (PSSR0)
Prescaler selection
(PSS1, PSS0)

UART baud rate setting register (BRSR0)
Baud rate setting
(BRS7 to BRS0)

Table 20.7-2 Sample Asynchronous Transfer Rates Based on Dedicated Baud Rate Generator (Machine Clock = 10MHz, 16MHz, 16.25MHz)

Dedicated baud rate generator setting		UART Internal division	Total division ratio (PSS × BRS × 4)	Baud rate (10MHz/ Total division ratio)	Baud rate (16MHz/ Total division ratio)	Baud rate (16.25MHz/ Total division ratio)
Prescaler selection PSS[1:0]	Baud rate counter setting BRS[7:0]					
1 (Setting value:0, 0)	20	4	80	125000	200000	203125
1 (Setting value:0, 0)	22	4	88	113636	181818	184659
1 (Setting value:0, 0)	44	4	176	56818	90909	92330
1 (Setting value:0, 0)	87	4	348	28736	45977	46695
1 (Setting value:0, 0)	130	4	520	19231	30769	31250
2 (Setting value:0, 1)	130	4	1040	9615	15385	15625
4 (Setting value:1, 0)	130	4	2080	4808	7692	7813
8 (Setting value:1, 1)	130	4	4160	2404	3846	3906

The baud rate in clock asynchronous mode can be set in the following range.

Table 20.7-3 Baud Rate Setting Range in Clock Asynchronous Mode

PSS[1:0]	BRS[7:0]
00_B to 11_B	02_H (2) to FF_H (255)

● Transfer data format

UART can treat data only in NRZ (Non-Return-to-Zero) format. Figure 20.7-2 shows the transfer data format.

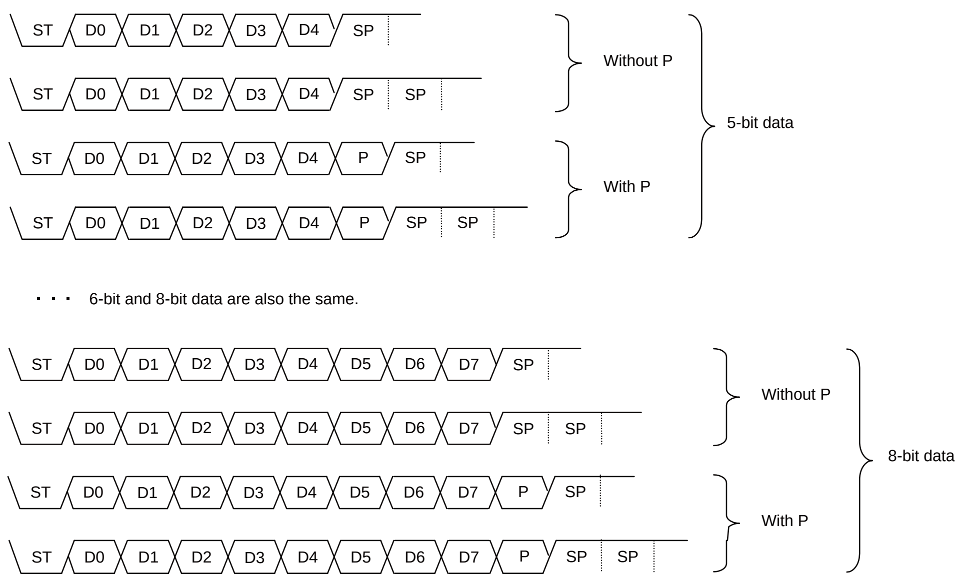
The character bit length can be selected from among 5 to 8 bits depending on the CBL1 and CBL0 settings.

The stop bit length can be set to 1 or 2 bits depending on the SBL setting.

PEN and TDP can be used to enable/disable parity and to select parity polarity.

As is shown in Figure 20.7-2, the transfer data always starts from the start bit ("L" level) and ends with the stop bit ("H" level) by performing the specified data bit length transfer with MSB first or LSB first ("LSB first" or "MSB first" can be selected by the BDS bit).It becomes "H" level at the idle state.

Figure 20.7-2 Transfer Data Format



ST : Start bit
SP : Stop bit
P : Parity bit
D0 to D7 : Data. The sequence can be selected from "LSB first" or "MSB first" by the direction control register (BDS bit)

● Receiving operation in asynchronous clock mode (UART)

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

Reception remains performed as long as the reception operation enable bit (RXE) contains "1".

Upon detection of a start bit in receive data with the reception operation enable bit (RXE) set to "1", one frame of data is received according to the data format set in UART/SIO serial control register 1 (SMC10).

When the reception of one frame of data has been completed, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next frame of serial data can be received.

When the UART/SIO serial input data register (RDR0) stores data, the receive data register full (RDRF) bit is set to "1".

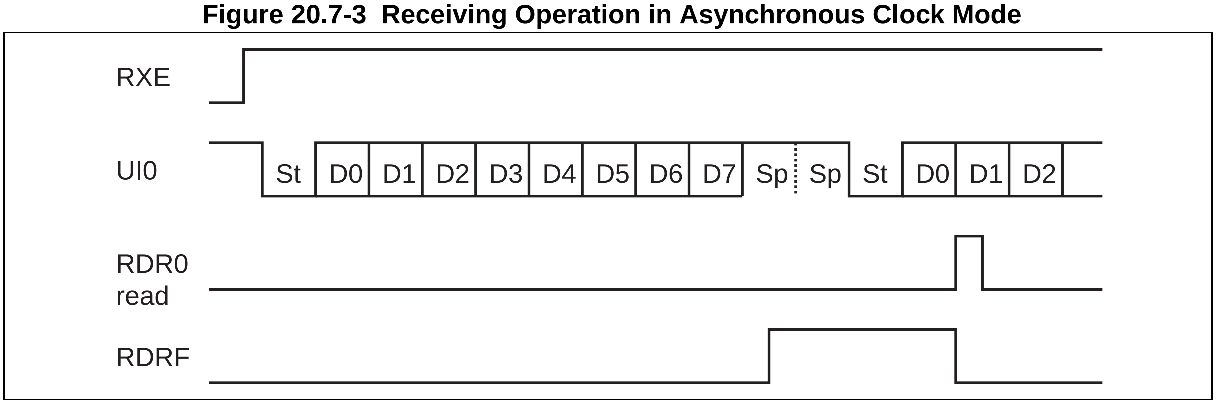
A reception interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the reception interrupt enable bit (RIE) contains "1".

Received data is read from the UART/SIO serial input data register (RDR0) after each error flag (PER, OVE, FER) in the UART/SIO serial status and data register is checked.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

Note that modifying UART/SIO serial mode control register 1 (SMC10) during reception may result in unpredictable operation. If the RXE bit is set to "0" during reception, the reception is immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the serial input data register.

Figure 20.7-3 Receiving Operation in Asynchronous Clock Mode



● Reception error in asynchronous clock mode (UART)

If any of the following three error flags (PER, FER, OVE) has been set, receive data is not transferred to the UART/SIO serial input data register (RDR0) and the receive data register full (RDRF) bit is not set to "1" either.

- Parity error (PER)

  The parity error (PER) bit is set to "1" if the parity bit in received serial data does not match the parity polarity bit (TDP) when the parity control bit (PEN) contains "1".

- Framing error (FER)

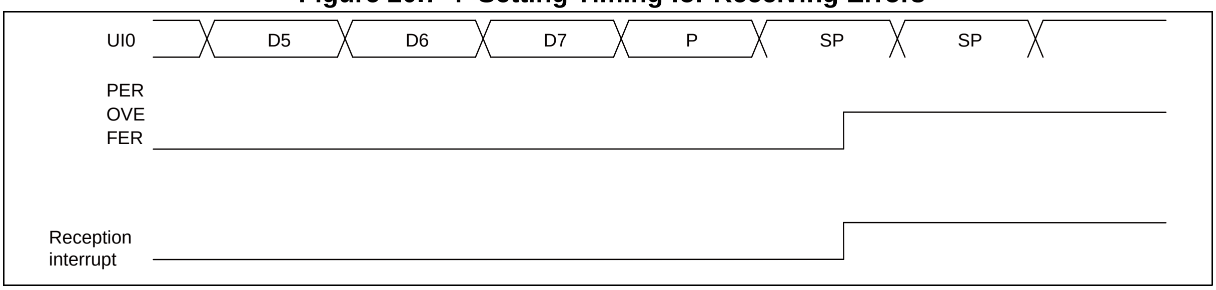
  The framing error (FER) bit is set to "1" if "1" is not detected at the position of the first stop bit in serial data received in the set character bit length (CBL) under parity control (PEN). Note that the stop bit is not checked if it appears at the second bit or later.

- Overrun error (OVE)

  Upon completion of reception of serial data, the overrun error (OVE) bit is set to "1" if the reception of the next data is performed before the previous receive data is read.

Each flag is set at the position of the first stop bit.

Figure 20.7-4 Setting Timing for Receiving Errors



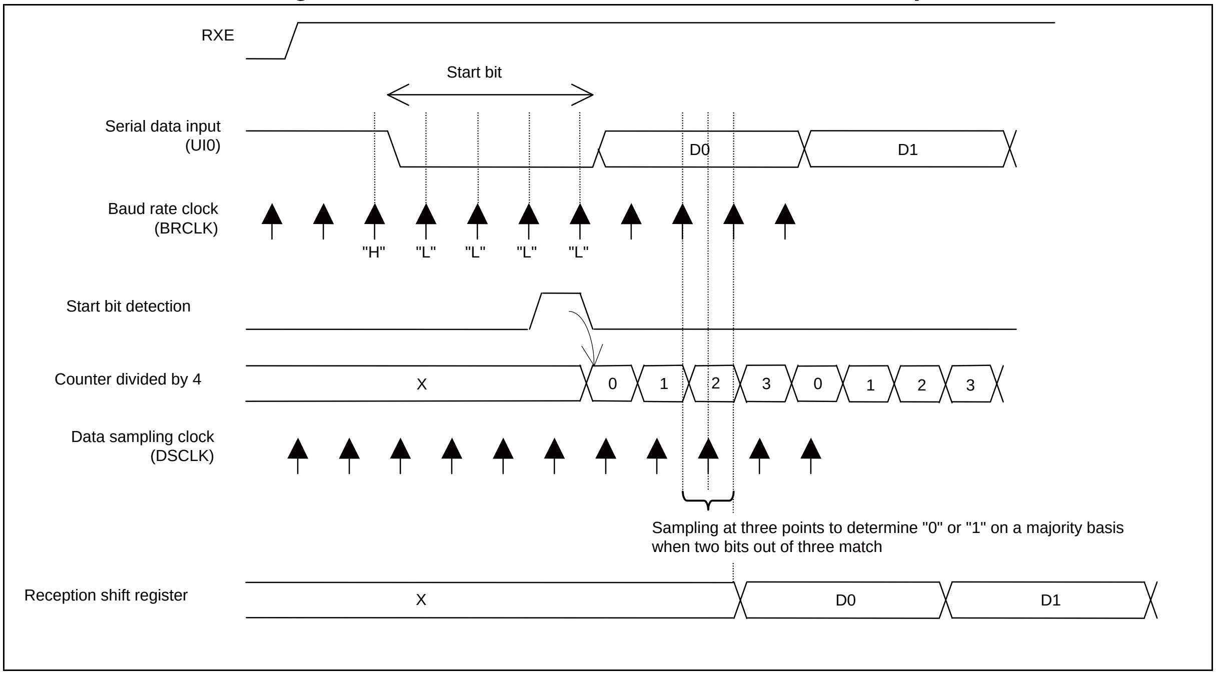
● Start bit detection and confirmation of receive data during reception

The start bit is detected by a falling of the serial input followed by a succession of three "L" levels after the serial data input is sampled according to the clock (BRCLK) signal provided by the dedicated baud rate generator with the reception operation enable bit (RXE) set to "1". When the first "H", "L", "L", "L" train is detected in a BRCLK sample, therefore, the current bit is regarded as the start bit.

The frequency-quartered circuit is activated upon detection of the start bit and serial data is inputted to the reception shift register at intervals of four periods of BRCLK.

When data is received, sampling is performed at three points of the baud rate clock (BRCLK) and data sampling clock (DSCLK) and received data is confirmed on a majority basis when two bits out of three match.

Figure 20.7-5 Start Bit Detection and Serial Data Input



● Transmission in asynchronous clock mode

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

The following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", and then write transmit data to the serial output data register to start transmission.
- Write transmit data to the serial output data register, and then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit is set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

The transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, and the transmit data register empty (TDRE) is set to "1".

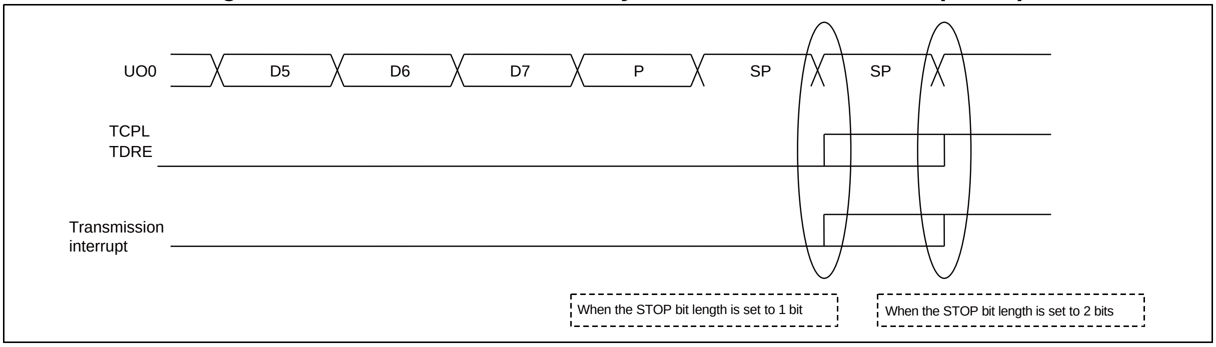
When the transmission interrupt enable bit (TIE) contains "1", a transmission interrupt occurs if the transmit data register empty (TDRE) bit is set to "1". This allows the next piece of transmit data to be written to the UART/SIO serial output data register (TDR0) by interrupt handling.

To detect the completion of serial transmission by transmission interrupt, set the transmission completion interrupt enable bits as follows: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmission interrupt occurs.

Both the transmission completion flag (TCPL) and the transmission data register empty flag (TDRE), when transmitting data consecutively, are set at the position which the transmission of the last bit was completed (it varies depending on the data length, parity enable, or stop bit length setting), as shown in Figure 20.7-6 below.

Note that modifying UART/SIO serial mode control register 1 (SMC10) during transmission may result in unpredictable operation.

Figure 20.7-6 Transmission in Asynchronous Clock Mode (UART)



The TDRE flag is set at the point indicated in the following figure if the preceding piece of transmit data does not exist in the transmission shift register.

Figure 20.7-7 Setting Timing 1 for Transmit Data Register Empty Flag (TDRE) (When TXE is "1")

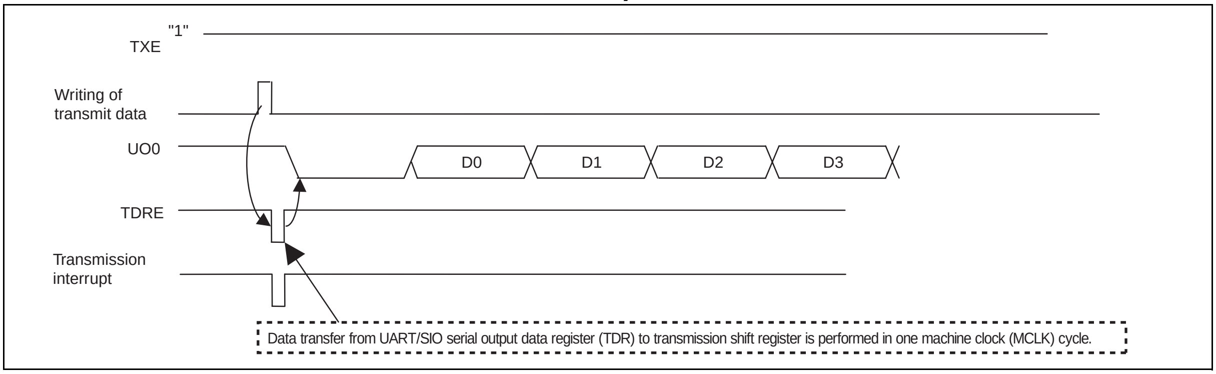
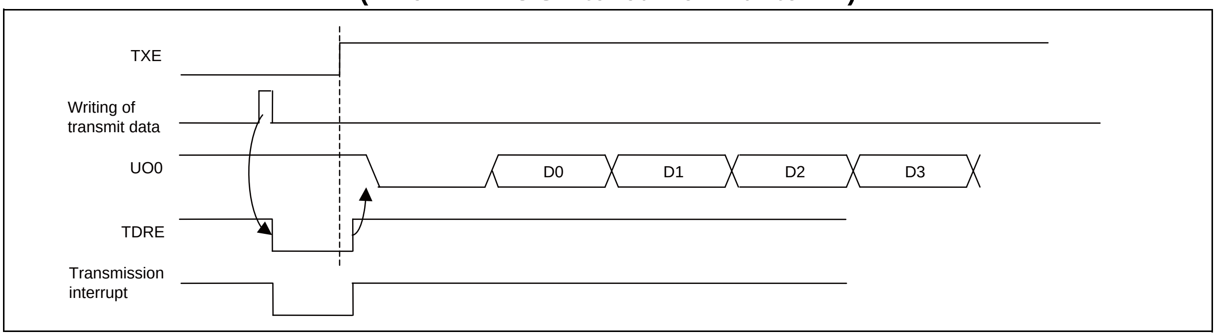


Figure 20.7-8 Setting Timing 2 for Transmit Data Register Empty Flag (TDRE) (When TXE Is Switched from "0" to "1")



- Concurrent transmission and reception

In asynchronous clock mode (UART), transmission and reception can be performed independently. Therefore, transmission and reception can be performed at the same time or even with transmitting and receiving frames overlapping each other in shifted phases.

20.7.2 Operating Description of Operation Mode 1

Operation mode 1 operates in synchronous clock mode.

■ Operating Description of UART/SIO Operation Mode 1

Setting the MD bit in UART/SIO serial mode control register 1 (SMC10) to "1" selects synchronous clock mode (SIO).

The character bit length in synchronous clock mode (SIO) is variable between 5 bits and 8 bits. Note, however, that parity is disabled and no stop bit is used.

The serial clock is selected by the CKS bit in the SMC10 register. Select the dedicated baud rate generator or external clock. The SIO performs shift operation using the selected serial clock as a shift clock.

To input the external clock signal, set the SCKE bit to "0".

To output the dedicated baud rate generator output as a shift clock signal, set the SCKE bit to "1". The serial clock signal is obtained by dividing clock by two, which is supplied by the dedicated baud rate generator. The baud rate in the SIO mode can be set in the following range. (For more information about the dedicated baud rate generator, also refer to "CHAPTER 21 UART/SIO DEDICATED BAUD RATE GENERATOR").

Table 20.7-4 Baud Rate Setting Range in SIO Mode

PSS[1:0]	BRS[7:0]
00_B to 11_B	01_H(1) to FF_H(255), 00_H(256) (The highest and lowest baud rate settings are 01_H and 00_H, respectively.)

The baud rate applied when the external clock or dedicated baud rate generator is used is obtained from the corresponding equation illustrated below. (Figure 20.7-9, Figure 20.7-10)

Figure 20.7-9 Calculating Baud Rate Based on External Clock

$$\text{Baud rate} = \frac{1}{\text{External Clock*}} \text{ [bps]}$$

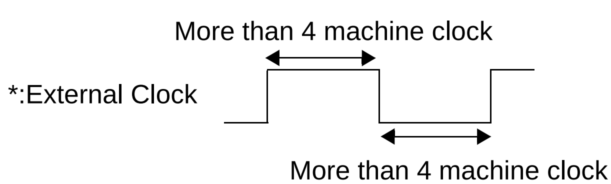


Figure 20.7-10 Baud Rate Calculation Formula for Using Dedicated Baud Rate Generator

$$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{2 \times \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{bmatrix} \times \begin{bmatrix} 1 \\ : \\ 256 \end{bmatrix}} \text{ [bps]}$$

UART prescaler selection register (PSSR0)
Prescaler selection
(PSS1, PSS0)

UART baud rate setting register (BRSR0)
Baud Rate Setting
(BRS7 to BRS0)

● Serial clock

The serial clock signal is outputted under control of the output for transmit data. When only reception is performed, therefore, set transmission control (TXE = 1) to write dummy transmit data to the UART/SIO serial output register.Refer to the data sheet for the UCK0 clock value.

● Reception in UART/SIO operation mode 1

For reception in operation mode 1, each register is used as follows.

Figure 20.7-11 Registers Used for Reception in Operation Mode 1

SMC10 (UART/SIO Serial Mode Control Register 1)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
◎	×	×	×	◎	◎	◎	1

SMC20 (UART/SIO Serial Mode Control Register 2)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
◎	0	◎	◎	◎	◎	×	×

SSR0 (UART/SIO serial status and data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
–	–	PER	OVE	FER	RDRF	TCPL	TDRE
×	×	×	◎	×	◎	×	×

TDR0 (UART/SIO serial output data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
×	×	×	×	×	×	×	×

RDR0 (UART/SIO serial input data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
◎	◎	◎	◎	◎	◎	◎	◎

◎: Used bit
×: Unused bit
0: Set "0"
1:Set "1"

The reception depends on whether the serial clock has been set to external or internal clock.

<When external clock is enabled>

When the reception operation enable bit (RXE) contains "1", serial data is received always at the rising edge of the external clock signal.

<When internal clock is enabled>

The serial clock signal is outputted in accordance with transmission. Therefore, transmission must be performed even when only performing reception. The following two procedures can be used.

- Set the transmission operation enable bit (TXE) to "1", then write transmit data to the UART/SIO serial output data register to generate the serial clock signal and start reception.
- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to generate the serial clock signal and start reception.

When 5-bit to 8-bit serial data is received by the reception shift register, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next piece of serial data can be received.

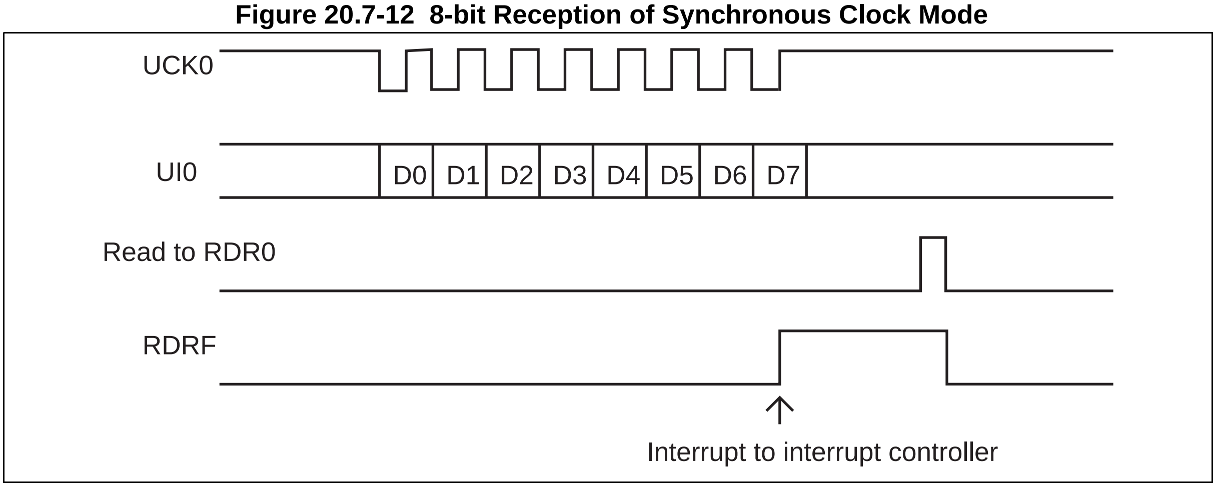
When the UART/SIO serial input data register stores data, the receive data register full (RDRF) bit is set to "1".

A reception interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the reception interrupt enable bit (RIE) contains "1".

To read received data, read it from the UART/SIO serial input data register after checking the error flag (OVE) in the UART/SIO serial status and data register.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

Figure 20.7-12 8-bit Reception of Synchronous Clock Mode



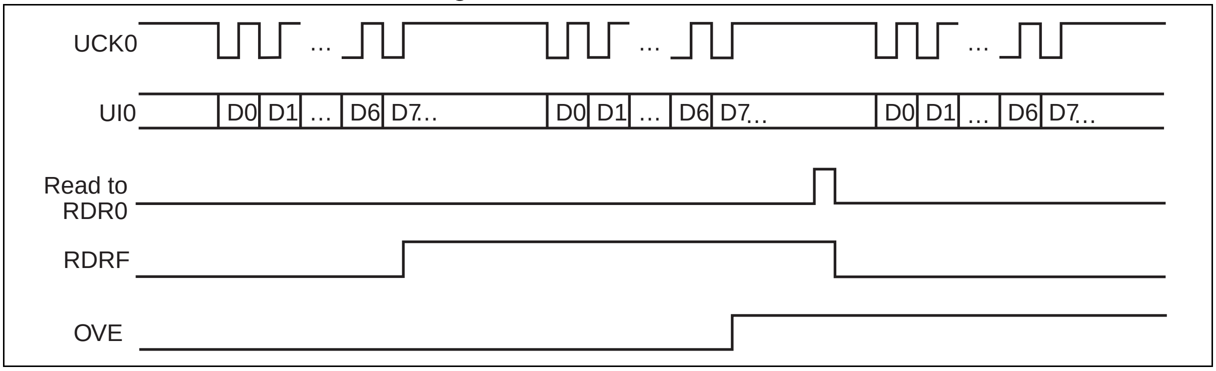
Operation when reception error occurs

When an overrun error (OVE) exists, received data is not transferred to the UART/SIO serial input data register (RDR0).

Overrun error (OVE)

Upon completion of reception for serial data, the overrun error (OVE) bit is set to "1" if the receive data register full (RDRF) bit has been set to "1" by the reception for the preceding piece of data.

Figure 20.7-13 Overrun error



● Transmission in UART/SIO operation mode 1

For transmission in operation mode 1, each register is used as follows.

Figure 20.7-14 Registers Used for Transmission in Operation Mode 1

SMC10 (UART/SIO Serial Mode Control Register 1)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
◎	×	×	×	◎	◎	◎	1

SMC20 (UART/SIO Serial Mode Control Register 2)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
◎	0	◎	◎	◎	◎	×	×

SSR0 (UART/SIO Serial Status and Data Register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
–	–	PER	OVE	FER	RDRF	TCPL	TDRE
×	×	×	◎	×	◎	×	×

TDR0 (UART/SIO Serial Output Data Register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
×	×	×	×	×	×	×	×

RDR0 (UART/SIO Serial Input Data Register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
◎	◎	◎	◎	◎	◎	◎	◎

◎: Used bit
×: Unused bit
0 : Set "0"
1 :Set "1"

The following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", and then write transmit data to the UART/SIO serial output data register to start transmission.

- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit is set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

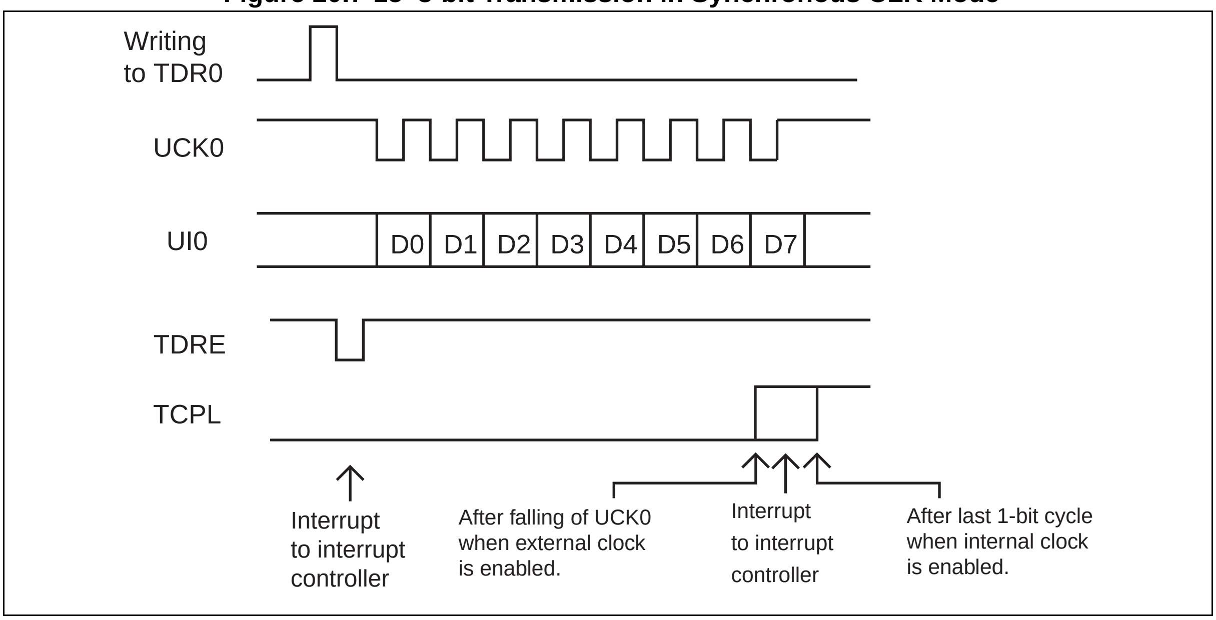
When serial transmission is started after transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, and the transmit data register empty (TDRE) is set to "1".

When the use of the external clock signal has been set, serial data transmission starts at the fall of the first serial clock signal after the transmission process is started.

A transmission completion interrupt occurs the moment the transmit data register empty (TDRE) bit is set to "1" when the transmission interrupt enable bit (TIE) contains "1". At this time, the next piece of transmit data can be written to the UART/SIO serial output data register (TDR0). Serial transmission can be continued with the transmission operation enable bit (TXE) set to "1".

To use a transmission completion interrupt to detect the completion of serial transmission, enable transmission completion interrupt output this way: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmission completion interrupt occurs.

Figure 20.7-15 8-bit Transmission in Synchronous CLK Mode



● Concurrent transmission and reception

<When external clock is enabled>

Transmission and reception can be performed independently of each other. Transmission and reception can therefore be performed at the same time or even when their phases are shifted from each other and overlapping.

<When internal clock is enabled>

As the transmitting side generates a serial clock, reception is influenced.

If transmission stops during reception, the receiving side is suspended. It resumes reception when the transmitting side is restarted.

- Refer to "20.4 Pins of UART/SIO" for operation with serial clock output and operation with serial clock input.

20.8 Sample Programs for UART/SIO

We provide sample programs that can be used to operate UART/SIO.

■ Sample Programs for UART/SIO

For information about the sample programs for UART/SIO, refer to "■ Sample Programs" in Preface.

■ Setting Methods not Covered by Sample Programs

● How to select the operation mode

The operation mode select bit (SMC10.MD) is used.

Operation mode		Operation mode selection (MD)
Mode 0	Asynchronous clock mode (UART)	Set the bit to "0"
Mode 1	Synchronous clock mode (SIO)	Set the bit to "1"

● Operation clock types and how to select it

The clock select bit (SMC10.CKS) is used.

Clock input	Clock selection (CKS)
To select a dedicated baud rate generator	Set the bit to "0"
To select an external clock	Set the bit to "1"

● How to use UCK0, UI0, and UO0 pin

Uses the following setting.

	UART
To set the UCK0 pin as input	DDR1.P12 = 0 SMC20:SCKE = 0
To set the UCK0 pin as output	SMC20:SCKE = 1
When using UI0 pin	DDR1.P10 = 0
When using UO0 pin	SMC20:TXOE = 1

● How to enable/stop UART operation

The reception operation enable bit (SMC20.RXE) is used.

Control item	Reception interrupt enable bit (RXE)
Disabling (stopping) reception	Set the bit to "0".
Enabling reception	Set the bit to "1".

The transmission operation control bit (SMC20.TXE) is used.

Control item	Transmission operation enable bit (TXE)
Disabling (stopping) transmission	Set the bit to "0".
Enabling transmission	Set the bit to "1".

● How to set the parity

The parity control (SMC10.PEN) and parity polarity (SMC10.TDP) bits are used.

Operation	Parity control (PEN)	Parity polarity (TDP)
To set to no parity	Set the bit to "0".	–
To set to even parity	Set the bit to "1".	Set the bit to "0".
To set to odd parity	Set the bit to "1".	Set the bit to "1".

● How to set the data length

The data length select bit (SMC10.CBL[1:0]) is used.

Operation	Data length select bit (CBL[1:0])
To set the bit length to 5	Set the bits to "00_B".
To set the bit length to 6	Set the bits to "01_B".
To set the bit length to 7	Set the bits to "10_B".
To set the bit length to 8	Set the bits to "11_B".

● How to select the STOP bit length

The STOP bit length control bit (SMC10.SBL) is used.

Operation	STOP bit length control (SBL)
To set STOP bit length to 1	Set the bit to "0".
To set STOP bit length to 2	Set the bit to "1".

● How to clear the error flag

The reception error flag clear bit (SMC20.RERC) is used.

Control item	Reception error flag clear bit (RERC)
When clearing error flags (PER, OVE, FER)	Set the bit to "0".

● How to set the transfer direction

The serial data direction control bit (SMC10.BDS) is used.

LSB first/MSB first can be selected for transfer direction in any operation mode.

Control item	Serial data direction control (BDS)
When selecting LSB first transfer (from least significant bit)	Set the bit to "0".
When selecting MSB first transfer (from most significant bit)	Set the bit to "1".

● How to clear the reception completion flag

Uses the following setting.

Control item	Method
To clear the reception completion flag	Read the RDR0 register

The first RDR0 register read is the reception initiation.

● How to clear the transmit buffer empty flag

Uses the following setting.

Control item	Method
To clear the transmit buffer empty flag	Write to TDR0 register

The first TDR0 register write is the transmit initiation.

● How to set the baud rate

See Section "20.7.1 Operating Description of Operation Mode 0".

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

Channel	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR1) Address: $0007A_H$	#4 Address: $0FFF2_H$

● Enabling, disabling, and clearing interrupts

The interrupt request enable bits (SMC20:RIE), (SMC20:TCIE), (SMC20:TEIE) are used to enable interrupts.

	UART reception	UART transmission	
	Reception interrupt enable bit (RIE)	Transmission completion interrupt enable bit (TCIE)	Transmission data register empty interrupt enable bit (TEIE)
To disable interrupt requests	Set to "0"		
To enable interrupt requests	Set to "1"		

The following setting is used to clear interrupt requests.

	UART reception	UART transmission
To clear interrupt requests	Read from serial input register (RDR 0) to clear reception data register full bit (RDRF).	The transmit data register empty (TDRE) is set to "0" by writing data to the serial output data register (TDR0).
	Write "0" to error flag clear bit (RERC) to clear error flags (PER, OVE, FER) to "0".	

CHAPTER 21
UART/SIO DEDICATED BAUD RATE GENERATOR

This chapter describes the functions and operations of the dedicated baud rate generator of UART/SIO.

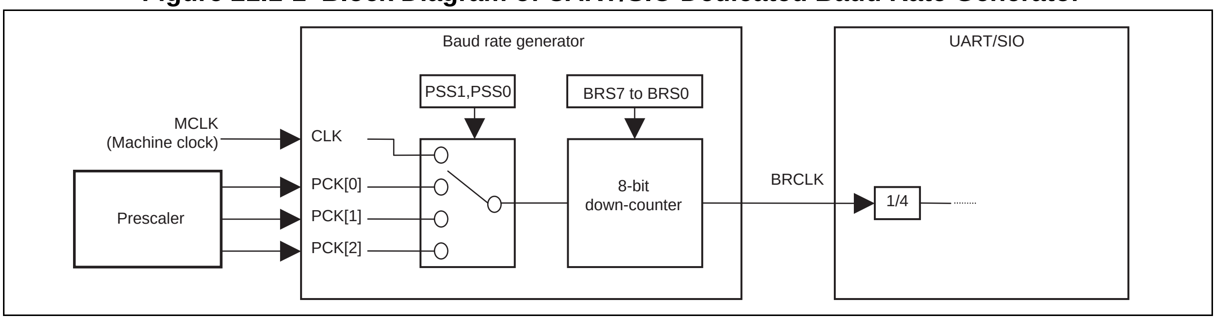
Code: CM26-00121-1E

21.1 Overview of UART/SIO Dedicated Baud Rate Generator

The UART/SIO dedicated baud rate generator generates the baud rate for the UART/SIO.
The generator consists of the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Block Diagram of UART/SIO Dedicated Baud Rate Generator

Figure 21.1-1 Block Diagram of UART/SIO Dedicated Baud Rate Generator



■ Input Clock

The UART/SIO dedicated baud rate generator uses the output clock from the prescaler or the machine clock as its input clock.

■ Output Clock

The UART/SIO dedicated baud rate generator supplies its clock to the UART/SIO.

21.2 Channels of UART/SIO Dedicated Baud Rate Generator

This section describes the channels of the UART/SIO dedicated baud rate generator.

■ Channels of UART/SIO Dedicated Baud Rate Generator

This series contains one channel of the UART/SIO dedicated baud rate generator.

Table 21.2-1 shows the registers of the UART/SIO dedicated baud rate generator.

Table 21.2-1 Registers of UART/SIO Dedicated Baud Rate Generator

Channel	Register name	Corresponding register (Representation in this manual)
0	PSSR0	UART/SIO dedicated baud rate generator prescaler selection register
	BRSR0	UART/SIO dedicated baud rate generator baud rate setting register

21.3 Registers of UART/SIO Dedicated Baud Rate Generator

The registers related to the UART/SIO dedicated baud rate generator are namely the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Registers Related to UART/SIO Dedicated Baud Rate Generator

Figure 21.3-1 Registers Related to UART/SIO Dedicated Baud Rate Generator

UART/SIO dedicated baud rate generator prescaler selection register (PSSR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PSSR0	$0FBE_H$	–	–	–	–	–	BRGE	PSS1	PSS0	00000000_B
		R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	

UART/SIO dedicated baud rate generator baud rate setting register (BRSR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
BRSR0	$0FBF_H$	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

21.3.1 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

The UART/SIO dedicated baud rate generator prescaler register (PSSR0) controls the output of the baud rate clock and the prescaler.

■ UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

Figure 21.3-2 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

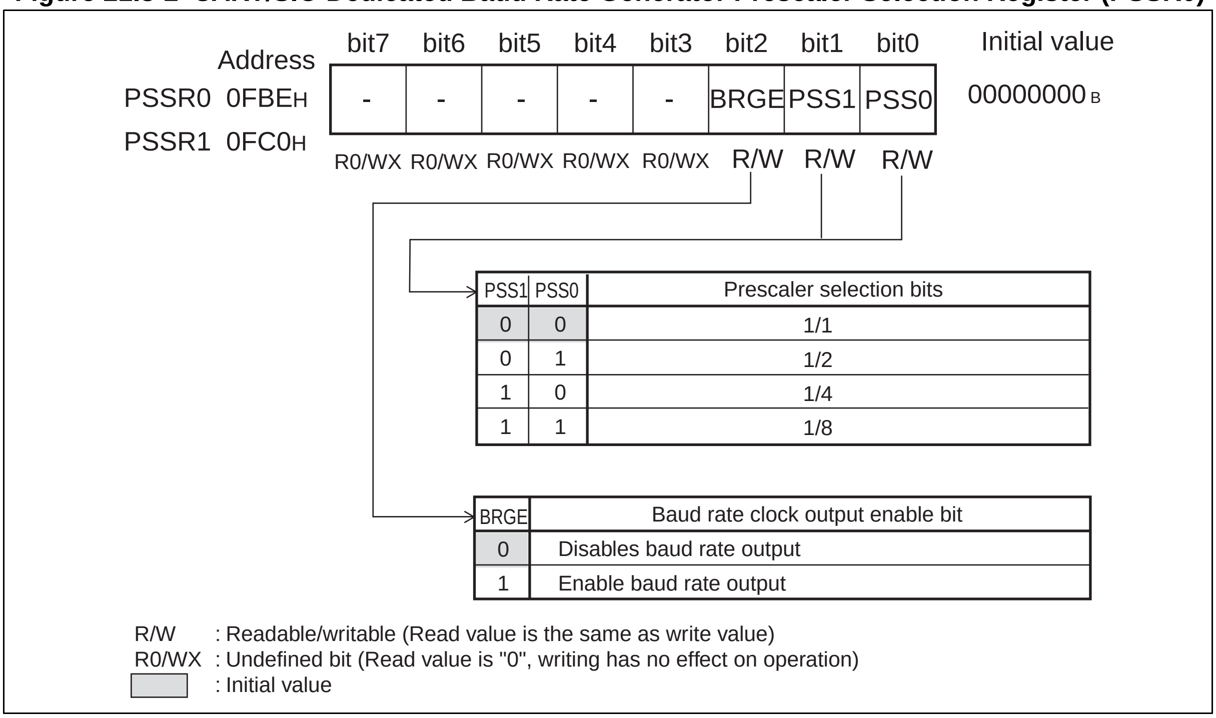


Table 21.3-1 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7 to bit3</td><td>Undefined bits</td><td>These bits are undefined.Reading the bits always returns "0".</td></tr>
<tr><td>bit2</td><td>BRGE:
Baud rate clock output enable bit</td><td>This bit enables the output of the baud rate clock "BRCLK".
When set to "1": loads BRS[7:0] to the 8-bit down-counter and outputs "BRCLK", which is supplied to the UART/SIO.
When set to "0": stops the output of "BRCLK".</td></tr>
<tr><td>bit1, bit0</td><td>PSS1, PSS0:
Prescaler selection bits</td><td><table>
<tr><th>PSS1</th><th>PSS0</th><th>Prescaler selection</th></tr>
<tr><td>0</td><td>0</td><td>1/1</td></tr>
<tr><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>1</td><td>0</td><td>1/4</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td></tr>
</table></td></tr>
</table>

21.3.2 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

The UART/SIO dedicated baud rate generator baud rate setting register (BRSR0) controls the baud rate settings.

■ UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

Figure 21.3-3 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
BRSR0	$0FBF_H$	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

This register sets the cycle of the 8-bit down-counter. This register can be used to set any baud rate clock. Write to the register when the UART is stopped.

Do not set BRS[7:0] to "00_H" or "01_H" in clock asynchronous mode.

21.4 Operating Description of UART/SIO Dedicated Baud Rate Generator

The UART/SIO dedicated baud rate generator serves as the baud rate generator for asynchronous clock mode.

■ Baud Rate Setting

The SMC10 register (CKS bit) of the UART/SIO is used to select the serial clock. This selects the UART/SIO dedicated baud rate generator.

In asynchronous CLK mode, the shift clock that is selected by the CKS bit and divided by four is used and transfers can be performed within the range from -2% to +2%. The baud rate calculation formula for the UART/SIO dedicated baud rate generator is shown below.

Figure 21.4-1 Baud Rate Calculation Formula when UART/SIO Dedicated Baud Rate Generator Is Used

$$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{4 \times \{1, 2, 4, 8\} \times \{2 : 255\}} \text{ [bps]}$$

UART prescaler selection register (PSSR0)
Prescaler selection
(PSS1, PSS0)

UART baud rate setting register (BRSR0)
Baud Rate Setting
(BRS7 to BRS0)

Table 21.4-1 Sample Asynchronous Transfer Rates by Baud Rate Generator (Machine Clock = 10MHz, 16MHz, 16.25MHz)

Settings of UART/SIO dedicated baud rate generator		UART Internal division	Total division ratio (PSS × BRS × 4)	Baud rate (10MHz/ Total division ratio)	Baud rate (16MHz/ Total division ratio)	Baud rate (16.25MHz/ Total division ratio)
Prescaler selection PSS[1:0]	Baud rate counter setting BRS[7:0]					
1 (Setting value:0, 0)	20	4	80	125000	200000	203125
1 (Setting value:0, 0)	22	4	88	113636	181818	184659
1 (Setting value:0, 0)	44	4	176	56818	90909	92330
1 (Setting value:0, 0)	87	4	348	28736	45977	46695
1 (Setting value:0, 0)	130	4	520	19231	30769	31250
2 (Setting value:0, 1)	130	4	1040	9615	15385	15625
4 (Setting value:1, 0)	130	4	2080	4808	7692	7813
8 (Setting value:1, 1)	130	4	4160	2404	3846	3906

The baud rate can be set in UART mode within the following range.

Table 21.4-2 Permissible Baud Rate Range in UART Mode

PSS[1:0]	BRS[7:0]
00_B to 11_B	02_H(2) to FF_H(255)

CHAPTER 22 LIN-UART

This chapter describes the function and operation of the LIN-UART

Code: CM26-00127-2E

Page: 379, 411, 437

22.1 Overview of LIN-UART

The LIN (Local Interconnect Network)-UART is a general-purpose serial data communication interface for synchronous or asynchronous (start-stop synchronization) communication with external devices. In addition to a bi-directional communication function (normal mode) and master/slave communication function (multiprocessor mode: supports both master and slave operation), the LIN-UART also supports the special functions used by the LIN bus.

■ Functions of LIN-UART

The LIN-UART is a general-purpose serial data communication interface for transmitting serial data to and receiving data from other CPUs and peripheral devices. Table 22.1-1 lists the functions of the LIN-UART.

Table 22.1-1 Functions of LIN-UART

	Function
Data buffer	Full-duplex double buffer
Serial input	The LIN-UART oversamples received data for five times to determine the received value by majority (only asynchronous mode).
Transfer mode	• Clock synchronization (Select start/stop synchronization, or start/stop bit) • Clock asynchronous (Start/stop bits available)
Baud rate	• Dedicated baud rate generator provided (made of a 15-bit reload counter) • The external clock can be inputted.The reload counter can also be used to adjust the external clock.
Data length	• 7 bits (not in synchronous or LIN mode) • 8 bits
Signaling	NRZ (Non Return to Zero)
Start bit timing	Synchronization with the start bit falling edge in asynchronous mode.
Reception error detection	• Framing error • Overrun error • Parity error (Not supported in operation mode 1)
Interrupt request	• Reception interrupts (reception completed, reception error detected, LIN synch break detected) • Transmit interrupts (send data empty) • Interrupt requests to TII0 (LIN synch field detected: LSYN)
Master/slave mode communication function (Multiprocessor mode)	Capable of 1 (master) to n (slaves) communication (support both the master and slave system)
Synchronous Mode	Send side/receive side of serial clock
Pin access	Serial I/O pin states can be read directly.
LIN bus option	• Master device operation • Slave device operation • LIN synch break detection • LIN Synch break generation • Detection of LIN synch field start/stop edges connected to the 8/16-bit compound timer
Synchronous serial clock	Continuous output to the SCK pin is possible for synchronous communication using the start/stop bits
Clock delay option	Special synchronous clock mode for delaying the clock (used for serial peripheral interface (SPI))

The LIN-UART has four operation modes. The operation mode is selected by the MD0 and MD1 bits in the LIN-UART serial mode register (SMR). Mode 0 and mode 2 are used for bi-directional serial communication; mode 1 for master/slave communication; and mode 3 for LIN master/slave communication.

Table 22.1-2 LIN-UART Operation Modes

<table>
<tr><th colspan="2" rowspan="2">Operation mode</th><th colspan="2">Data length</th><th rowspan="2">Synchronous method</th><th rowspan="2">Stop bit length</th><th rowspan="2">Data bit format</th></tr>
<tr><th>No parity</th><th>With parity</th></tr>
<tr><td>0</td><td>Normal mode</td><td colspan="2">7 bits or 8 bits</td><td>Asynchronous</td><td rowspan="2">1 bit or 2 bits</td><td rowspan="3">LSB first
MSB first</td></tr>
<tr><td>1</td><td>Multi processor mode</td><td>7 bits or 8 bits +1*</td><td>–</td><td>Asynchronous</td></tr>
<tr><td>2</td><td>Normal mode</td><td colspan="2">8 bits</td><td>Synchronous</td><td>None, 1 bit, 2 bits</td></tr>
<tr><td>3</td><td>LIN mode</td><td>8 bits</td><td>–</td><td>Asynchronous</td><td>1 bit</td><td>LSB first</td></tr>
</table>

-: Unavailable
*: "+1" is the address/data selection bit (AD) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operation modes.

Table 22.1-3 LIN-UART Operation Modes

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

- Mode 1 supports both master and slave operation for the multiprocessor mode.
- Mode 3 is fixed to communication format 8-bit data, no parity, 1 stop bit, LSB-first.

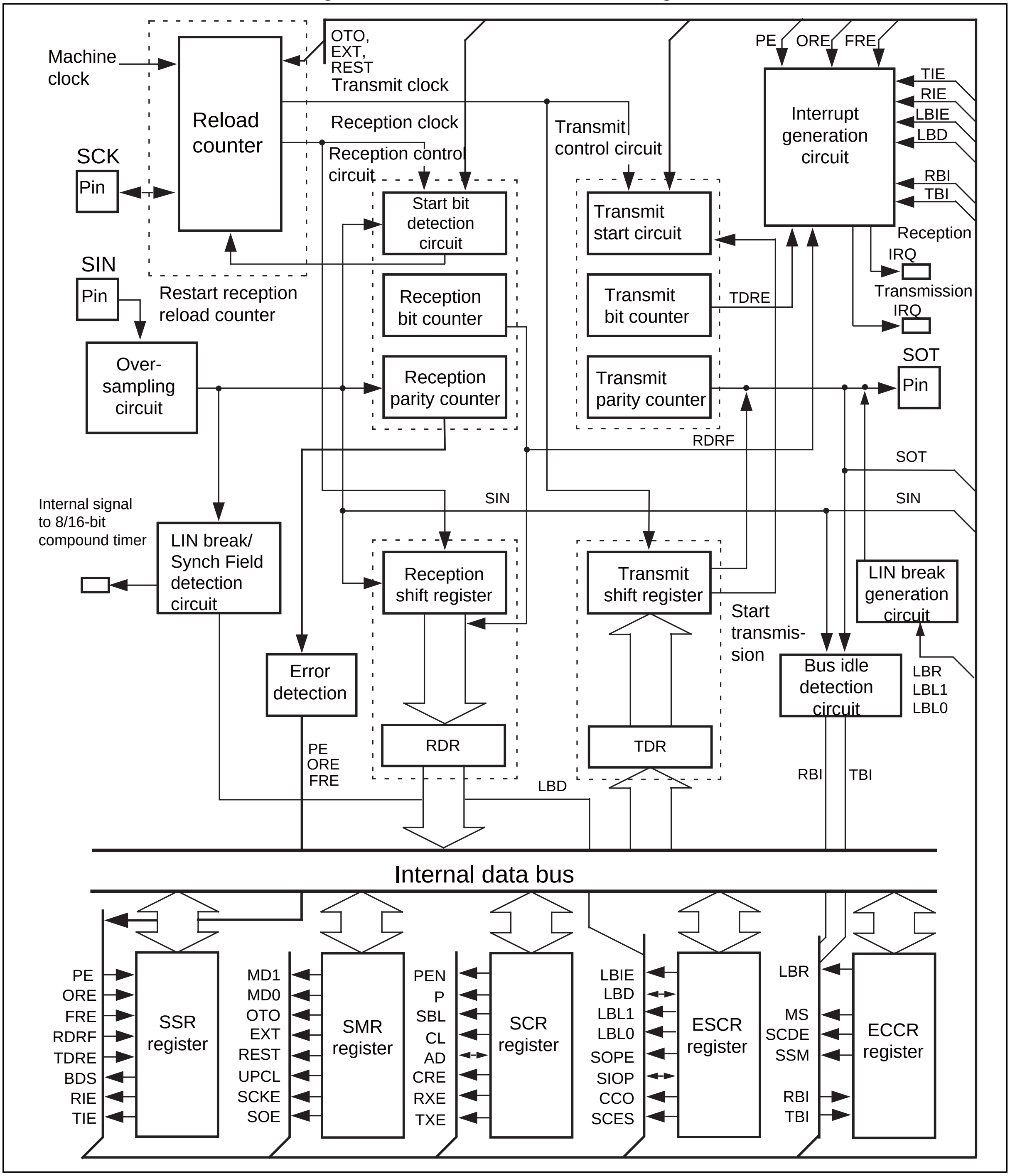
22.2 Configuration of LIN-UART

LIN-UART is made up of the following blocks.

- **Reload Counter**
- **Reception control circuit**
- **Reception shift register**
- **LIN-UART reception data register (RDR)**
- **Transmit control circuit**
- **Transmit shift register**
- **LIN-UART transmit data register (TDR)**
- **Error detection circuit**
- **Oversampling circuit**
- **Interrupt generation circuit**
- **LIN synch break/Synch Field detection circuit**
- **Bus idle detection circuit**
- **LIN-UART serial control register (SCR)**
- **LIN-UART serial mode register (SMR)**
- **LIN-UART serial status register (SSR)**
- **LIN-UART extended status control register (ESCR)**
- **LIN-UART extended communication control register (ECCR)**

■ LIN-UART Block Diagram

Figure 22.2-1 LIN-UART Block Diagram



● Reload counter

This block is a 15-bit reload counter serving as a dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/reception clock from the external or internal clock. The count value in the transmit reload counter is read from the baud rate generator 1, 0 (BGR1, BGR0).

- Reception control circuit

  This block consists of a reception bit counter, a start bit detection circuit, and a reception parity counter. The reception bit counter counts the reception data bits and sets a flag in the LIN-UART reception data register when one data reception is completed according to the specified data length. If the reception interrupt is enabled at this time, a reception interrupt request is generated. The start bit detection circuit detects a start bit in a serial input signal. When a start bit is detected, the circuit sends a signal to the reload counter in synchronization with the start bit falling edge. The reception parity counter calculates the parity of the received data.

- Reception shift register

  The circuit inputs received data from the SIN pin while bit-shifting and transfers it to the RDR register upon completion of reception.

- LIN-UART Reception Data Register (RDR)

  This register retains the received data. Serial input data is converted and stored in the LIN-UART reception data register.

- Transmit control circuit

  This block consists of a transmit bit counter, a transmission start circuit, and a transmit parity counter. The transmit bit counter counts the transmit data bits and sets a flag in the transmit data register when one data transmission is completed according to the specified data length. If the transmit interrupt is enabled at this time, a transmit interrupt request is generated. The transmit start circuit starts transmission when data is written to the TDR. The transmit parity counter generates a parity bit for data to be transmitted if the data is parity-checked.

- Transmit shift register

  The data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register, and output to the SOT pin during bit-shifting.

- LIN-UART transmit data register (TDR)

  This register sets the transmit data. The written data is converted to serial data and output.

- Error detection circuit

  This circuit detects an error upon completion of reception, if any. If an error occurs, the corresponding error flag is set.

- Oversampling circuit

  In asynchronous mode, the LIN-UART oversamples received data for five times to determine the received value by majority. The LIN-UART stops during operation in synchronous mode.

- Interrupt generation circuit

  This circuit controls all interrupt factors. An interrupt is generated immediately if the corresponding interrupt enable bit has been set.

- LIN synch break/Synch Field detection circuit

  This circuit detects a LIN synch break when the LIN master node transmits a message header. The LBD flag is set when the LIN synch break is detected. An internal signal is output to 8/16-bit compound timer in order to detect the first and fifth falling edges of the LIN synch Field and to measure the actual serial clock synchronization transmitted by the master node.

● LIN synch break generation circuit

This circuit generates a LIN synch break with the specified length.

● Bus idle detection circuit

This circuit detects that no transmission or reception is in progress, and generates the TBI and RBI flag bits.

● LIN-UART serial control register (SCR)

Operating functions are as follows:

- Sets parity bit existence
- Parity bit selection
- Sets stop bit length
- Sets data length
- Selects the frame data format in mode 1
- Clears error flag
- Enables/disables transmission
- Enables/disables reception

● LIN-UART serial mode register (SMR)

Operating functions are as follows:

- Selects the LIN-UART operation mode
- Selects a clock input source
- Selects between one-to-one connection or reload counter connection for the external clock
- Resets a dedicated reload timer
- LIN-UART software reset (maintains register settings)
- Enables/disables output to the serial data pin
- Enables/disables output to the clock pin

● LIN-UART serial status register (SSR)

Operating functions are as follows:

- Check transmission/reception or error status
- Selects the transfer direction (LSB-first or MSB-first)
- Enables/disables reception interrupts
- Enables/disables transmit interrupts

● Extended status control register (ESCR)

- Enables/disables LIN synch break interrupts
- LIN synch break detection
- Selects LIN synch break length
- Direct access to SIN pin and SOT pin
- Sets continuous clock output in LIN-UART synchronous clock mode
- Sampling clock edge selection

● LIN-UART extended communication control register (ECCR)

- Bus idle detection
- Synchronous clock setting
- LIN synch break generation

■ Input Clock

LIN-UART uses a machine clock or an input signal from the SCK pin as an input clock.

Input clock is used as clock source of transmission/reception of LIN-UART.

22.3 Pins of LIN-UART

This section describes LIN-UART pins.

■ Pins related to LIN-UART

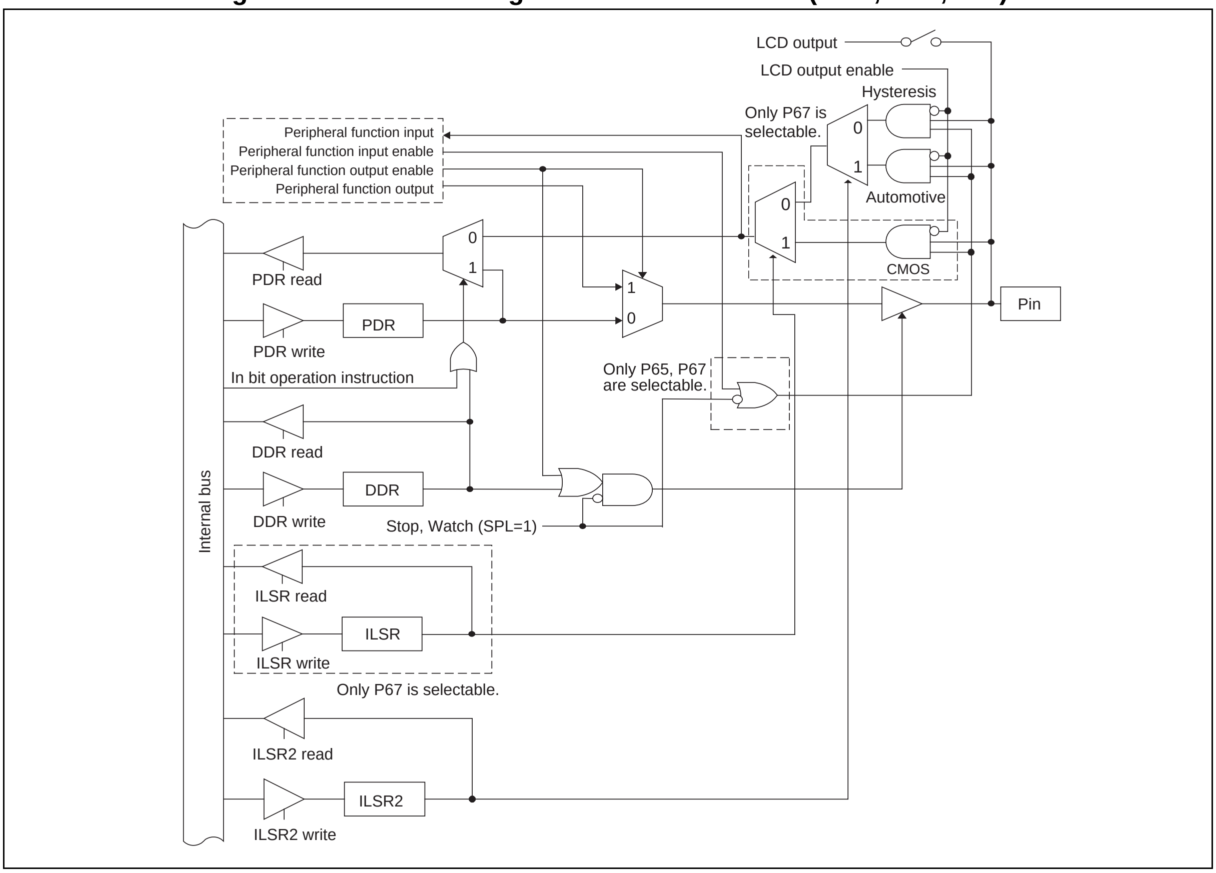
The LIN-UART pins also serve as general-purpose ports.Table 22.3-1 lists the LIN-UART pin.

Table 22.3-1 LIN-UART Pins

Pin name	Pin function	Required settings for using the pin
SIN	Serial data input	Set to the input port (DDR: corresponding bit = 0)
SOT	Serial data output	Set to output enable (SMR:SOE = 1)
SCK	Serial clock input/output	Set to the input port when used as clock input (DDR: corresponding bit = 0)
		Set to output enable when used as clock output (SMR:SCKE = 1)

■ Block Diagram of LIN-UART Pins

Figure 22.3-1 Block Diagram of LIN-UART Pins (SCK, SOT, SIN)



22.4 Registers of LIN-UART

This section lists the registers of LIN-UART.

■ Register List of LIN-UART

Figure 22.4-1 Register List of LIN-UART

LIN-UART serial control register (SCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050_H	PEN	P	SBL	CL	AD	CRE	RXE	TXE	00000000_B
	R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W	

LIN-UART serial mode register (SMR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0051_H	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE	00000000_B
	R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W	

LIN-UART serial status register (SSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0052_H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	00001000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W	

LIN-UART reception data register/LIN-UART transmit data register (RDR/TDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053_H	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LIN-UART extended status control register (ESCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0054_H	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	00000100_B
	R/W	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	

LIN-UART extended communication control register (ECCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0055_H	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI	$000000XX_B$
	RX,W0	R0,W	R/W	R/W	R/W	RX/W0	R/WX	R/WX	

LIN-UART baud rate generator register 1 (BGR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBC_H$	–	BGR14	BGR13	BGR12	BGR11	BGR10	BGR9	BGR8	00000000_B
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LIN-UART baud rate generator register 0 (BGR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBD_H$	BGR7	BGR6	BGR5	BGR4	BGR3	BGR2	BGR1	BGR0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R0, W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
RX,W0 : Reserved bit (Write value is indeterminate, write value is "0")

22.4.1 LIN-UART Serial Control Register (SCR)

The LIN-UART serial control register (SCR) is used to set parity, select the stop bit length and data length, select the frame data format in mode 1, clear the reception error flag, and enable/disable transmission/reception.

■ LIN-UART Serial Control Register (SCR)

Figure 22.4-2 LIN-UART Serial Control Register (SCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050H	PEN	P	SBL	CL	AD	CRE	RXE	TXE	00000000B
	R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W	

TXE	Transmit operation enable bit
0	Disable transmission
1	Enable transmission

RXE	Reception operation enable bit
0	Disable reception
1	Enable reception

CRE	Reception error flag clear bit	
	Write	Read
0	No effect	"0" is always read
1	Clear reception error flag (PE, FRE, ORE)	

AD	Address/data format selection bit
0	Data frame
1	Address data frame

CL	Data length selection bit
0	7-bit
1	8-bit

SBL	Stop bit length selection bit
0	1-bit
1	2-bit

P	Parity selection bit
0	Even parity
1	Odd parity

PEN	Parity enable bit
0	No parity
1	With parity

R/W : Readable/writable (Read value is the same as write value)
RX,WO : Write only (Writable, "0" is read)
(shaded) : Initial value

Table 22.4-1 Functions of Each Bit in LIN-UART Serial Control Register (SCR)

Bit name		Function
bit7	PEN: Parity enable bit	Specify whether or not to add (at transmission) and detect (at reception) a parity bit. Note: The parity bit is added only in operation mode 0, or in operation mode 2 with the settings that start/stop is set (ECCR:SSM=1). This bit is fixed to "0" in mode 3 (LIN).
bit6	P: Parity selection bit	Set either odd parity (1) or even parity (0) if the parity bit has been selected (SCR:PEN = 1).
bit5	SBL: Stop bit length selection bit	Set the bit length of the stop bit (frame end mark in transmit data) in operation mode 0, 1 (asynchronous) or in operation mode 2 (synchronous) with the settings that start/stop bit is set (ECCR:SSM=1). This bit is fixed to "0" in mode 3 (LIN).
bit4	CL: Data length selection bit	Specify the data length to be transmitted and received. This bit is fixed to "1" in mode 2 and mode 3.
bit3	AD: Address/Data format selection bit	Specify the data format for the frame to be transmitted and received in multiprocessor mode (mode 1). Write to this bit in master mode; read this bit in slave. The operation in master mode is as follows. **"0":** Set to data frame. **"1":** Set to address data frame. The value of last received data format is read. Note: See Section "22.8 Notes on Using LIN-UART" for using this bit.
bit2	CRE: Reception error flag clear bit	This bit is to clear FRE, ORE, and PE flags in serial status register (SSR). "0":No effect. **"1":** Clear the error flag. Reading this bit always returns "0". Note: Disable the reception operation (RXE=0) first, and then clear the reception error flags. If an error flag is cleared before the reception operation is disabled, the reception will be aborted at that time, and resume later. This may result in a data reception error.
bit1	RXE: Reception operation enable bit	Enable or disable the reception of LIN-UART. **"0":** Disable data frame reception. **"1":** Enable data frame reception. The LIN synch break detection in mode 3 is not affected. Note: When the reception is disabled (RXE = 0) during reception, the reception halts immediately. In that case, the data is not guaranteed.
bit0	TXE: Transmit operation enable bit	Enable or disable the transmission of LIN-UART. **"0":** Disable data frame transmission. **"1":** Enable data frame transmission. Note: When the transmission is disabled (TXE = 0) during transmission, the transmission halts immediately. In that case, the data is not guaranteed.

22.4.2 LIN-UART Serial Mode Register (SMR)

The LIN-UART serial mode register (SMR) is used to select the operation mode, specify the baud rate clock, and enable/disable output to the serial data and clock pins.

■ LIN-UART Serial Mode Register (SMR)

Figure 22.4-3 LIN-UART Serial Mode Register (SMR)

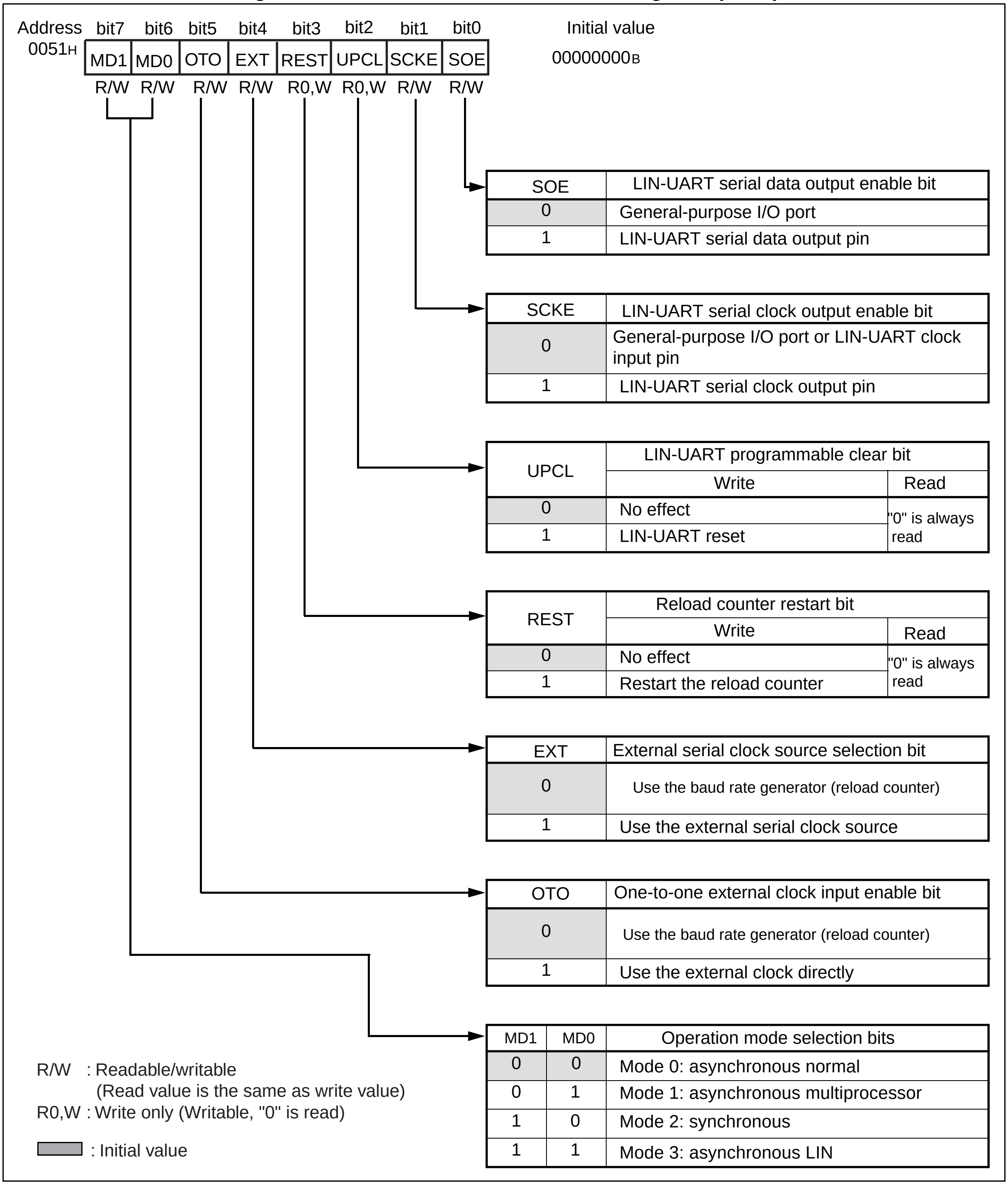


Table 22.4-2 Functions of Each Bit in LIN-UART Serial Mode Register (SMR)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7,
bit6</td><td>MD1, MD0:
Operation mode select bits</td><td>Set the operation mode.
Note: If the mode is changed during communication, the transmission/ reception of LIN-UART halts and waits for starting the next communication.
<table>
<tr><th>MD1</th><th>MD0</th><th>Mode</th><th>Type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Asynchronous (Normal mode)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Asynchronous (Multiprocessor mode)</td></tr>
<tr><td>1</td><td>0</td><td>2</td><td>Synchronous (Normal mode)</td></tr>
<tr><td>1</td><td>1</td><td>3</td><td>Asynchronous (LIN mode)</td></tr>
</table></td></tr>
<tr><td>bit5</td><td>OTO:
One-to-one external clock input enable bit</td><td>"1": Enable the external clock to be used directly as the LIN-UART serial clock.
Used for reception side of serial clock (ECCR:MS = 1) in operation mode 2 (synchronous).
When EXT = 0, the OTO bit is fixed to "0".</td></tr>
<tr><td>bit4</td><td>EXT:
External serial clock source select bit</td><td>Select a clock input.
"0": Select the clock of the internal baud rate generator (reload counter).
"1": Select the external serial clock source.</td></tr>
<tr><td>bit3</td><td>REST:
Reload counter restart bit</td><td>Restart the reload counter.
"0": No effect.
"1": Restart the reload counter.
Reading this bit always returns "0".</td></tr>
<tr><td>bit2</td><td>UPCL:
LIN-UART programmable clear bit
(Reset the LIN-UART software reset)</td><td>Reset the LIN-UART.
"0": No effect.
"1": Reset the LIN-UART immediately (LIN-UART software reset). However, the register settings are maintained. At that time, transmission and reception are halted.
All of the transmit/reception interrupt factors (TDRE, RDRF, LBD, PE, ORE, FRE) are reset. Reset the LIN-UART after the interrupt and transmission are disabled. Also, the reception data register is cleared (RDR = 00_H), and the reload counter is restarted.
Reading this bit always returns "0".</td></tr>
<tr><td>bit1</td><td>SCKE:
LIN-UART serial clock output enable bit</td><td>Control the serial clock I/O port.
"0": The SCK pin works as a general-purpose I/O port or a serial clock input pin.
"1": This pin works as the serial clock output pin and outputs the clock in operation mode 2 (synchronous).
Note: When the SCK pin is used as a serial clock input (SCKE = 0), set the corresponding DDR bits in the general-purpose I/O port as an input port. Also, select the external clock (EXT = 1) by using the clock select bit.
When the SCK pin is set as a serial clock output (SCKE = 1), this pin works as a serial clock output pin regardless of the state of the general-purpose I/O port.</td></tr>
<tr><td>bit0</td><td>SOE:
LIN-UART serial data output enable bit</td><td>Enable or disable output of serial data.
"0": The SOT pin works as a general-purpose I/O port.
"1": The SOT pin works as a serial data output pin (SOT).
When set as a serial data output (SOE = 1), the SOT pin works as a SOT pin regardless of a general-purpose I/O port.</td></tr>
</table>

MB95150/M Series

22.4.3 LIN-UART Serial Status Register (SSR)

The LIN-UART serial status register (SSR) is used to check the status of transmission/reception or error, and to enable/disable interrupts.

■ LIN-UART Serial Status Register (SSR)

Figure 22.4-4 LIN-UART serial status register (SSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0052H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	00001000B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W	

TIE	Transmit interrupt request enable bit
0	Disable transmit interrupts.
1	Enable transmit interrupts.

RIE	Reception interrupt request enable bit
0	Disable reception interrupts.
1	Enable reception interrupts.

BDS	Transfer direction selection bit
0	LSB-first (transfer from the least significant bit)
1	MSB-first (transfer from the most significant bit)

TDRE	Transmit data empty flag bit
0	Transmit data register (TDR) has data.
1	Transmit data register (TDR) is empty.

RDRF	Reception data full flag bit
0	Reception data register (RDR) is empty.
1	Reception data register (RDR) has data.

FRE	Framing error flag bit
0	No framing error
1	Framing error exists

ORE	Overrun error flag bit
0	No overrun error
1	Overrun error exists

PE	Parity error flag bit
0	Not parity error
1	Parity error exists

R/W : Readable/writable (Read value is the same as write value)
R/WX : Read only (Readable, writing has no effect on operation)
(shaded) : Initial value

Table 22.4-3 Functions of Each Bit in serial status register (SSR)

Bit name		Function
bit7	PE: Parity error flag bit	Detect a parity error in received data. • This bit is set to "1" when a parity error occurs during reception with PEN = 1, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • Output a reception interrupt request when both PE bit and RIE bit are "1". • When this flag is set, the data in the reception data register (RDR) is invalid.
bit6	ORE: Overrun error flag bit	Detect an overrun error in received data. • This bit is set to "1" when an overrun occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • Output a reception interrupt request when both ORE bit and RIE bit are "1". • When this flag is set, the data in the reception data register (RDR) is invalid.
bit5	FRE: Framing error flag bit	Detect a framing error in received data. • This bit is set to "1" when a framing error occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • Output a reception interrupt request when both FRE bit and RIE bit are "1". • When this flag is set, the data in the reception data register (RDR) is invalid.
bit4	RDRF: Reception data full flag bit	This flag shows the status of the reception data register (RDR). • This bit is set to "1" when received data is loaded into the reception data register (RDR), and cleared to "0" by reading RDR. • Output a reception interrupt request when both RDRF bit and RIE bit are "1".
bit3	TDRE: Transmit data empty flag bit	This flag shows the status of the transmit data register (TDR). • This bit is set to "0" by writing the transmit data to TDR, and indicates that the TDR has valid data.This bit is set to "1" when data is loaded into the transmit shift register and the transmission starts, and indicates that the TDR does not have effective data. • Output a transmit interrupt request when both TDRE bit and TIE bit are "1". • When the TDRE bit is "1", setting the LBR bit in the extended communication control register (ECCR) to "1" changes the TDRE bit to "0". Then, the TDRE bit goes back to "1" after LIN sync break is generated. Note: The initial state is TDRE = 1.
bit2	BDS: Transfer direction selection bit	Specify whether the transfer serial data is transfer from the least significant bit (LSB-first, BDS = 0) or from the most significant bit (MSB-first, BDS = 1). Note: Since data values are exchanged between the upper and lower when the data is read/written to the serial data register, changing BDS bit after writing data to the RDR register invalidates the written data. The BDS bit is fixed to "0" in mode 3 (LIN).
bit1	RIE: Reception interrupt request enable bit	Enable or disable the reception interrupt request output to the interrupt controller. Output a reception interrupt request when both the RIE bit and the reception data flag bit (RDRF) are "1", or when one or more error flag bits (PE, ORE, FRE) is "1".
bit0	TIE: Transmit interrupt request enable bit	Enable or disable the transmit interrupt request output to the interrupt controller. Output a transmit interrupt request when both TIE bit and TDRE bit are "1".

MB95150/M Series

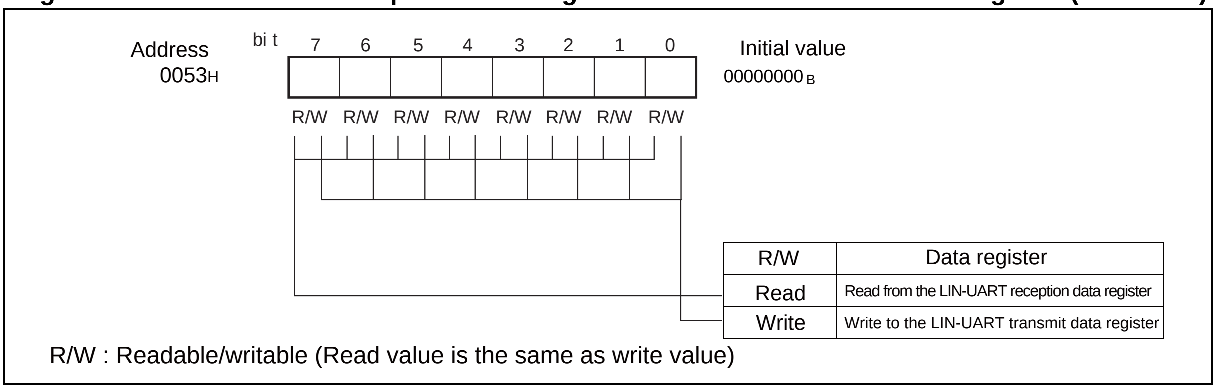
22.4.4 LIN-UART Reception Data Register/LIN-UART Transmit Data Register (RDR/TDR)

The LIN-UART reception and LIN-UART transmit data registers are located at the same address. If read, they work as the reception data register; if written, they work as the transmit data register.

■ LIN-UART Reception Data Register (RDR/TDR)

Figure 22.4-5 shows the LIN-UART reception data register/LIN-UART transmit data register.

Figure 22.4-5 LIN-UART Reception Data Register/LIN-UART Transmit Data Register (RDR/TDR)



■ LIN-UART Reception Data Register (RDR)

The LIN-UART reception data register (RDR) is the data buffer register for the serial data reception.

Serial data signal transmitted to the serial input pin (SIN pin) is converted via a shift register and stored in the LIN-UART reception data register (RDR).

If the data length is 7 bits, the upper 1 bit (RDR:D7) is "0".

The reception data full flag bit (SSR:RDRF) is set to "1" when received data is stored into the LIN-UART reception data register (RDR). If the reception interrupt is enabled (SSR:RIE = 1), a reception interrupt request is generated.

The LIN-UART reception data register (RDR) should be read when the reception data full flag bit (SSR:RDRF) is "1".The reception data full flag bit (SSR:RDRF) is automatically cleared to "0" by reading the LIN-UART reception data register (RDR). Also, the reception interrupt is cleared when the reception interrupt is enabled and no error occurs.

When the reception error occurs (any of SSR:PE, ORE, or FRE is "1"), the data in the LIN-UART reception data register (RDR) is invalid.

■ LIN-UART Transmit Data Register (TDR)

The LIN-UART transmit data register (TDR) is the data buffer register for the serial data transmission.

If the data to be transmitted is written to the LIN-UART transmit data register (TDR) when transmission is enabled (SCR:TXE = 1), the transmit data is transferred to the transmission shift register, converted to serial data, and output from the serial data output pin (SOT pin).

If the data length is 7 bits, the data in the upper 1 bit (TDR:D7) is invalid.

The transmit data empty flag (SSR:TDRE) is cleared to "0" when a transmit data is written to the LIN-UART transmit data register (TDR).

The transmit data empty flag (SSR:TDRE) is set to "1" after the data is transferred to the transmission shift register and the transmission starts.

If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written. If the transmit interrupt is enabled, a transmit interrupt is generated. The next transmit data should be written by generating the transmit interrupt, or when the transmit data empty flag (SSR:TDRE) is "1".

Note:

The LIN-UART transmit data register is a write-only register; the reception data register is a read-only register. Since both registers are located at the same address, the write value and read value are different. Thus, the instructions to operate the read-modify-write (RMW) instruction, such as the INC/DEC instruction, cannot be used.

22.4.5 LIN-UART Extended Status Control Register (ESCR)

The LIN-UART extended status control register (ESCR) has the settings for enabling/disabling LIN synch break interrupt, LIN synch break length selection, LIN synch break detection, direct access to the SIN and SOT pins, continuos clock output in LIN-UART synchronous clock mode and sampling clock edge.

■ Bit Configuration of LIN-UART Extended Status Control Register (ESCR)

Figure 22.4-6 shows the bit configuration of the LIN-UART extended status control register (ESCR). Table 22.4-4 lists the function of each bit in LIN-UART extended status control register (ESCR).

Figure 22.4-6 Bit Configuration of LIN-UART Extended Status Control Register (ESCR)

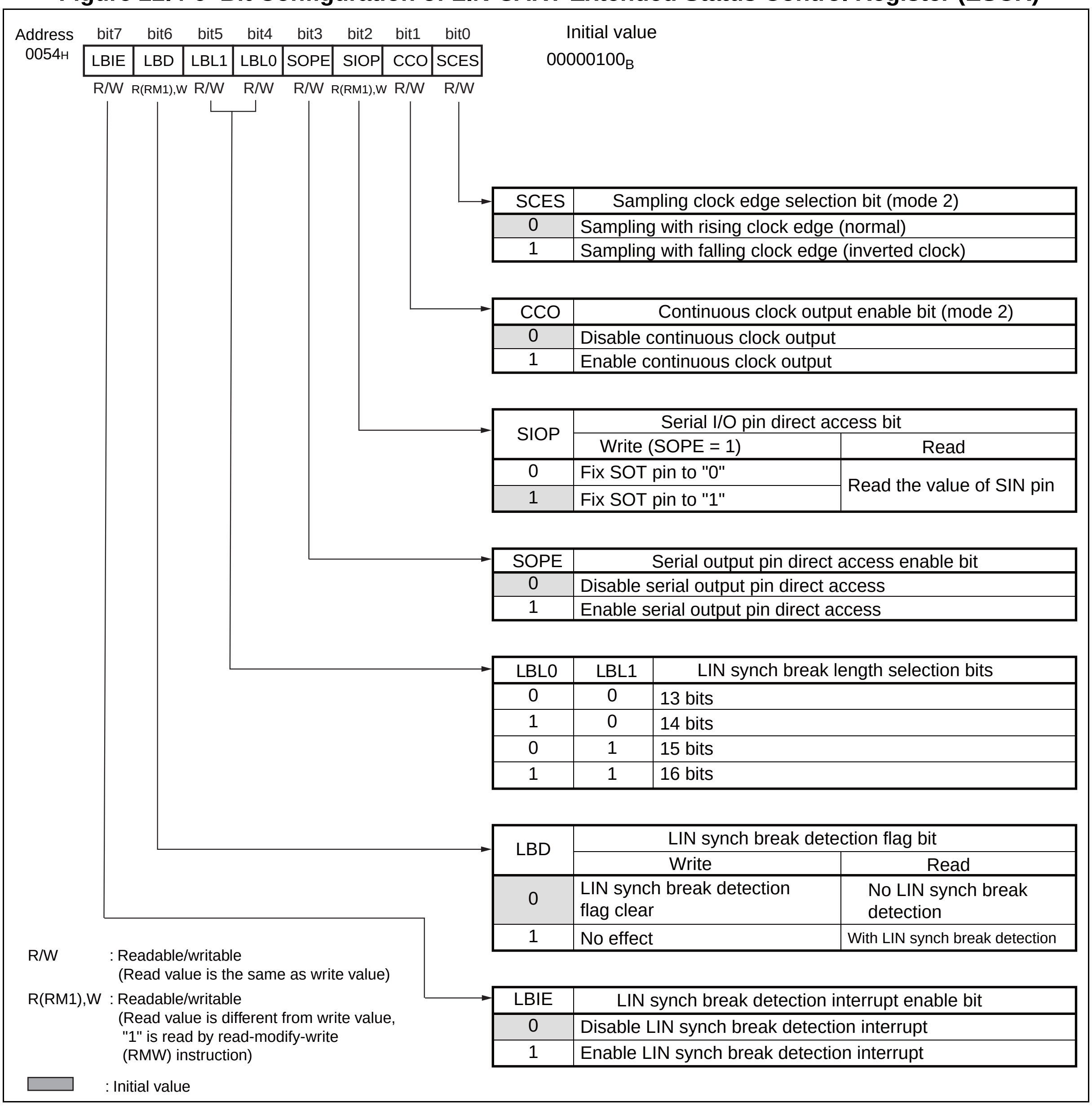


Table 22.4-4 Functions of Each Bit in LIN-UART Extended Status Control Register (ESCR)

Bit name		Function
bit7	LBIE: LIN synch break detection interrupt enable bit	This bit enables or disables LIN synch break detection interrupts. An interrupt is generated when the LIN synch break detection flag (LBD) is "1" and the interrupt is enabled (LBIE = 1). This bit is fixed to "0" in mode 1 and mode 2.
bit6	LBD: LIN synch break detection flag bit	Detect LIN synch break. This bit is set to "1" when the LIN synch break is detected in operation mode 3 (the serial input is "0" when bit width is 11 bits or more). Also, writing "0" clears the LBD bit and the interrupt.Although the bit is always read as "1" when the read-modify-write (RMW) instruction is executed, this does not indicate that a LIN synch break detected. Note: To detect a LIN synch break, enable the LIN synch break detection interrupt (LBIE = 1), and then disable the reception (SCR:RXE = 0).
bit5, bit4	LBL1/LBL0: LIN synch break length selection bits	These bits specify the bit length for the LIN synch break generation time. The LIN synch break length for reception is always 11 bits.
bit3	SOPE: Serial output pin direct access enable bit*	Enable or disable direct writing to the SOT pin. Setting this bit to "1" when serial data output is enabled (SMR:SOE = 1) enables direct writing to the SOT pin.*
bit2	SIOP: Serial I/O pin direct access bit*	Control direct access to the serial I/O pin. Normal read instruction always returns the value of the SIN pin. When direct access to the serial output pin data is enabled (SOPE = 1), the value written to this bit reflects the SOT pin.* Note: The bit operation instruction returns the bit value of the SOT pin in the read cycle.
bit1	CCO: Continuous clock output enable bit	Enable or disable continuous serial clock output from the SCK pin. Setting this bit to "1" with sending side of serial clock in operation mode 2 (synchronous) enables the continuous serial clock output from the SCK pin if the pin is set as a clock output. Note: When the CCO bit is "1", the SSM bit in the ECCR should be "1".
bit0	SCES: Sampling clock edge select bit	Select a sampling edge. Setting the SCES to "1" in reception side of serial clock in operation mode 2 (synchronous) switches the sampling edge from the rising edge to the falling edge. When the SCK pin is set as the clock output with sending side of serial clock (ECCR:MS = 0) in operation mode 2, the internal serial clock and the output clock signal are inverted. This bit should be "0" in operation modes 0, 1, and 3.

***: Interaction between SOPE and SIOP**

SOPE	SIOP	Write to SIOP	Read from SIOP
0	R/W	No effect (however, the write value is retained)	Return the SIN value
1	R/W	Write "0" or "1" to SOT	Return the SIN value
1	RMW	Read the SOT value, write "0" or "1"	

22.4.6 LIN-UART Extended Communication Control Register (ECCR)

The LIN-UART extended communication control register (ECCR) is used for the bus idle detection, the synchronous clock setting, and the LIN Synch break generation.

■ Bit Configuration of LIN-UART Extended Communication Control Register (ECCR)

Figure 22.4-7 shows the bit configuration of the LIN-UART extended communication control register (ECCR). Table 22.4-5 lists the function of each bit in the LIN-UART extended communication control register (ECCR).

Figure 22.4-7 Bit Configuration of LIN-UART Extended Communication Control Register (ECCR)

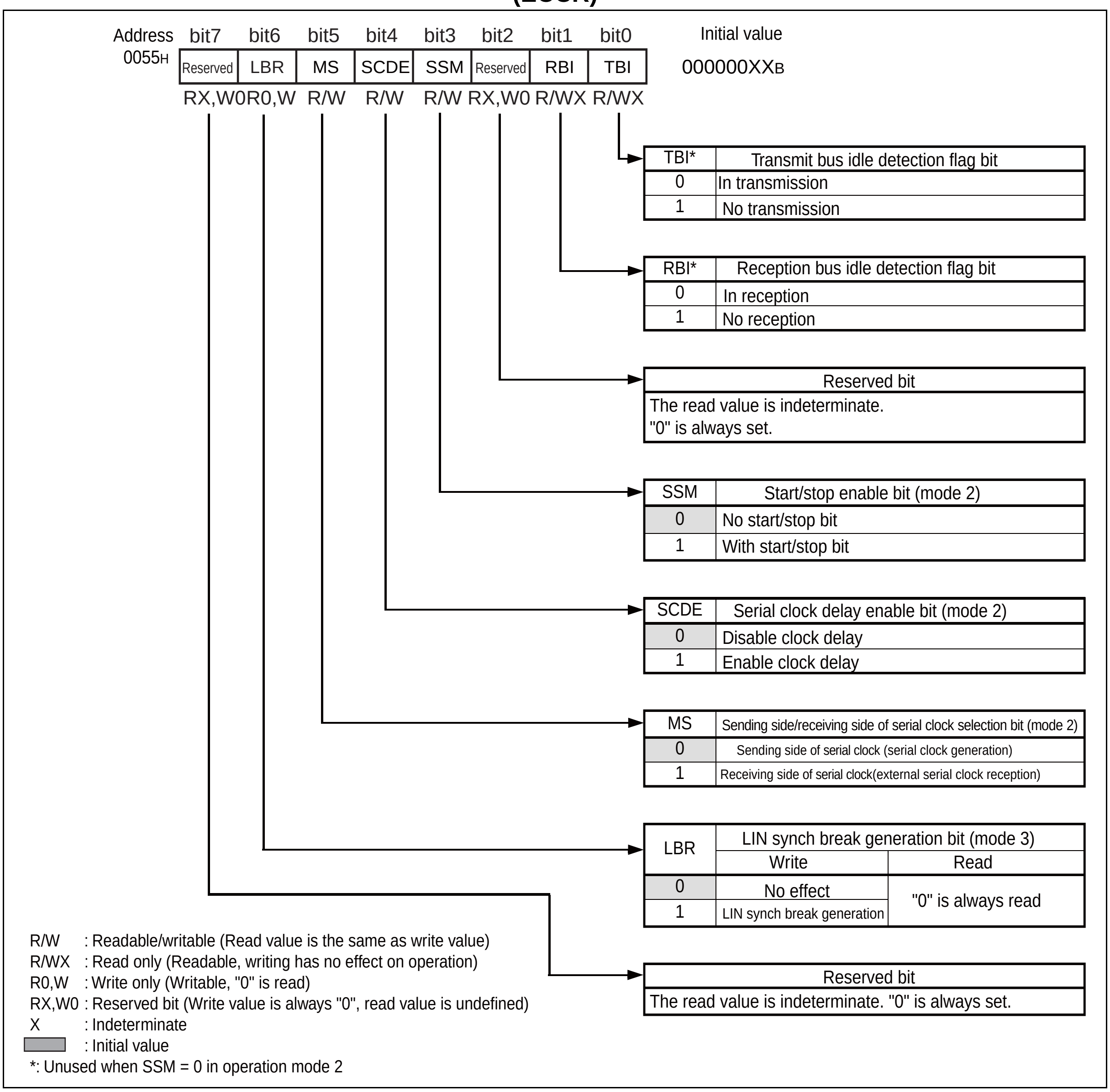


Table 22.4-5 Functions of Each Bit in LIN-UART Extended Communication Control Register (ECCR)

Bit name		Function
bit7	Reserved bit	The read value is indeterminate. "0" is always set.
bit6	LBR: LIN Synch break generation bit	Setting this bit to "1" in mode 3 generates a LIN synch break which has the length specified by LBL0/1 in the ESCR. This bit should be "0" in mode 0, 1, and 2.
bit5	MS: Sending side/receiving side of serial clock select bit	Select sending side/receiving side of serial clock in mode 2. When sending side "0" is selected, generate a synchronous clock. When receiving side "1" is selected, receive an external serial clock. This bit is fixed to "0" in modes 0, 1, and 3. Modify this bit only when the SCR:TXE bit is "0". Note: When receiving side of serial clock is selected, the clock source must be set as an external clock and the external clock input must be enabled (SMR:SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	Setting the SCDE bit to "1" at sending side of serial clock in mode 2 outputs a delayed serial clock as shown in Figure 22.7-5. This bit is effective in serial peripheral interface. This bit is fixed to "0" in modes 0, 1, and 3.
bit3	SSM: Start/stop bit mode enable bit	Add the start/stop bit to the synchronous data format when this bit is set to "1" in mode 2. This bit is fixed to "0" in modes 0, 1, and 3.
bit2	Reserved bit	The read value is indeterminate. "0" is always set.
bit1	RBI: Reception bus idle detection flag bit	When the SIN pin is "H" level and reception is not performed, this bit is "1". Do not use this bit when SSM = 0 in operation mode 2.
bit0	TBI: Transmit bus idle detection flag bit	This bit is "1" when there is no transmission on the SOT pin.Do not use this bit when SSM = 0 in operation mode 2.

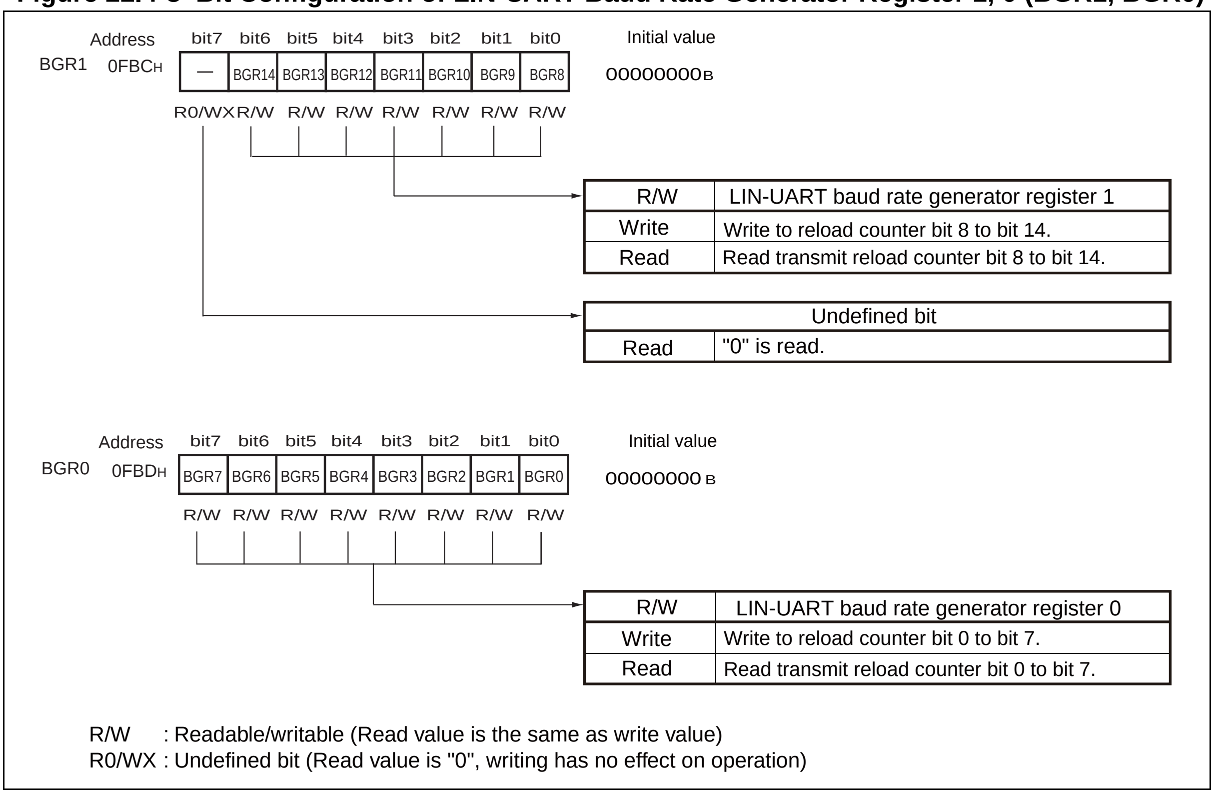
22.4.7 LIN-UART Baud Rate Generator Register 1, 0 1, 0 (BGR1, BGR0)

The LIN-UART baud rate generator register 1, 0 (BGR1, BGR0) sets the division ratio of the serial clock. Also, the count value in the transmit reload counter is read from this generator.

■ Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Figure 22.4-8 shows the bit configuration of LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

Figure 22.4-8 Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0)



The LIN-UART baud rate generator register sets the division ratio of the serial clock.

BGR1 is associated with the upper bits; BGR0 is associated with the lower bits. The reload value of the counter can be written and the transmit reload counter value can be read from them. Byte/word access is also possible.

Writing a reload value to the LIN-UART baud rate generator registers causes the reload counter to start counting.

Note:

Write to this register when LIN-UART stops.

22.5 Interrupt of LIN-UART

The LIN-UART has reception interrupts and transmit interrupts, which are generated by following factor and have the assigned interrupt number and interrupt vector. Also, it has the LIN synch field edge detection interrupt function using the 8/16-bit compound timer interrupt.

- **Reception interrupt**
  When the received data is set in the reception data register (RDR), or when a reception error occurs. Also, when a LIN synch break is detected.
- **Transmit interrupt**
  When the transmit data is transferred from the transmit data register (TDR) to the transmission shift register, and the transmission starts.

■ Reception Interrupt

Table 22.5-1 shows the interrupt control bits and interrupt factors of reception interrupts.

Table 22.5-1 Interrupt Control Bits and Interrupt Factors of Reception Interrupts

Interrupt request flag bit	Flag register	Operation mode				Interrupt source	Interrupt factor enable bit	Clearing of interrupt request flag
		0	1	2	3			
RDRF	SSR	❍	❍	❍	❍	Write received data to RDR	SSR:RIE	Read received data
ORE	SSR	❍	❍	❍	❍	Overrun error		Write "1" to reception error flag clear bit (SCR:CRE)
FRE	SSR	❍	❍	∆	❍	Framing error		
PE	SSR	❍	×	∆	×	Parity error		
LBD	ESCR	×	×	×	❍	LIN synch break detection	ESCR:LBIE	Write "0" to ESCR:LBD

❍: Used bit
×: Unused bit
∆: Only ECCR:SSM = 1 available

● Reception interrupt

Each flag bit in the LIN-UART serial status register (SSR) is set to "1" when any of following operation occurs in reception mode:

Data reception completed

When the reception data is transferred from the serial input shift register to the LIN-UART reception data register (RDR) (RDRF = 1)

Overrun error

When the following serial data is received when RDRF = 1 and the RDR is not read by the CPU (ORE = 1)

Framing error

When the stop bit reception error occurs (FRE = 1)

Parity error

When the parity detection error occurs (PE = 1)

A reception interrupt request is generated if the reception interrupt is enabled (SSR:RIE = 1) when any of the above flag bits is "1".

RDRF flag is automatically cleared to "0" by reading the LIN-UART reception data register (RDR). All of error flags are cleared to "0" by writing "1" to the reception error flag clear bit (CRE) in the LIN-UART serial control register (SCR).

Note:

For the CRE bit, disable the reception operation (RXE=0) first, and then clear the reception error flags.If an error flag is cleared before the reception operation is disabled, the reception will be aborted at that time, and resume later. This may result in a data reception error.

- LIN synch break interrupts

Works for LIN slave operation in operation mode 3.

The LIN synch break detection flag bit (LBD) in the LIN-UART extended status control register (ESCR) is set to "1" when the internal data bus (serial input) is "0" for 11 bits or longer. The LIN synch break interrupt and the LBD flag are cleared by writing "0" to the LBD flag. The LBD flag must be cleared before the 8/16-bit compound timer interrupt is generated in the LIN synch field.

To detect a LIN synch break, the reception must be disabled (SCR:RXE = 0).

■ Transmit Interrupts

Table 22.5-2 shows the interrupt control bits and interrupt factors of transmit interrupts.

Table 22.5-2 Interrupt Control Bits and Interrupt Factors of Transmit Interrupts

Interrupt request flag bit	Flag register	Operation mode				Interrupt factor	Interrupt request enable bit	Clearing of interrupt request flag
		0	1	2	3			
TDRE	SSR	❍	❍	❍	❍	Transmit register is empty	SSR:TIE	Write transmit data

❍: Used bit

- Transmit interrupt

The transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is set to "1" when the transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmission shift register, and the transmission starts. If the transmit interrupt is enabled (SSR:TIE = 1) in this case, a transmit interrupt request is generated.

Note:

Since the initial value of TDRE is "1", an interrupt is generated immediately after the TIE bit is set to "1" after hardware/software reset.Also, the TDRE is cleared only by writing data to the LIN-UART transmit data register (TDR).

■ LIN Synch Field Edge Detection Interrupt (8/16-bit Compound Timer Interrupt)

Table 22.5-3 shows the interrupt control bits and interrupt factors of the LIN synch field edge detection interrupt.

Table 22.5-3 Interrupt Control Bits and Interrupt Factors of LIN Synch Field Edge Detection Interrupt

Interrupt request flag bit	Flag register	Operation mode				Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
		0	1	2	3			
IR	T00CR1	×	×	×	❍	First falling edge of the LIN synch field	T00CR1:IE	Write "0" to T00CR1:IR
IR	T00CR1	×	×	×	❍	Fifth falling edge of the LIN synch field		

❍: Used bit
X: Unused bit

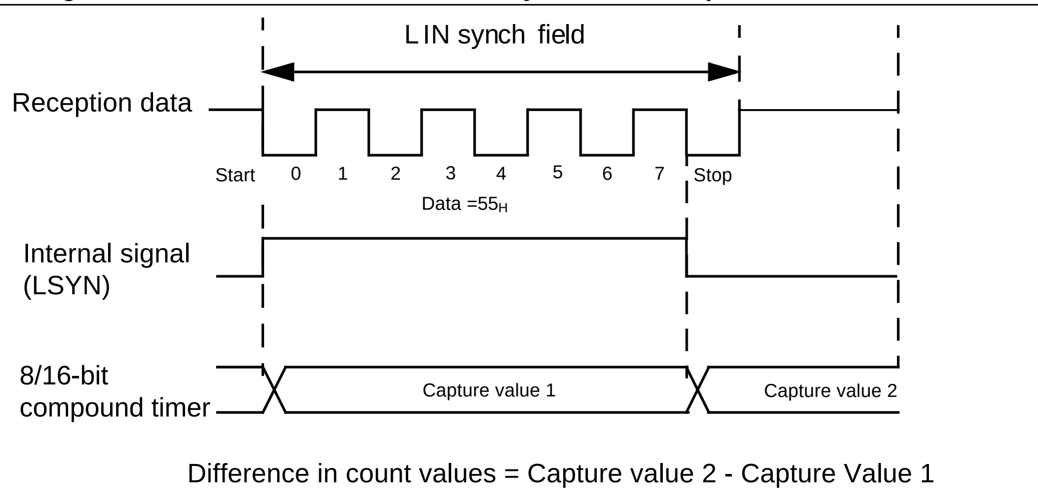
● LIN synch field edge detection interrupt (8/16-bit compound timer interrupt)

Works for LIN slave operation in operation mode 3.

After a LIN synch break is detected, the internal signal (LSYN) is set to "1" at the first falling edge of the LIN synch field, and set to "0" after the fifth falling edge. When the 8/16-bit compound timer is configured to be input the internal signal and to detect both edges, a 8/16-bit compound timer interrupt is generated if enabled.

The difference in the count values detected by the 8/16-bit compound timer (see Figure 22.5-1) corresponds to the 8 bits in the master serial clock. The new baud rate can be calculated from this value.
When a new baud rate is set, the rate become effective from the falling edge detection of the specified next start bit.

Figure 22.5-1 Baud Rate Calculation by 8/16-bit Compound Timer



■ Register and Vector Table Related to LIN-UART Interrupt

Table 22.5-4 Register and Vector Table Related to LIN-UART Interrupt

Interrupt sources	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Reception	IRQ7	ILR1	L07	$FFEC_H$	$FFED_H$
Transmission	IRQ8	ILR2	L08	$FFEA_H$	$FFEB_H$

22.5.1 Reception Interrupt Generation and Flag Set Timing

Reception interrupts are a reception completion and an occurrence of a reception error.

■ Reception Interrupt Generation and Flag Set Timing

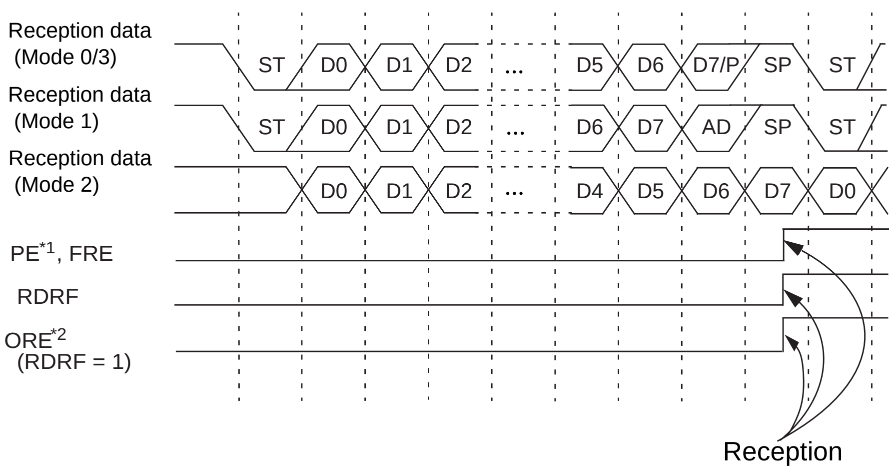
Received data is stored in the LIN-UART reception data register (RDR) when the first stop bit is detected in mode 0, 1, 2 (SSM =1), 3, or when the last data bit is detected in mode 2 (SSM = 0). Each error flag is set when a reception is completed (SSR:RDRF = 1), or when a reception error occurs (SSR:PE, ORE, FRE = 1). If the reception interrupt is enabled (SSR:RIE = 1) at this time, a reception interrupt is generated.

Note:

When a reception error occurs in each mode, the data in the LIN-UART reception data register (RDR) is invalid.

Figure 22.5-2 shows the timing of reception and flag set.

Figure 22.5-2 Timing of Reception and Flag Set

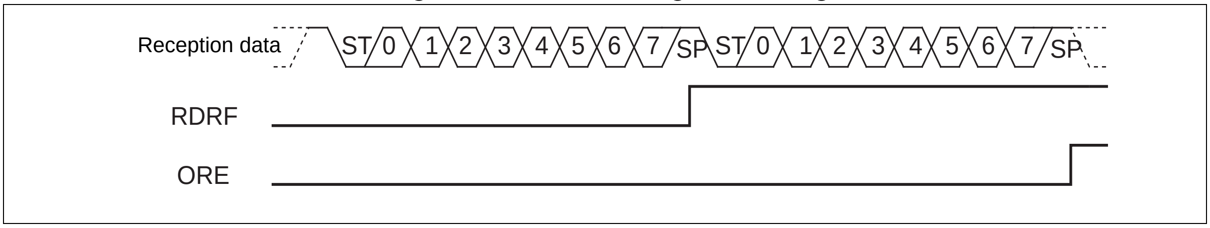


*1 : PE flag is always "0" in modes 1 and 3.
*2 : An overrun error is generated if the next data is transferred before a received data is read (RDRF = 1).
ST : Start bit, SP: Stop bit, AD: Mode 1 (multiprocessor) address data select bit

Note:

Figure 22.5-2 does not show all receptions in mode 0. It only shows examples for 7-bit data, parity (even parity or odd parity), 1 stop bit and 8-bit data, no parity, 1 stop bit.

Figure 22.5-3 ORE Flag Set Timing



22.5.2 Transmit Interrupt Generation and Flag Set Timing

Transmit interrupts are generated when the transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmission shift register and then the transmission starts.

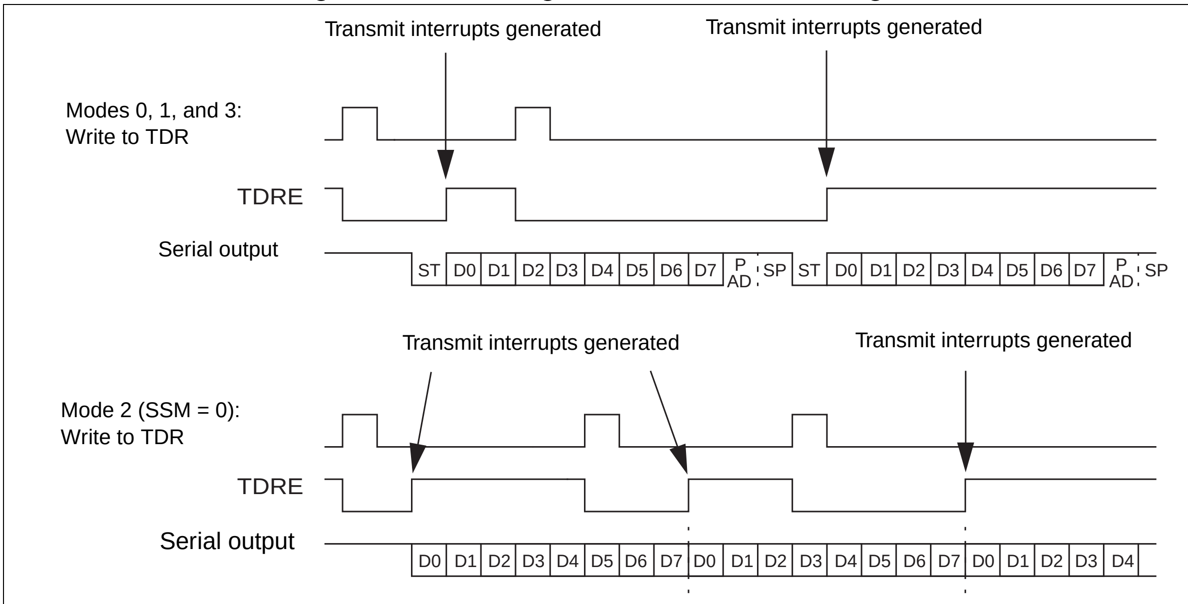
■ Transmit Interrupt Generation and Flag Set Timing

When the data written to the LIN-UART transmit data register (TDR) is transferred to the transmission shift register and then the transmission starts, the next data becomes to be writable (SSR:TDRE = 1). If the transmit interrupt is enabled (SSR:TIE = 1) at this time, a transmit interrupt is generated.

TDRE bit is a read-only bit and cleared to "0" only by writing data to the LIN-UART transmit data register (TDR).

Figure 22.5-4 shows the timing of the transmission and flag set.

Figure 22.5-4 Timing of Transmission and Flag Set



ST: Start bit, D0 to D7: Data bits, P: Parity, SP: Stop bit
AD: Address data select bit (mode 1)

Note:

Figure 22.5-4 does not show all transmissions in mode 0. It only shows "8P1" (P= "even parity" or "odd parity").

No parity bit is transmitted in mode 3, or in mode 2 with SSM = 0.

■ Transmit Interrupt Request Generation Timing

When TDRE flag is set to "1" if the transmit interrupt is enabled (SSR:TIE = 1), a transmit interrupt is generated.

Note:

Since the TDRE bit is initially set to "1", a transmit interrupt is generated immediately after the transmit interrupt is enabled (SSR:TIE = 1).Be careful with the timing for enabling the transmit interrupt since the TDRE bit can be cleared only by writing new data to the LIN-UART transmit data register (TDR).

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

22.6 LIN-UART Baud Rate

One of the following can be selected for the LIN-UART input clock (send/receive clock source):

- **Input a machine clock into a baud rate generator (reload counter)**
- **Input an external clock into a baud rate generator (reload counter)**
- **Use the external clock (SCK pin input clock) directly**

■ LIN-UART Baud Rate Selection

You can select one of the following three different baud rates. Figure 22.6-1 shows the LIN-UART baud rate selection circuit.

● Baud rate derived from the internal clock divided by the dedicated baud rate generator (reload counter)

Two internal reload counters are provided and assigned the transmit and reception serial clock respectively. The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock by the specified value.

It is used in asynchronous mode and in synchronous mode (sending side of serial clock).

To set the clock source, select the use of the internal clock and baud rate generator (SMR:EXT = 0, OTO = 0).

● Baud rate derived from the external clock divided by the dedicated baud rate generator (reload counter)

The external clock is used as the clock source for the reload counter.

The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

The reload counter divides the external clock by the specified value.

It is used in asynchronous mode.

To set the clock source, select the use of the external clock and baud rate generator (SMR:EXT = 1, OTO = 0).

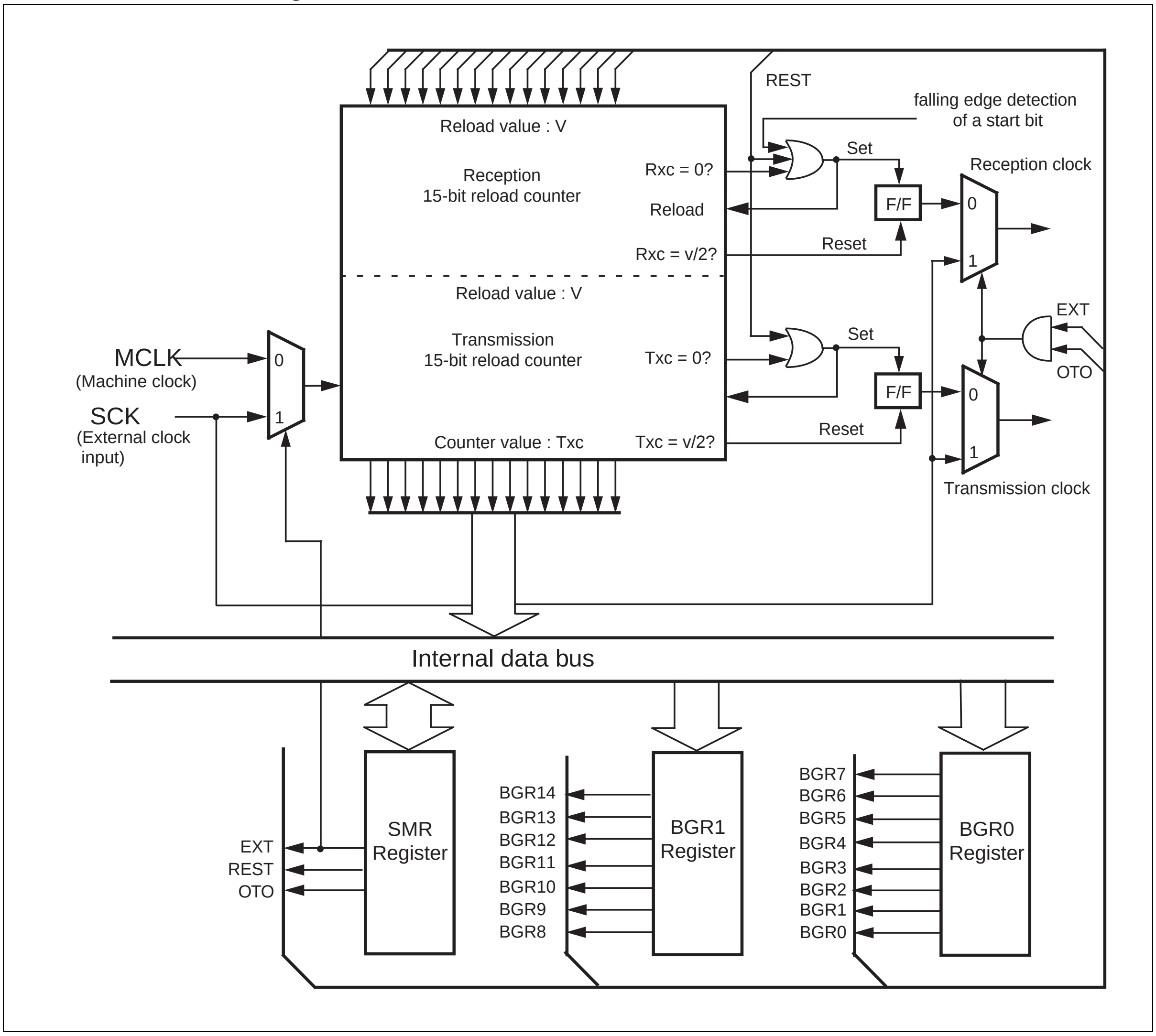
● Baud rate by the external clock (one-to-one mode)

The clock input from the clock input pin (SCK) of the LIN-UART is used as the baud rate (slave 2 operation (ECCR:MS =1) in synchronous mode).

It is used in synchronous mode (receiving side of serial clock).

To set the clock source, select the external clock and its direct use (SMR:EXT = 1, OTO = 1).

Figure 22.6-1 LIN-UART Baud Rate Selection Circuit



22.6.1 Baud Rate Setting

This section shows baud rate settings and the calculation result of serial clock frequencies.

■ Baud Rate Calculation

The two 15-bit reload counters are set by the baud rate generator register 1, 0 (BGR1, BGR0).

The expressions for the baud rate are as follows.

Reload value:

$$v = \left(\frac{MCLK}{b}\right) - 1$$

v: Reload value, b: Baud rate, MCLK: Machine clock, or external clock frequency

Calculation example

Assuming that the machine clock is 10MHz, the internal clock is used, and the baud rate is set to 19200 bps:

Reload value:

$$v = \left(\frac{10 \times 10^6}{19200}\right) - 1 \quad = 519.83... \quad \cong 520$$

Thus, the actual baud rate can be calculated as follows.

$$b = \frac{MCLK}{(v + 1)} = \frac{10 \times 10^6}{521} \quad = 19193.8579$$

Note:

The reload counter halts if the reload value is set to "0".Therefore, the least reload value should be "1".

For transmission/reception in asynchronous mode, the reload value must be at least "4" in order to determine the reception value by oversampling on five times.

■ Reload Value and Baud Rate of Each Clock Speed

Table 22.6-1 shows the reload value and baud rate.

Table 22.6-1 Reload Value and Baud Rate

Baud rate (bps)	8MHz (MCLK)		10MHz (MCLK)		16MHz (MCLK)		16.25MHz(MCLK)	
	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation
2M	–	–	4	0	7	0	–	–
1M	7	0	9	0	15	0	–	–
500000	15	0	19	0	31	0	–	–
400800	–	–	–	–	–	–	–	–
250000	31	0	39	0	63	0	64	0
230400	–	–	–	–	68	- 0.64	–	–
153600	51	- 0.16	64	- 0.16	103	- 0.16	105	0.19
125000	63	0	79	0	127	0	129	0
115200	68	- 0.64	86	0.22	138	0.08	140	- 0.04
76800	103	0.16	129	0.16	207	- 0.16	211	0.19
57600	138	0.08	173	0.22	277	0.08	281	- 0.04
38400	207	0.16	259	0.16	416	0.08	422	- 0.04
28800	277	0.08	346	- 0.06	555	0.08	563	- 0.04
19200	416	0.08	520	0.03	832	- 0.04	845	- 0.04
10417	767	< 0.01	959	< 0.01	1535	< 0.01	1559	< 0.01
9600	832	- 0.04	1041	0.03	1666	0.02	1692	0.02
7200	1110	< 0.01	1388	< 0.01	2221	< 0.01	2256	< 0.01
4800	1666	0.02	2082	- 0.02	3332	< 0.01	3384	< 0.01
2400	3332	< 0.01	4166	< 0.01	6666	< 0.01	6770	< 0.01
1200	6666	< 0.01	8334	< 0.01	13332	< 0.01	13541	< 0.01
600	13332	< 0.01	16666	< 0.01	26666	< 0.01	27082	< 0.01
300	26666	< 0.01	–	–	53332	< 0.01	54166	< 0.01

The unit of frequency deviation (dev.) is %.MCLK indicates the machine clock.

■ External Clock

The external clock is selected by writing "1" to the EXT bit in the LIN-UART serial mode register (SMR). In the baud rate generator, the external clock can be used in the same way as the internal clock.

When slave operation is used in synchronous mode 2, select the one-to-one external clock input mode (SMR:OTO = 1). In this mode, the external clock input to SCK is input directly to the LIN-UART serial clock.

Note:

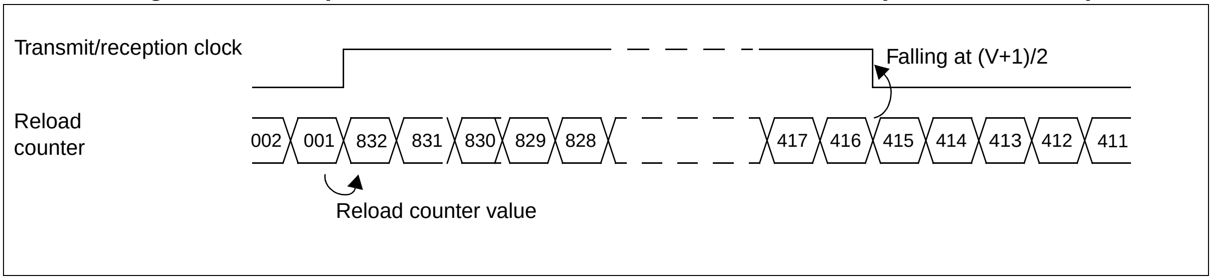
The external clock signal is synchronized with the internal clock (MCLK: machine clock) in the LIN-UART. Therefore, the signal is unstable because the external clock cannot be divided if its cycle is faster than half cycle of the internal clock.
Be sure not to set the cycle of the external clock is faster than half cycle of the internal clock.
For the value of the SCK clock, see "Data Sheet".

MB95150/M Series

■ Operation of Dedicated Baud Rate Generator (Reload Counter)

Figure 22.6-2 shows the operation of dedicated baud rate generator (reload counter).

Figure 22.6-2 Operation of Dedicated Baud Rate Generator (Reload Counter)



Note:

The falling edge of the serial clock signal is generated after the reload value divided by 2 ((v+1)/2) is counted.

22.6.2 Reload Counter

This block is a 15-bit reload counter serving as a dedicated baud rate generator. It generates the transmit/reception clock from the external or internal clock.
The count value in the transmit reload counter is read from the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

■ Function of Reload Counter

There are two kinds of reload counters; transmit and reception. They work as the dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/reception clock from the external or internal clock. The count value in the transmit reload counter is read from the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

● Start counting

Writing a reload value to the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) causes the reload counter to start counting.

● Restart

The reload counter restarts in the following conditions.

For both transmit/reception reload counter

- LIN-UART programmable reset (SMR:UPCL bit)
- Programmable restart (SMR:REST bit)

For reception reload counter

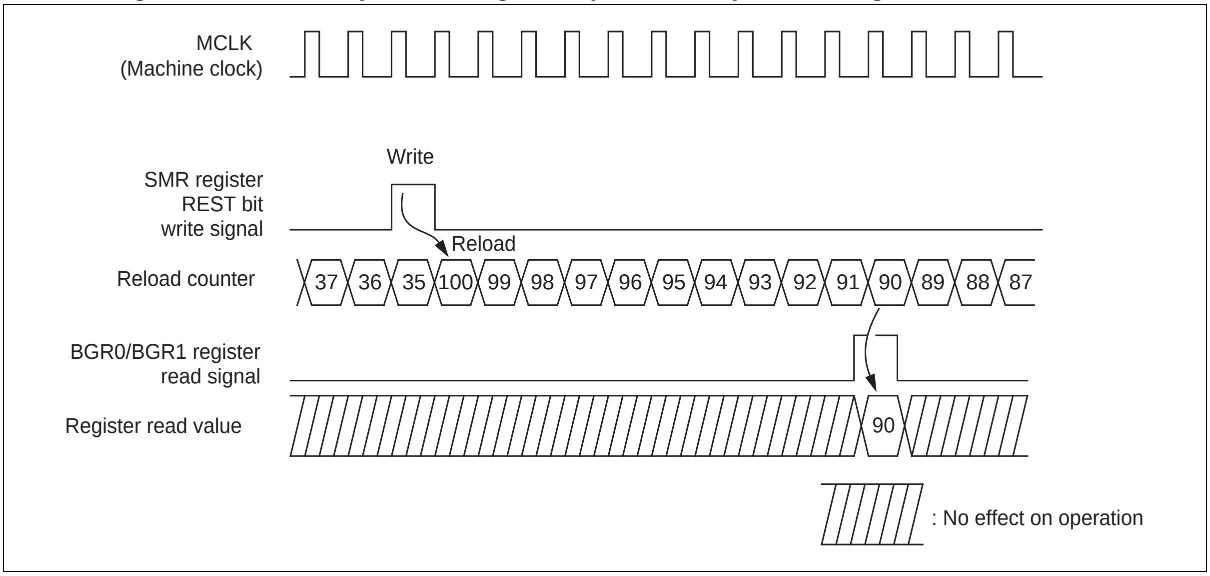
Start bit falling edge detection in asynchronous mode

● Simple timer function

Two reload counters restart at the next clock cycle when the REST bit in the LIN-UART serial mode register (SMR) is set to "1".

This function enables the transmit reload counter to be used as a simple timer.
Figure 22.6-3 shows an example of using a simple timer by restarting the reload timer (when reload value is 100).

Figure 22.6-3 Example of Using a Simple Timer by Restarting the Reload Timer



The number of machine cycles "cyc" after restart in this example is obtained by the following expression.

cyc = v - c + 1 = 100 - 90 + 1 = 11

v: Reload value, c: Reload counter value

Note:

The reload counters also restart when the LIN-UART is reset by writing "1" to the SMR:UPCL bit.

Automatic restart (reception reload counter only)

The reception reload counter is restarted when the start bit falling edge is detected in asynchronous mode. This is the function to synchronize the reception shift register with the reception data.

● Clear counter

When a reset occurs, the reload values in the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) and the reload counter are cleared to "00_H", and the reload counter halts.

Although the counter value is temporarily cleared to "00_H" by the LIN-UART reset (writing "1" to SMR:UPCL), the reload counter restarts since the reload value is retained.

The counter value is not cleared to "00_H" by the restart setting (writing "1" to SMR:REST), and the reload counter restarts.

22.7 Operations and Setup Procedure Example of LIN-UART

LIN-UART operates in mode 0, 2 for bi-directional serial communication, in mode 1 for master/slave communication, and in mode 3 for LIN master/slave communication.

■ Operation of LIN-UART

● Operation mode

The LIN-UART has four operation modes (0 to 3), allowing the connections between CPUs and the data transfer methods to be selected as listed in Table 22.7-1.

Table 22.7-1 LIN-UART Operation Modes

<table>
<tr><th colspan="2" rowspan="2">Operation mode</th><th colspan="2">Data length</th><th rowspan="2">Synchronous method</th><th rowspan="2">Stop bit length</th><th rowspan="2">Data bit format</th></tr>
<tr><th>No parity</th><th>With parity</th></tr>
<tr><td>0</td><td>Normal mode</td><td colspan="2">7 bits or 8 bits</td><td>Asynchronous</td><td rowspan="2">1 bit
or
2 bits</td><td rowspan="3">LSB first
MSB first</td></tr>
<tr><td>1</td><td>Multiprocessor mode</td><td>7 bits or
8 bits + 1*</td><td>–</td><td>Asynchronous</td></tr>
<tr><td>2</td><td>Normal mode</td><td colspan="2">8 bits</td><td>Synchronous</td><td>None, 1 bit, 2 bits</td></tr>
<tr><td>3</td><td>LIN mode</td><td>8 bits</td><td>–</td><td>Asynchronous</td><td>1 bit</td><td>LSB first</td></tr>
</table>

–: Setting disabled
*: "+1" is the address/data selection bit (AD) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operation modes.

Table 22.7-2 LIN-UART Operation Modes

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

Notes:

- Both master and slave operation are supported in a system with master/slave connection in mode 1.
- In mode 3, the communication format is fixed to 8-bit data, no parity, 1 stop bit, LSB-first.
- If the mode is changed, all transmissions and receptions are canceled, and the LIN-UART waits for the next operation.

■ Inter-CPU Connection Method

You can select either external clock one-to-one connection (normal mode) or master/slave connection (multiprocessor mode). In either methods, data length, parity setting, synchronization type must be the same between all CPUs and thus the operation mode must be selected as follows.

- One-to-one connection: Two CPUs must use the same method in either operation mode 0 or 2. Choose operation mode 0 in an asynchronous system and operation mode 2 in a synchronous system. Also, for the operation mode 2, set one CPU as sending side of serial clock and the other as the receiving side of serial clock.
- Master/slave connection: Select operation mode 1. Use the system as a master/slave system.

■ Synchronous Method

In asynchronous method, the reception clock is synchronized with the reception start bit falling edge. In synchronous method, the reception clock can be synchronized by the sending side of serial clock signal or the clock signal at operating as sending side of serial clock.

■ Signaling

NRZ (Non Return to Zero).

■ Enable Transmission/Reception

The LIN-UART uses the SCR:TXE bit and the SCR:RXE bit to control transmission and reception, respectively. To disable transmission or reception, set as follows.

- If the reception is in progress, wait until the reception completed, read the reception data register (RDR), and then disable the reception.
- If the transmission is in progress, wait until the transmission completed, and then disable the transmission.

■ Setup Procedure Example

LIN-UART is set in the following procedure:

● Initial setting

1) Set the port input (DDR6).

2) Set the interrupt level (ILR1, ILR2).

3) Set the data format, enable transmission/reception (SCR).

4) The operation mode, baud rate selection, pin output enabled (SMR)

5) The baud rate generator1, 0 (BGR1,BGR0)

1)

22.7.1 Operation of Asynchronous Mode (Operation Mode 0, 1)

When LIN-UART is used in operation mode 0 (normal mode) or operation mode 1 (multiprocessor mode), the transfer method is asynchronous.

■ Asynchronous Mode Operation

● Transmit/reception data format

Transmit/reception data always begins with a start bit ("L" level) followed by a specified data bits length and ends up with at least one stop bit ("H" level).

The bit transfer direction (LSB-first or MSB-first) is determined by the BDS bit in the LIN-UART serial status register (SSR). When a parity is used, the parity bit is always placed between the last data bit and the first stop bit.

In operation mode 0, select 7-bit or 8-bit for the data length. You can select whether or not to use a parity. Also, the stop bit length (1 or 2) can be selected.

In operation mode 1, a data length is 7-bit or 8-bit, the parity is not added, and the address/data bit is added. The stop bit length (1 or 2) can be selected.

The bit length of transmit/reception frame is calculated as follows:

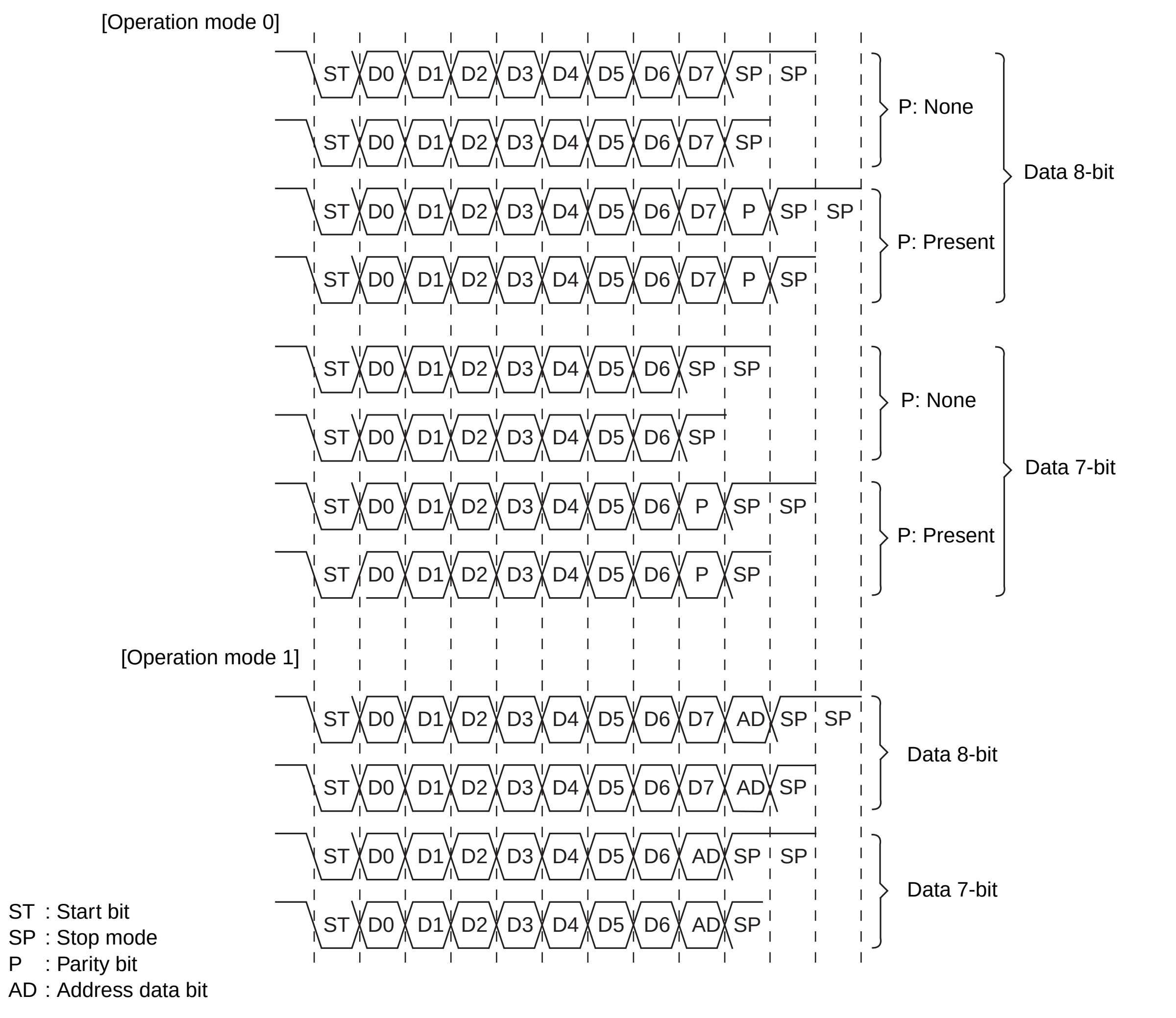
Length = 1 + d + p + s

(d = Number of data bits [7 or 8], p = parity [0 or 1],

s = Number of stop bits [1 or 2])

Figure 22.7-1 shows the transmit/reception data format (Operation Mode 0, 1).

Figure 22.7-1 Transmit/Reception Data Format (Operation Mode 0, 1)



Note:

When the BDS bit in the LIN-UART serial status register (SSR) is set to "1" (MSB-first), the bits are processed in the order of D7, D6, ... D1, D0 (P).

- **Transmission**

If the transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is "1", transmit data can be written into the LIN-UART transmit data register (TDR). Writing data sets the TDRE flag to "0". If transmission is enabled (SCR:TXE=1) at this time, the data is written to the transmit shift register and the transmission is started sequentially from the start bit in the next serial clock cycle.

If the transmit interrupt is enabled (TIE=1), the transmit data is transferred from the LIN-UART transmit data register (TDR) to transmit shift register, the TDRE flag is set to "1", and an interrupt occurs.

When the data length is set to 7-bit (CL=0), the bit7 in the TDR register is an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Since the initial value of transmit data empty flag bit (SSR:TDRE) is "1", an interrupt is generated immediately when transmit interrupts are enabled (SSR:TIE =1).

- **Reception**

The reception is performed when reception is enabled (SCR:RXE =1). When the start bit is detected, one frame data is received according to the data format defined in the LIN-UART serial control register (SCR). If an error occurs, the error flag (SSR:PE, ORE, FRE) is set. After the reception of the one frame data is completed, the received data is transferred from the reception shift register to the LIN-UART reception data register (RDR), and the reception data register full flag bit (SSR:RDRF) is set to "1". If the reception interrupt request is enabled (SSR:RIE = 1) at this time, a reception interrupt request is output.

To read the received data, check the error flag status and read the received data from the LIN-UART reception data register (RDR) if the reception is normal. If a reception error occurs, perform error handlings.

When the received data is read, the reception data register full flag bit (SSR:RDRF) is cleared to "0".

When the data length is set to 7-bit (CL=0), the bit7 in the RDR register is an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Data in the LIN-UART reception data register (RDR) becomes valid when the reception data register full flag bit (SSR:RDRF) is set to "1" and no error occurs (SSR:PE, ORE, FRE=0).

● Input clock

Internal or external clock is used.For the baud rate, select the baud rate generator (SMR:EXT = 0 or 1, OTO = 0).

● Stop bit and reception bus idle flag

You can select one or two stop bits at transmission. When 2-bit of the stop bit are selected, both of the stop bits are detected during reception.

When the first stop bit is detected, the reception data register full flag (SSR:RDRF) is set to "1". When no start bit is detected after that, the reception bus idle flag (ECCR:RBI) is set to "1", indicating that the reception is not performed.

● Error detection

In mode 0, parity, overrun, and framing errors can be detected.

In mode 1, overrun and framing errors can be detected. But, parity errors cannot be detected.

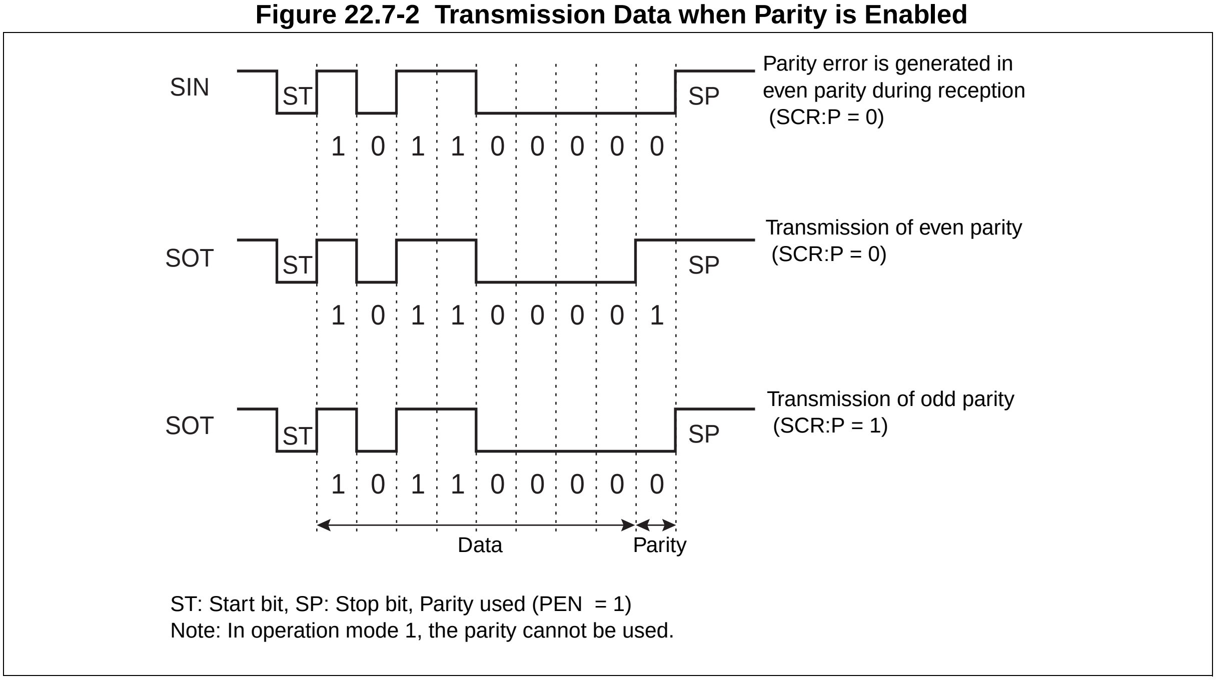
● Parity

You can specify whether or not to add (at transmission) and detect (at reception) a parity bit.

The parity enable bit (SCR:PEN) can be used whether or not to use a parity; the parity selection bit (SCR:P) can be used to select the odd or even parity.

In operation mode 1, the parity cannot be used.

Figure 22.7-2 Transmission Data when Parity is Enabled



● Data signaling

NRZ data format.

● Data transfer method

The data bit transfer method can be the LSB-first or MSB-first.

22.7.2 Operation of Synchronous Mode (Operation Mode 2)

When LIN-UART is used in operation mode 2 (normal mode), the transfer method is clock synchronous.

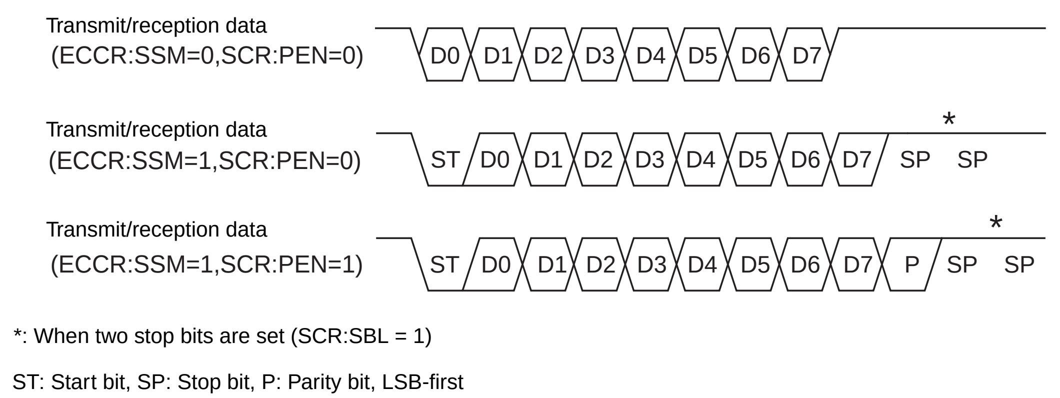
■ Operation of Synchronous Mode (Operation Mode 2)

● Transmit/reception data format

In synchronous mode, you can transmit and receive 8-bit data and select whether or not to include the start bit and stop bit (ECCR:SSM). When the start/stop bit is included (ECCR:SSM = 1), you can select whether or not to include the parity bit (SCR:PEN).

Figure 22.7-3 shows the transmit/reception data format (operation mode 2).

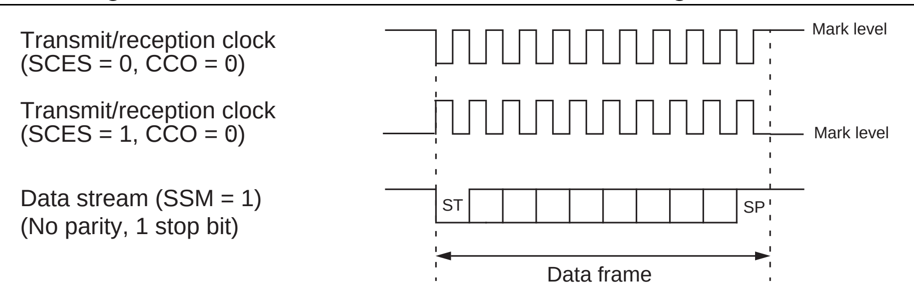
Figure 22.7-3 Transmit/Reception Data Format (Operation Mode 2)



● Clock inversion function

When the SCES bit in the LIN-UART extended status control register (ESCR) is "1", the serial clock is inverted.In receiving side of serial clock, the LIN-UART samples data at the falling edge of the received serial clock.Note that, in sending side of serial clock, the mark level is set to "0" when the SCES bit is "1".

Figure 22.7-4 Transmission Data Format During Clock Inverted



● Start/stop bit

When the SSM bit in the LIN-UART extended communication control register (ECCR) is "1", the start and stop bits are added to the data format as in asynchronous mode.

● Clock supply

In clock synchronous mode (normal), the number of the transmit/reception bits must be equal to the number of the clock cycles. When the start/stop bit is enabled, the number of the added start/stop bits must be equal, as well.

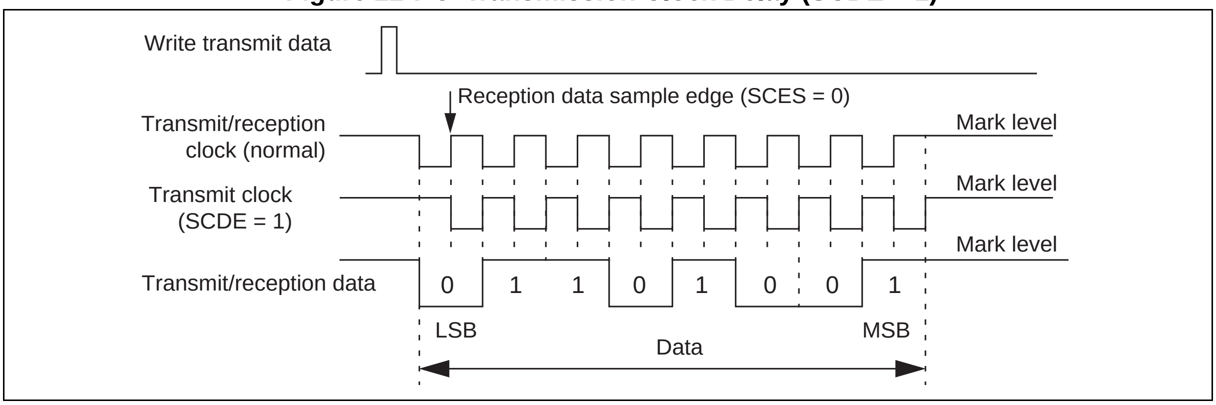
When the serial clock output is enabled (SMR: SCKE = 1) in sending side of serial clock (ECCR:MS = 0), a synchronous clock is output automatically at transmission/reception. When the serial clock output is disabled (SMR: SCKE = 0) in receiving side of serial clock (ECCR:MS = 1), the clock for each bit of transmit/reception data must be supplied from the outside.

The clock signal must remain at the mark level ("H") as long as it is irrelevant to transmission/reception.

● Clock delay

Setting the SCDE bit in the ECCR to "1", a delayed transmit clock is output as shown in Figure 22.7-5. This function is required when the receiving device samples data at the rising or falling edge of the clock.

Figure 22.7-5 Transmission Clock Delay (SCDE = 1)



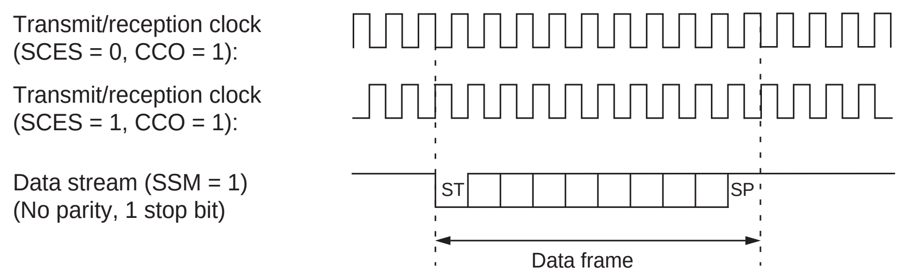
● Clock inversion

When the SCES bit in the LIN-UART extended status register (ESCR) is "1", the LIN-UART clock is inverted, and received data is sampled at the falling edge of the clock. At this time, the value of the serial data must be enabled at the timing of the clock falling edge.

- Continuous clock supply

When the CCO bit in the ESCR is "1", the serial clock output from the SCK pin is supplied in the sending side of serial clock continuously. In this mode, add the start/stop bit to the data format (SSM = 1) in order to identify the beginning and end of the data frame. Figure 22.7-6 shows the continuous clock supply (mode 2).

Figure 22.7-6 Continuous Clock Supply (Mode 2)



- Error detection

When the start/stop bits are disabled (ECCR:SSM=0), only overrun errors are detected.

- Communication settings for synchronous mode

To communicate in synchronous mode, the following settings are required.

- LIN-UART baud rate generator register 1, 0 (BGR0, BGR1)

  Set the dedicated baud rate reload counter to a required value.

- LIN-UART serial mode register (SMR)

  MD1, MD0: "10_B" (Mode 2)

  SCKE: "1": Use the dedicated baud rate reload counter

  "0": Input external clock

  SOE: "1": Enable transmission/reception

  : "0": Enable reception only

- LIN-UART serial control register (SCR)

  RXE, TXE: Set either bit to "1".

  AD : The value of this bit is disabled so that the address/data selection function cannot be used.

  CL : This bit is set to 8 bits length automatically, and its value is disabled.

  CRE : "1": Since the error flag is cleared, transmission/reception is stopped.

  - For SSM = 0:

    PEN, P, SBL: Since not used, parity bit and stop bit are disabled.

  - For SSM = 1:

    PEN : "1": Add/detect parity bit, "0": Not use parity bit

    P : "1": Odd parity, "0": Even parity

    SBL : "1": Stop bit length 2, "0": Stop bit length 1

- LIN-UART serial status register (SSR)

  BDS: "0": LSB first, "1": MSB-first

  RIE: "1": Enable reception interrupt, "0": Disable reception interrupt

  TIE: "1": Enable transmit interrupt, "0": Disable transmit interrupt

- LIN-UART extended communication control register (ECCR)

  SSM: "0": Not use start/stop bit (normal),

  "1": Use start/stop bit (extended function),

  MS: "0": Sending side of serial clock (serial clock output),

  "1": Receiving side of serial clock (input serial clock from sending side of serial clock)

Note:

To start communication, write data into the LIN-UART transmit data register (TDR).

To receive data, disable the serial output (SMR:SOE = 0), and then write dummy data into the TDR.

Enabling continuous clock and start/stop bit allows bi-directional communication as in asynchronous mode.

22.7.3 Operation of LIN function (Operation Mode 3)

In operation mode 3, the LIN-UART works as the LIN master and the LIN slave. In operation mode 3, the communication format is set to 8-bit data, no parity, stop bit1, LSB first.

■ Asynchronous LIN Mode Operation

● Operation as LIN master

In LIN mode, the master determines the baud rate for the entire bus, and the slave synchronizes to the master.

Writing "1" to the LBR bit in the LIN-UART extended communication control register (ECCR) outputs 13 to 16 bits at the "L" level from the SOT pin.This bit is the LIN synch break signifying the beginning of a LIN message.

The TDRE flag bit in the LIN-UART serial status register (SSR) is set to "0". After the break, it is set to "1" (initial value). If the TIE bit in SSR is "1" at this time, a transmit interrupt is output.

The length of the LIN Synch break transmitted is set by the LBL0/LBL1 bits in ESCR as in the following table.

Table 22.7-3 LIN Break Length

LBL0	LBL1	Break length
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

Synch field is transmitted as byte data 55_H following the LIN break. To prevent generation of a transmit interrupt, 55_H can be written to the TDR after the LBR bit in ECCR is set to "1" even if the TDRE flag is "0".

● Operation as LIN slave

In LIN slave mode, the LIN-UART must synchronize to the baud rate for the master. The LIN-UART generates a reception interrupt when LIN break interrupt is enabled (LBIE = 1) even though reception is disabled (RXE = 0). The LBD bit in the ESCR is set to "1" at this time.

Writing "0" to the LBD bit clears the reception interrupt request flag.

For calculation of the baud rate, the following example shows the operation of the LIN-UART. When the LIN-UART detects the first falling edge of Synch field, set an internal signal, which is input to the 8/16-bit compound timer, to "H", and then start the timer. The internal signal should be "L" at the fifth falling edge. The 8/16-bit compound timer must be set to the input capture mode. Also, the 8/16-bit compound timer interrupts must be enabled and set for the detection at both edges.The time for which the input signal to the 8/16-bit compound timer is "1" becomes the value obtained by multiplying the baud rate by 8.

The baud rate setting value is calculated by the following expressions.

When the counter of the 8/16-bit compound timer is not overflowing

: BGR value = (b - a)/8 - 1

When the counter of the 8/16-bit compound timer is overflowing

: BGR value = (max + b - a)/8 - 1

max: Maximum value of free-run timer

a: TII0 data register value after the first interrupt

b: TII0 data register value after the second interrupt

Note:

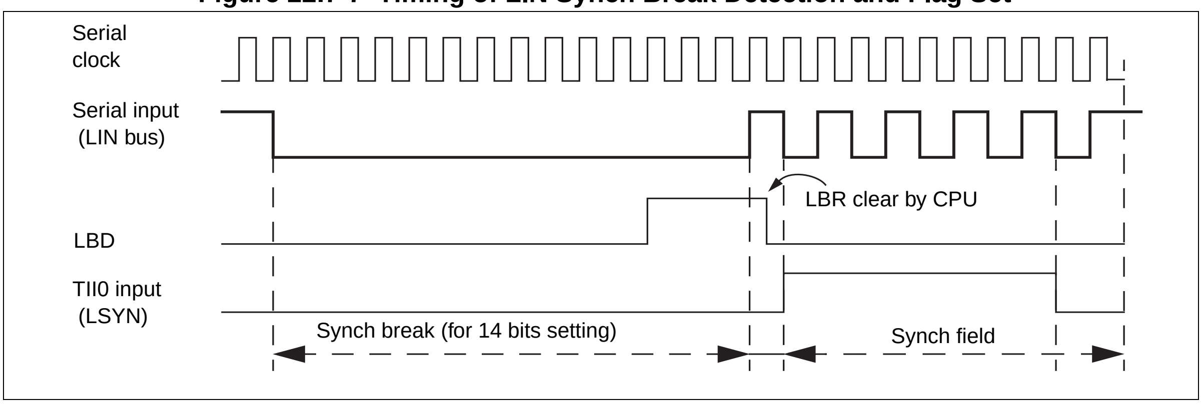
Do not set the baud rate if the new BGR value calculated based on Synch field as above in LIN slave mode involves an error over ±15%.

For the operations of the input capture function on the 8/16-bit compound timer, see Section "15.13 Operating Description of Input Capture Function".

● LIN synch break detection interrupt and flag

The LIN break detection (LBD) flag in ESCR is set to "1" when the LIN synch break is detected in slave mode.When the LIN break interrupt is enabled (LBIE = 1), an interrupt is generated.

Figure 22.7-7 Timing of LIN Synch Break Detection and Flag Set



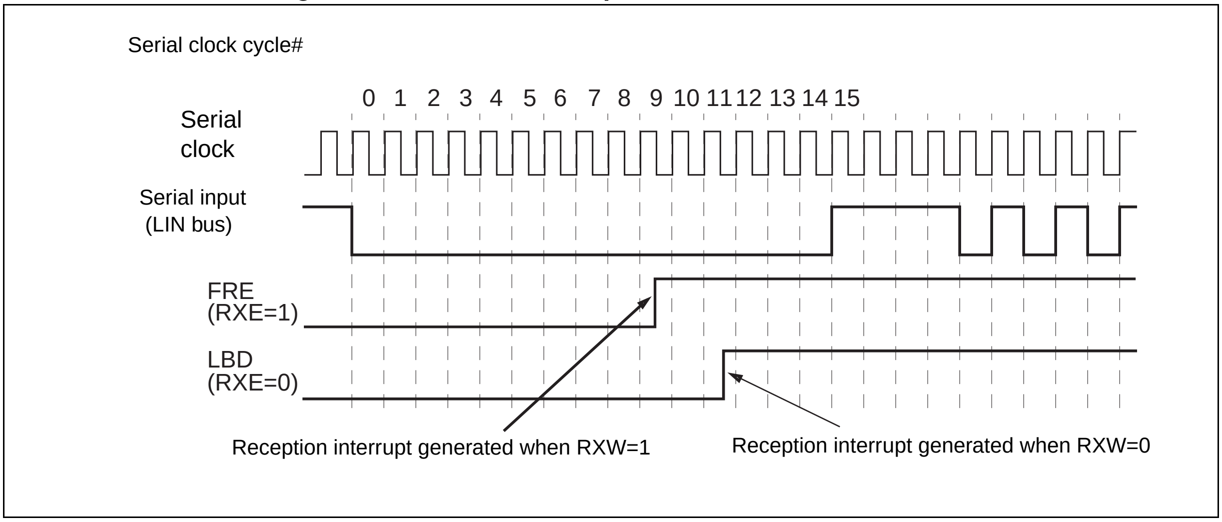
The above diagram shows the timing of the LIN synch break detection and flag.

Since the data framing error (FRE) flag bit in SSR generates a reception interrupt two bits earlier than a LIN break interrupt (for communication format is 8-bit data, no parity, "1" stop bit.), set the RXE to "0" when a LIN break is used.

The LIN synch break detection works only in operation mode 3.

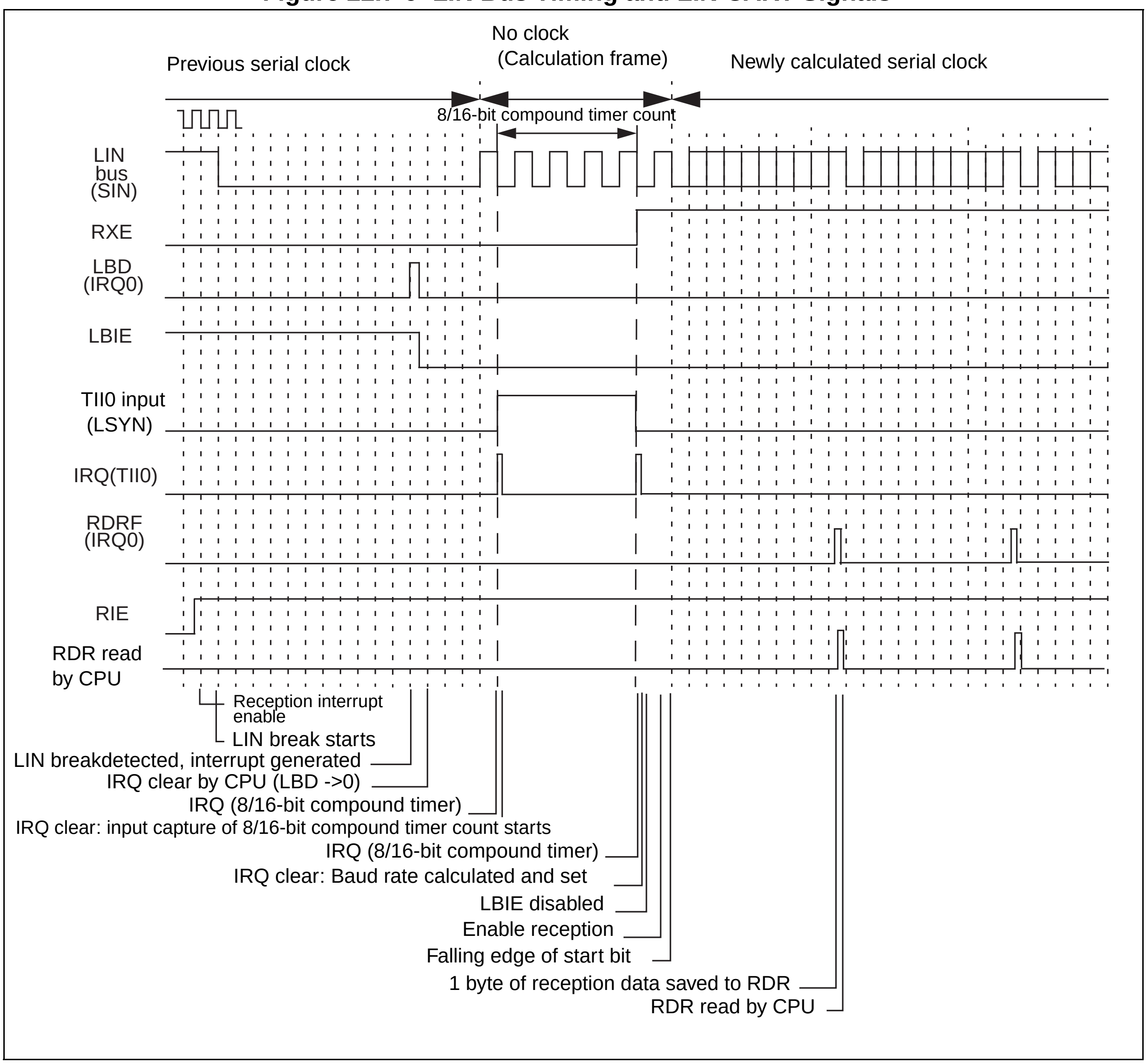
Figure 22.7-8 shows the LIN-UART operation in LIN slave modes.

Figure 22.7-8 LIN-UART Operation in LIN Slave Modes



● LIN bus timing

Figure 22.7-9 LIN Bus Timing and LIN-UART Signals



22.7.4 Serial Pin Direct Access

Transmission pin (SOT) or reception pin (SIN) can be accessed directly.

■ LIN-UART Pin Direct Access

The LIN-UART allows the programmer to directly access the serial I/O pins.

The status of the serial input pin (SIN) can be read by using the serial I/O pin direct access bit (ESCR:SIOP).

You can set the value of the serial output pin (SOT) arbitrarily when the serial output is enabled (SMR:SOE=1) after direct write to the serial output pin (SOT) is enabled (ESCR:SOPE = 1), and then "0" or "1" is written to the serial I/O pin direct access bit (ESCR:SIOP).

In LIN mode, this feature is used for reading transmitted data or for error handling when a LIN bus line signal is physically incorrect.

Note:

Direct access is allowed only when transmission is not in progress (the transmission shift register is empty).

Before enabling transmission (SMR:SOE = 1), write a value to the serial output pin direct access bit (ESCR:SIOP). This prevents a signal of an unexpected level from being output since the SIOP bit holds a previous value.

While the value of the SIN pin is read by normal read, the value of the SOT pin is read for the SIOP bit by the read-modify-write (RMW) instructions.

22.7.5 Bi-directional Communication Function (Normal Mode)

Normal serial bi-directional communication can be performed in operation mode 0 or 2. Asynchronous mode and synchronous mode can be selected in operation modes 0 and 2, respectively.

■ Bi-directional Communication Function

To operate the LIN-UART in normal mode (operation mode 0 or 2), the settings shown in Figure 22.7-10 are required.

Figure 22.7-10 Settings of LIN-UART Operation Modes 0 and 2

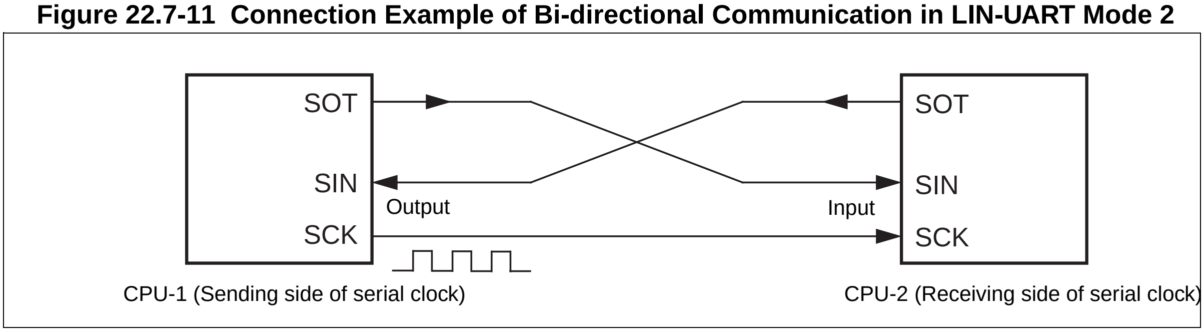
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 0)→	◎	◎	◎	◎	×	0	◎	◎	0	0	0	◎	0	0	◎	◎
(Mode 2)→	▣	▣	▣	+	×	0	◎	◎	1	0	◎	◎	0	0	◎	◎
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set comparison data (during writing) Retain reception data (during reading)							
(Mode 0)→	◎	◎	◎	◎	◎	◎	◎	◎								
(Mode 2)→	▣	◎	▣	◎	◎	◎	◎	◎								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 0)→	×	×	×	×	◎	◎	0	0	0	0	×	×	×	0	◎	◎
(Mode 2)→	×	×	×	×	◎	◎	▣	◎	0	×	◎	◎	◎	0	▣	▣

◎ : Used bit
× : Unused bit
1 : Set to "1"
0 : Set to "0"
▣ : Used when SSM = 1 (Synchronous star/stop bit mode)
\+ : Bit correctly set automatically

● Inter-CPU connection

For bi-directional communication, interconnect two CPUs as shown in Figure 22.7-11.

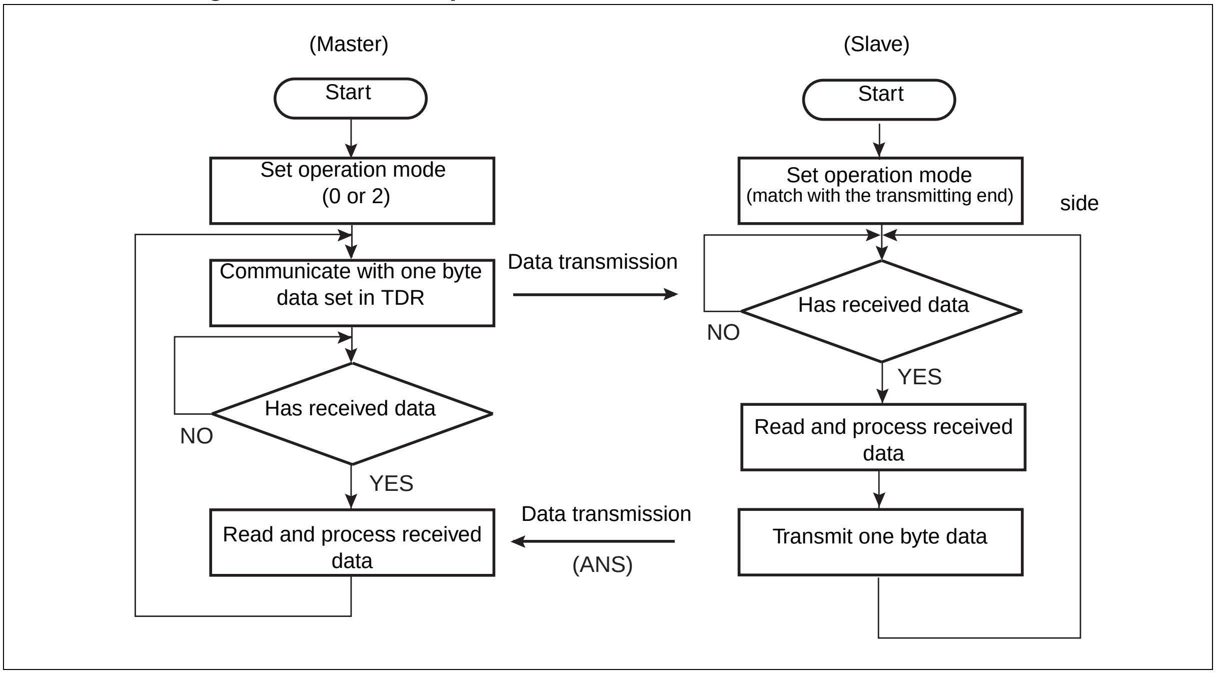
Figure 22.7-11 Connection Example of Bi-directional Communication in LIN-UART Mode 2



● Communication procedure example

The communication is started from transmitting end at arbitrary timing when data is ready to be transmitted. When transmission data is received in the receiving side, ANS (1 byte in the example) is returned on a regular basis. Figure 22.7-12 shows an example of bi-directional communication flowchart.

Figure 22.7-12 Example of Bi-directional Communication Flowchart



22.7.6 Master/slave Mode Communication Function (Multi-processor Mode)

Operation mode 1 allows communication between multiple CPUs connected in master/slave mode. It can be used as a master or slave.

■ Master/Slave Mode Communication Function

To operate the LIN-UART in multiprocessor mode (operation mode 1), the settings shown in Figure 22.7-13 are required.

Figure 22.7-13 Settings of LIN-UART Operation Mode 1

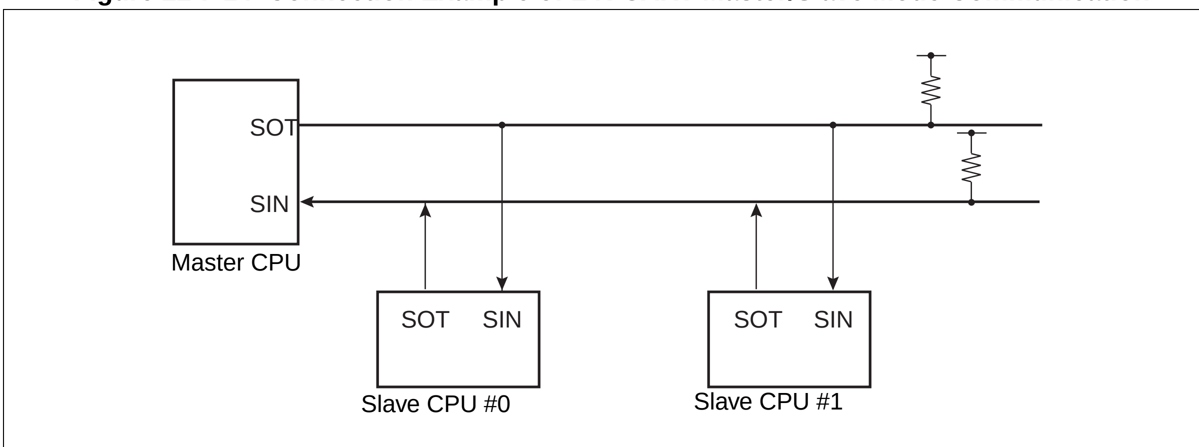
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 1)→	+	×	◎	◎	◎	0	◎	◎	0	1	0	◎	0	0	◎	◎
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)							
(Mode 1)→	×	◎	◎	◎	◎	◎	◎	◎								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 1)→	×	×	×	×	◎	◎	0	0	0	×	×	×	×	0	◎	◎

◎ : Used bit
× : Unused bit
1 : Set to "1"
0 : Set to "0"
+ : Bit correctly set automatically

● Inter-CPU Connection

For master/slave mode communication, a communication system is configured by connecting between one master CPU and multiple slave CPUs with two common communication lines, as shown in Figure 22.7-14. The LIN-UART can be used as the master or slave.

Figure 22.7-14 Connection Example of LIN-UART Master/Slave Mode Communication



● Function Selection

For master/slave mode communication, select the operation mode and the data transfer method, as shown in Table 22.7-4.

Table 22.7-4 Select of Master/Slave Mode Communication Function

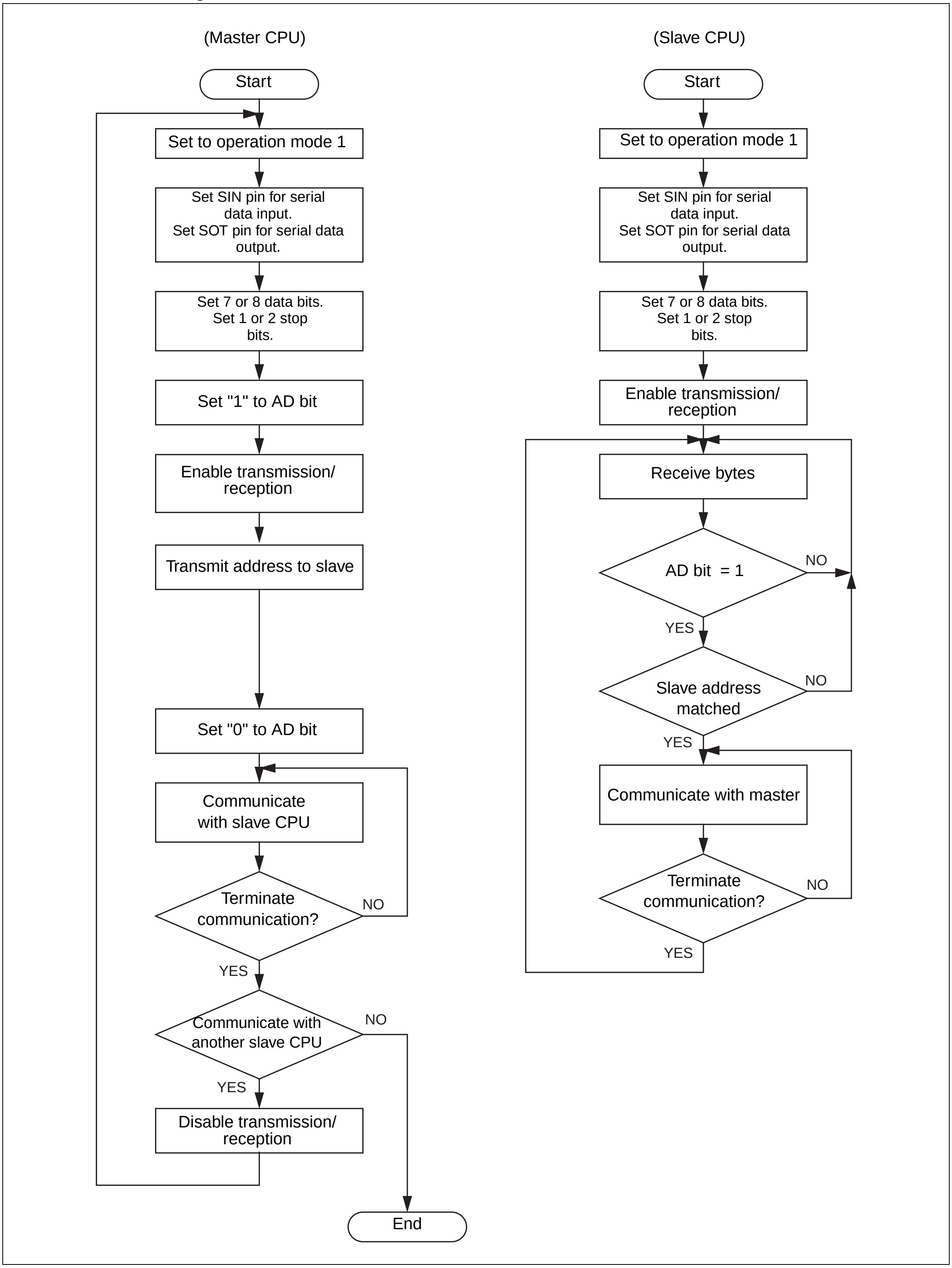
	Operation mode		Data	Parity	Synchronous method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission/ reception	Mode 1 (AD bit transmission and reception)	Mode 1 (AD bit transmission and reception)	AD = 1 + 7-bit or 8-bit address	None	Asynchronous	1 bit or 2 bits	LSB first or MSB first
Data transmission/ reception			AD = 0 + 7-bit or 8-bit data				

● Communication procedure

Communication is started by transmitting address data from the master CPU. The address data, whose AD bit is set as "1", determines the slave CPU to be the destination. Each slave CPU checks address data by using a program, and communicates with the master CPU when the data matches an assigned address.

Figure 22.7-15 shows a flowchart for master/slave mode communication.

Figure 22.7-15 Master/Slave Mode Communication Flowchart



22.7.7 LIN Communication Function

For LIN-UART communication, a LIN device can be used in the LIN master system or the LIN slave system.

■ LIN Master/Slave Mode Communication Function

Figure 22.7-16 shows the required settings for the LIN communication mode of LIN-UART (operation mode 3).

Figure 22.7-16 Settings of LIN-UART Operation Mode 3 (LIN)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 3)→	+	×	+	+	×	0	◎	◎	1	1	0	◎	0	0	◎	◎
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)							
(Mode 3)→	×	◎	◎	◎	◎	+	◎	◎								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 3)→	◎	◎	◎	◎	◎	◎	0	0	0	◎	×	×	×	0	◎	◎

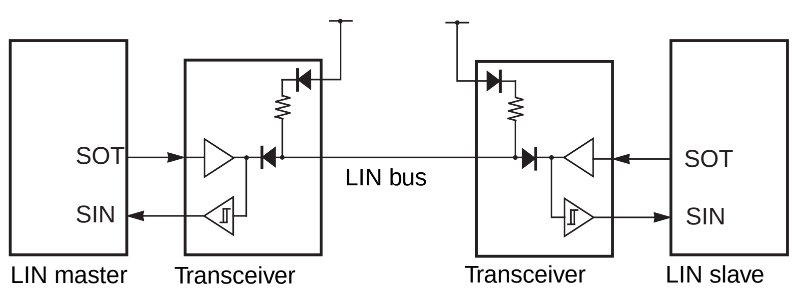
◎ : Used bit
× : Unused bit
1 : Set to "1"
0 : Set to "0"
\+ : Bit correctly set automatically

● LIN device connection

Figure 22.7-17 shows an example of the LIN bus system communication.

The LIN-UART can serve as the LIN master or LIN slave.

Figure 22.7-17 Example of LIN Bus System Communication

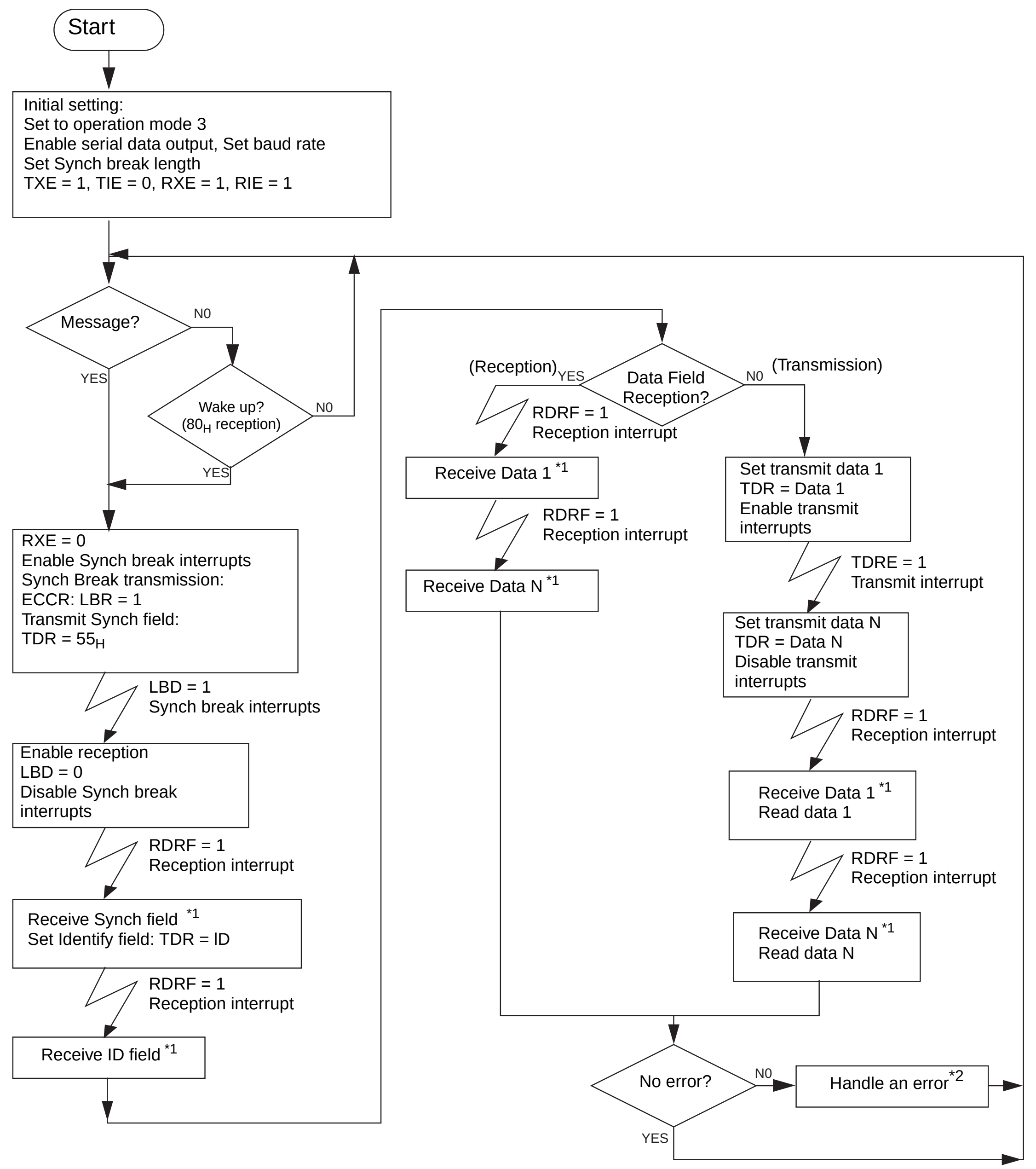


22.7.8 Example of LIN-UART LIN Communication Flowchart(Operation Mode 3)

This section shows examples of LIN-UART LIN communication flowchart.

■ LIN Master Device

Figure 22.7-18 LIN Master Flowchart



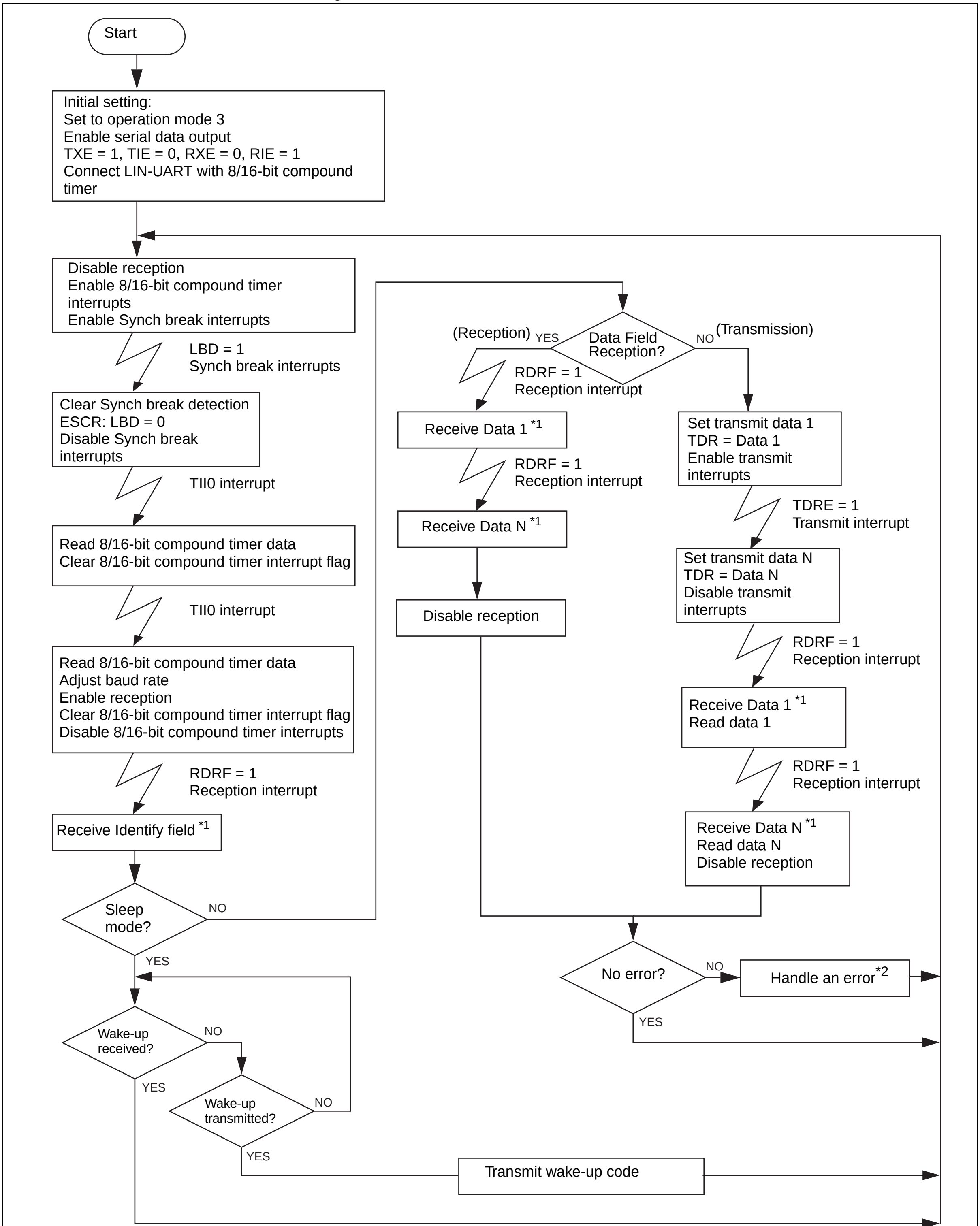
*1: Handle an error if it occurs.

*2:
- If the FRE or ORE flag is set to "1", write "1" to the SCR:CRE bit to clear the error flag.
- If the ESCR: LBD bit is set to "1", execute the LIN-UART reset.

Note: Detect an error in each process and handle it appropriately.

■ LIN Slave Device

Figure 22.7-19 LIN Slave Flowchart



*1: Handle an error if it occurs.

*2:
- If the FRE or ORE flag is set to "1", write "1" to the SCR:CRE bit to clear the error flag.
- If the ESCR:LBD bit is set to "1", execute the LIN-UART reset.

Note: Detect an error in each process and handle it appropriately.

22.8 Notes on Using LIN-UART

This section shows notes on using the LIN-UART.

■ Notes on Using LIN-UART

● Enabling operation

The LIN-UART has the TXE (transmission) and RXE (reception) enable bit in the LIN-UART serial control register (SCR) for transmission and reception, respectively. Since both transmission and reception are disabled by default (internal value), these operations must be enabled before transfer. Also, you can disable these operations to stop transfer as required.

● Setting communication mode

The communication mode must be set while the LIN-UART is stopped. If the mode is set during transmission/reception, the transmitted/received data is not guaranteed.

● Timing of enabling transmit interrupts

Since the default (initial) value of the transmit data empty flag bit (SSR:TDRE) is "1" (no transmit data, transmit data write enabled), a transmit interrupt request is generated immediately when transmit interrupt request is enabled (SSR:TIE =1). To prevent this, be sure to set the transmit data before setting the TIE flag to "1".

● Changing operation setting

Reset the LIN-UART after changing its settings, such as adding the start/stop bit or changing the data format.

The correct operation settings are not guaranteed even if you reset the LIN-UART (SMR:UPCL = 1) concurrently with setting the LIN-UART serial mode register (SMR). Therefore, after setting the bit in LIN-UART serial mode register (SMR), reset the LIN-UART (SMR:UPCL = 1) again.

● Using LIN function

Although the LIN functions are available in the mode 3, the LIN format is automatically set in the mode 3 (8-bit data, no parity, 1 stop bit, LSB-first).

While the length of LIN break transmit bit is variable, the detection bit length is fixed to 11 bits.

● Setting LIN slave

When starting LIN slave mode, be sure to set the baud rate before receiving the LIN synch break in order to make sure that the minimum 13 bits length of the LIN synch break is detected.

● Bus idle function

The bus idle is not available in synchronous mode 2.

● AD bit (LIN-UART serial control register (SCR): Address/data format selection bit)

Be sure to note the followings when using the AD bit.

The AD bit is used to select the address/data for transmission when it is written, and to read the AD bit received last when it reads. Internally, the AD bit values for transmission and reception are stored in separate registers.

The transmit AD bit value is read when read-modify-write (RMW) instructions are used. Therefore, an incorrect value may be written to the AD bit when another bit in the SCR is bit-accessed.

For the above reason, the AD bit must be set at the last access to the SCR before transmission. Or, the above problem can be prevented by byte-accessing whenever the SCR is written.

● LIN-UART software reset

Execute the LIN-UART software reset (SMR:UPCL = 1) when the TXE bit in the LIN-UART serial control register (SCR) is "0".

● Synch break detection

In mode 3 (LIN mode), when serial input has 11 bits width or more and becomes "L", the LBD bit in the extended status control register (ESCR) is set to "1" (Synch break detection) and the LIN-UART waits for the Synch field. As a result, when serial input has more than 11 bits of "0" except Synch break, the LIN-UART recognizes that the Synch break is input (LBD = 1), and then waits for the Synch field.

In this case, execute the LIN-UART reset (SMR:UPCL = 1).

● Handling framing errors

1) (CRE resets reception state machine and next falling edge at SINn starts reception of new byte (Figure 22.8-1).
In order to avoid desynchronization of the data stream, it is necessary to set the CRE bit within a half-bit time immediately after an error is received (as shown in Figure 22.8-2), or to wait for the application-dependent time while SINn is idling after an error is received.

2) If a framing error occurs (stop bit: SINn= "0") and the next start bit (SINn= "0") immediately follows it, this start bit is recognized regardless of a falling edge for the start bit and reception is started. This sequence is used for detecting the continuous "L" state of the serial data input (SINn) when the next framing error is detected while the data stream is synchronized (See "When reception is always enabled (RXE=1)" in Figure 22.8-3).
If this operation is not necessary, disable data reception temporarily after receiving a framing error (RXE = 1 → 0 → 1). Therefore, the falling edge of the serial data input (SINn) is detected, the start bit is recognized when "L" is detected at the reception sampling point, and the reception is started (See "When reception is temporarily disabled (RXE=1→0→1)" in Figure 22.8-3).

Figure 22.8-1 CRE bit timing

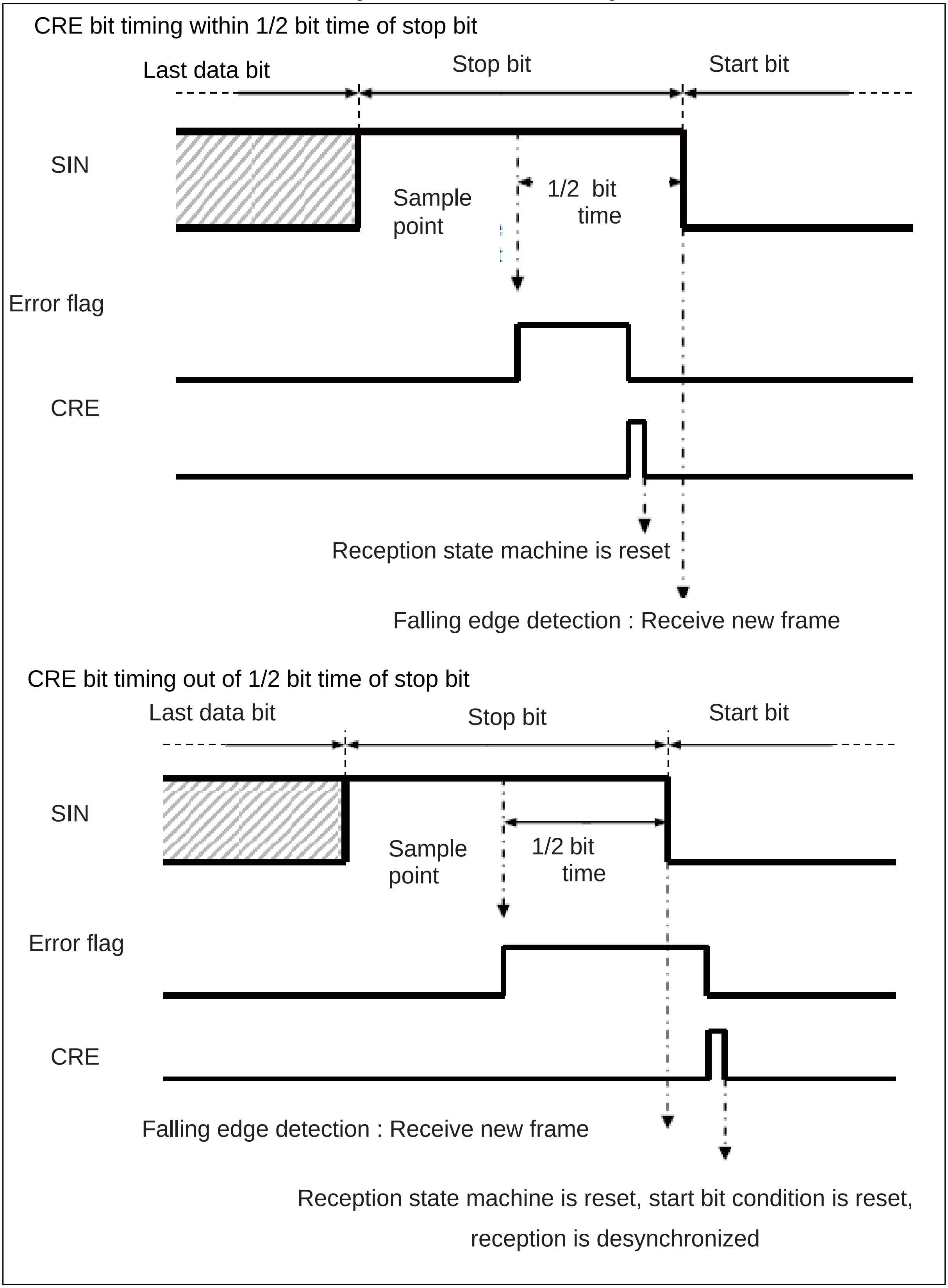


Figure 22.8-2 Example of desynchronization

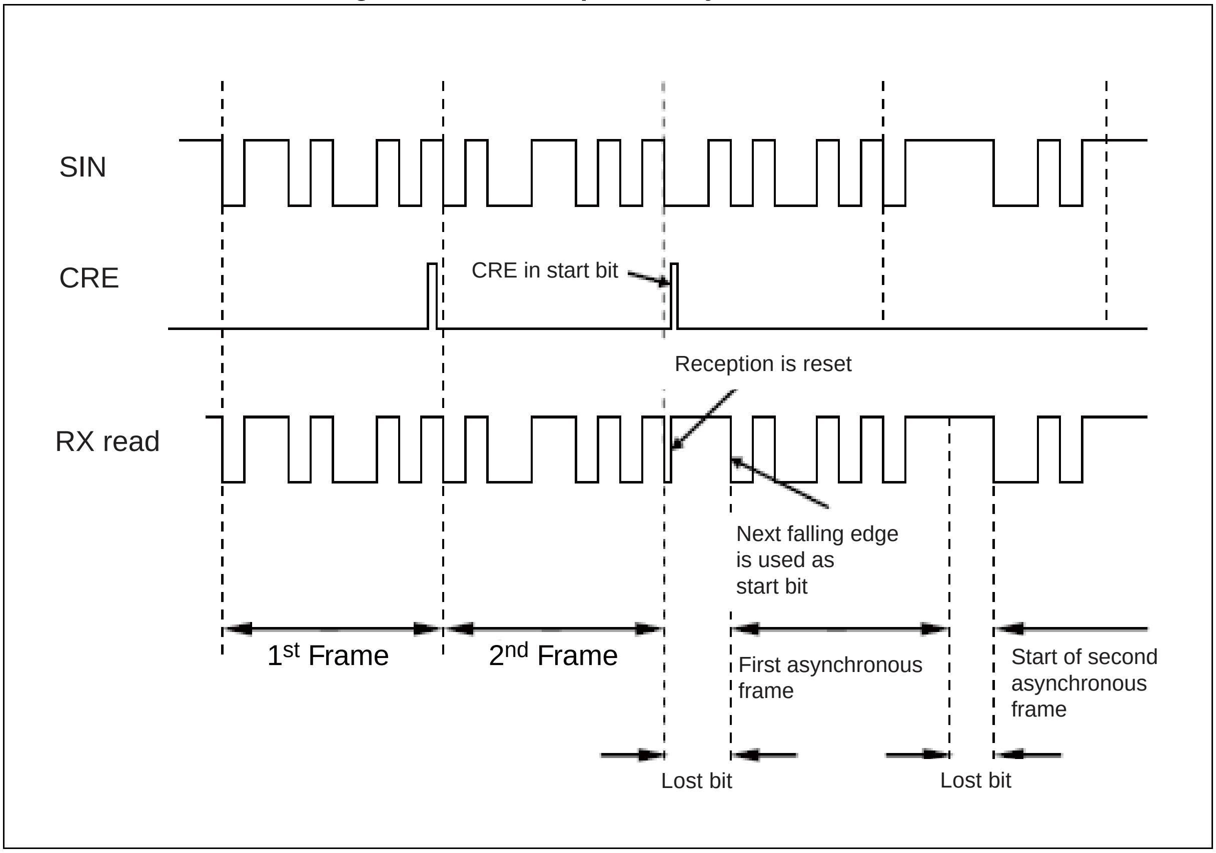
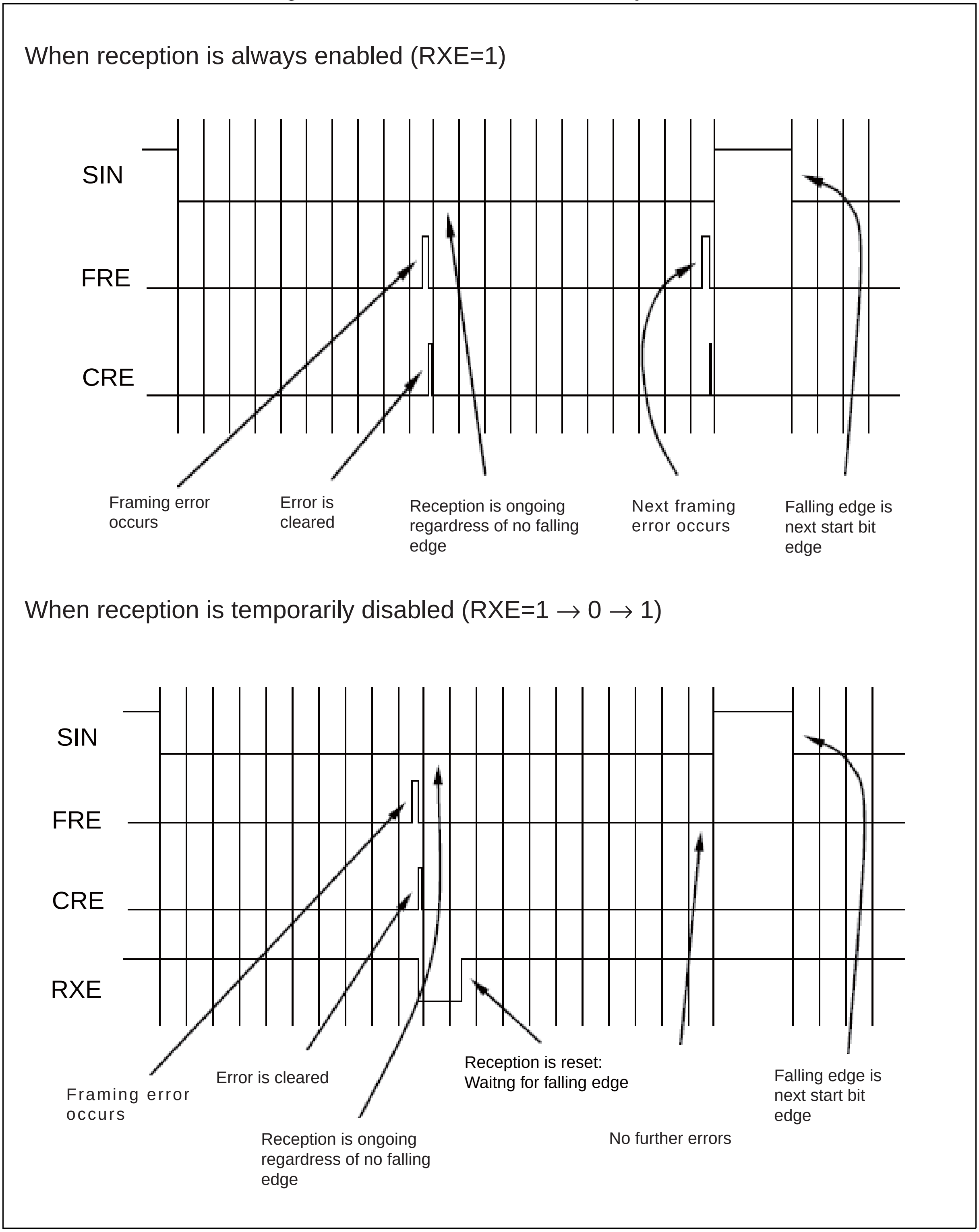


Figure 22.8-3 UART dominant bus operation



22.9 Sample Programs of LIN-UART

This section provides sample programs for operating the LIN-UART.

■ Sample Programs of LIN-UART

For sample programs of LIN-UART, see "■Sample Programs" in Preface.

■ Setting Methods not Covered by Sample Programs

● How to select the operation mode

Use the operation mode selection (SMR.MD[1:0]).

Operation mode		Operation mode selection (MD[1:0]).
Mode 0	Normal (Asynchronous)	Set to "00_B"
Mode 1	Multiprocessor	Set to "01_B"
Mode 2	Normal (Synchronous)	Set to "10_B"
Mode 3	LIN	Set to "11_B"

● Operation clock types and how to select it

Use the external clock select bit (SMR.EXT).

Clock input	External clock select bit (EXT)
To select a dedicated baud rate generator	Set to "0"
To select an external clock	Set to "1"

● How to control the SCK, SIN, and SOT pins

Use the following setting.

Operation	LIN-UART
To set the SCK pin as input	DDR6:P65 =0 SMR:SCKE =0
To set the SCK pin as output	SMR:SCKE =1
To use the SIN pin	DDR6:P67 =0
To use the SOT pin	SMR:SOE =1

● How to enable/disable the LIN-UART operation

Use the reception enable bit (SCR.RXE).

Control item	Reception enable bit (RXE)
Disable reception	Set to "0"
Enable reception	Set to "1"

Use the transmit control bit (SCR.TXE).

Control item	Transmit control bit (TXE)
Disable transmission	Set to "0"
Enable transmission	Set to "1"

● How to use an external clock as the LIN-UART serial clock

Use the one-to-one external clock enable bit (SMR.OTO).

Control item	Reception enable bit (OTO)
Enable external clock	Set to "1"

● How to restart the reload counter

Use the reload counter restart bit (SMR.REST).

Control item	Reload counter restart bit (REST)
Restart the reload counter	Set to "1"

● How to reset the LIN-UART

Use the LIN-UART programmable clear bit (SMR:UPCL).

Control item	LIN-UART programmable clear bit (UPCL)
Reset the LIN-UART software	Set to "1"

● How to set the parity

Use the parity enable bit (SCR.PEN) and the parity select bit (SCR.P).

Operation	Parity control (PEN)	Parity polarity (P)
To set to no parity	Set to "0"	–
To set to even parity	Set to "1"	Set to "0"
To set to odd parity	Set to "1"	Set to "1"

● How to set the data length

Use the data length select bit (SCR.CL).

Operation	Data length select bit (CL)
To set the bit length to 7	Set to "0"
To set the bit length to 8	Set to "1"

● How to select the STOP bit length

Use the STOP bit length control (SCR.SBL).

Operation	STOP bit length control (SBL)
To set STOP bit length to 1	Set to "0"
To set STOP bit length to 2	Set to "1"

● How to clear the error flag

Use the reception error flag clear bit (SCR.CRE).

Control item	Reception error flag clear bit (CRE)
To clear the error flag (PE, ORE, FRE)	Set to "0"

● How to set the transfer direction

Use the transfer direction selection bit (SSR.BDS).

LSB/MSB can be selected for transfer direction in any operation mode.

Control item	Serial data direction control (BDS)
To select the LSB first transfer (from the least significant bit)	Set to "0"
To select the MSB first transfer (from the most significant bit)	Set to "1"

● How to clear the reception completion flag

Uses the following setting.

Control item	Method
To clear the reception completion flag	Read the RDR register

The first RDR register read is the reception initiation.

● How to clear the transmit buffer empty flag

Uses the following setting.

Control item	Method
To clear the transmit buffer empty flag	Write to TDR register

The first TDR register write is the transmit initiation.

● How to select the data format (Address/Data) (Only in mode 1)

Use the address/data selection bit (SCR:AD).

Operation	Address/Data select bit (AD)
To select the data frame	Set to "0"
To select the address frame	Set to "1"

This is effective only at transmission. The AD bit is ignored at reception.

● How to set the baud rate

See Section "22.6 LIN-UART Baud Rate".

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

	Interrupt level setting register	Interrupt vector
Reception	Interrupt level register (ILR1) Address: $0007A_H$	#7 Address: $0FFFC_H$
Transmission	Interrupt level register (ILR2) Address: $0007B_H$	#8 Address: $0FFEA_H$

● How to enable/disable/clear interrupts

Interrupt request enable bit (SSR.RIE), (SSR.TIE) is used to enable interrupts.

Operation	UART reception	UART transmission
	Reception interrupt enable bit (RIE)	Reception interrupt enable bit (TIE)
To disable interrupt requests	Set to "0"	
To enable interrupt requests	Set to "1"	

The following setting is used to clear interrupt requests.

Operation	UART reception	UART transmission
To clear interrupt requests	The reception data register full (RDRF) is cleared by reading the LIN-UART serial input register (RDR).	The transmit data register empty (TDRE) is set to "0" by writing data to the serial output data register (TDR).
	The error flags (PE, ORE, FRE) are set to "0" by writing "1" to the error flag clear bit (CRE).	

CHAPTER 23
8/10-BIT A/D CONVERTER

This chapter describes the functions and operations of the 8/10-bit A/D converter.

Code: CM26-00125-2E

Page: 445, 447, 448, 450, 451, 458, 459, 461

23.1 Overview of 8/10-bit A/D Converter

The 8/10-bit A/D converter is a 10-bit successive approximation type of 8/10-bit A/D converter. It can be started via software, external trigger, and internal clock, with one input signal selected from among multiple analog input pins.

■ A/D Conversion Functions

The A/D converter converts analog voltages (input voltages) input to an analog input pin to 10-bit digital values.

- One of multiple analog input pins can be selected.
- The conversion speed is programmable to be configured (selected according to the operating voltage and frequency).
- An interrupt is generated when A/D conversion completes.
- The completion of conversion can also be checked with the ADI bit in the ADC1 register.

To activate A/D conversion functions, follow one of the methods given below.

- Activation using the AD bit in the ADC1 register
- Continuous activation using the external pin (ADTG)
- Continuous activation using the 8/16-bit compound timer output TO00

23.2 Configuration of 8/10-bit A/D Converter

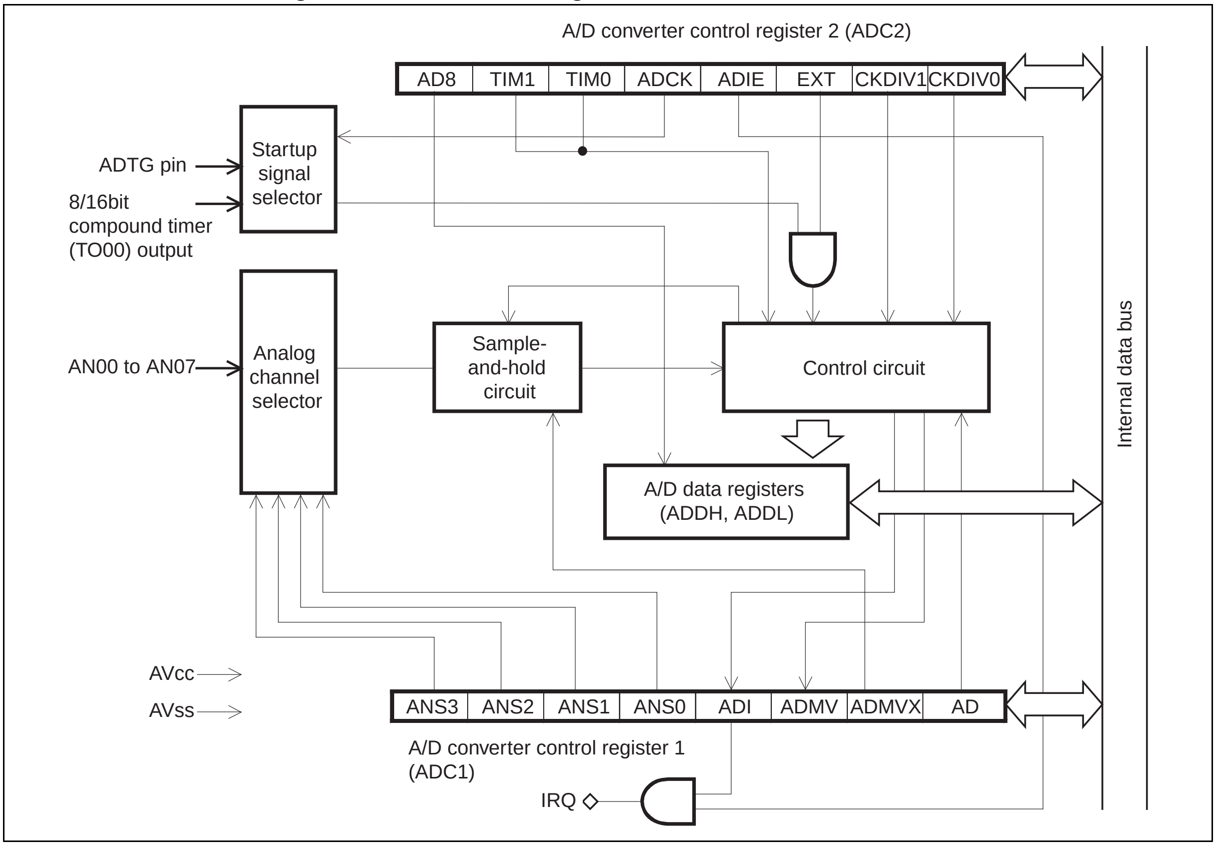
The 8/10-bit A/D converter consists of the following blocks:

- **Clock selector (input clock selector for starting A/D conversion)**
- **Analog channel selector**
- **Sample-and-hold circuit**
- **Control circuit**
- **A/D converter data registers (ADDH, ADDL)**
- **A/D converter control register 1 (ADC1)**
- **A/D converter control register 2 (ADC2)**

■ Block Diagram of 8/10-bit A/D Converter

Figure 23.2-1 shows a block diagram of the 8/10-bit A/D converter.

Figure 23.2-1 Block Diagram of 8/10-bit A/D Converter



● Clock selector

This block selects the A/D conversion clock with continuous activation enabled (ADC2:EXT = 1).

● Analog channel selector

This circuit selects one of multiple analog input pins.

● Sample-and-hold circuit

This circuit holds the input voltage selected by the analog channel selector. This enables A/D conversion to be performed without being affected by variation in input voltage during conversion (comparison) by sampling and holding the input voltage immediately after starting A/D conversion.

● Control circuit

The A/D conversion function determines the values in the 10-bit A/D converter data register sequentially from MSB to LSB based on the signals from the comparator. When conversion is completed, the A/D conversion function sets the interrupt request flag bit (ADC1: ADI).

● A/D converter data registers (ADDH/ADDL)

The high-order two bits of 10-bit A/D data are stored in the ADDH register; the low-order eight bits are stored in the ADDL register.

Setting the A/D conversion precision bit (ADC2:AD8) to "1" provides 8-bit precision, storing the upper eight bits of the 10-bit A/D data in the ADDL register.

● A/D converter control register 1 (ADC1)

This register is used to enable and disable functions, select an analog input pin, check statuses, and control interrupts.

● A/D converter control register 2 (ADC2)

This register is used to select an input clock, enable and disable interrupts, and select functions.

■ Input Clock

The 8/10-bit A/D converter uses the output clock from the prescaler as the input clock (operation clock).

23.3 Pins of 8/10-bit A/D Converter

This section describes the pins of the 8/10-bit A/D converter.

■ Pins of 8/10-bit A/D Converter

MB95150/M series has 8 channels of analog input pin.

Analog input pins also serve as general-purpose I/O ports.

● AN07 to AN00 Pins

AN07 to AN00 : When using the A/D conversion function, input the analog voltage you wish to convert to one of these pins. Each of the pins serves as an analog input pin by selecting it using the analog input channel select bits (ADC1: ANS0 to ANS3) with the corresponding bit in the port direction register (DDR) set to "0". Even when the 8/10-bit A/D converter is used, the pins not used for analog input can be used as general-purpose I/O ports.
Note that the number of analog input pins differs depending on the series.

● ADTG Pin

ADTG : This is a pin used to activate A/D conversion function by external trigger.

● AVcc pin

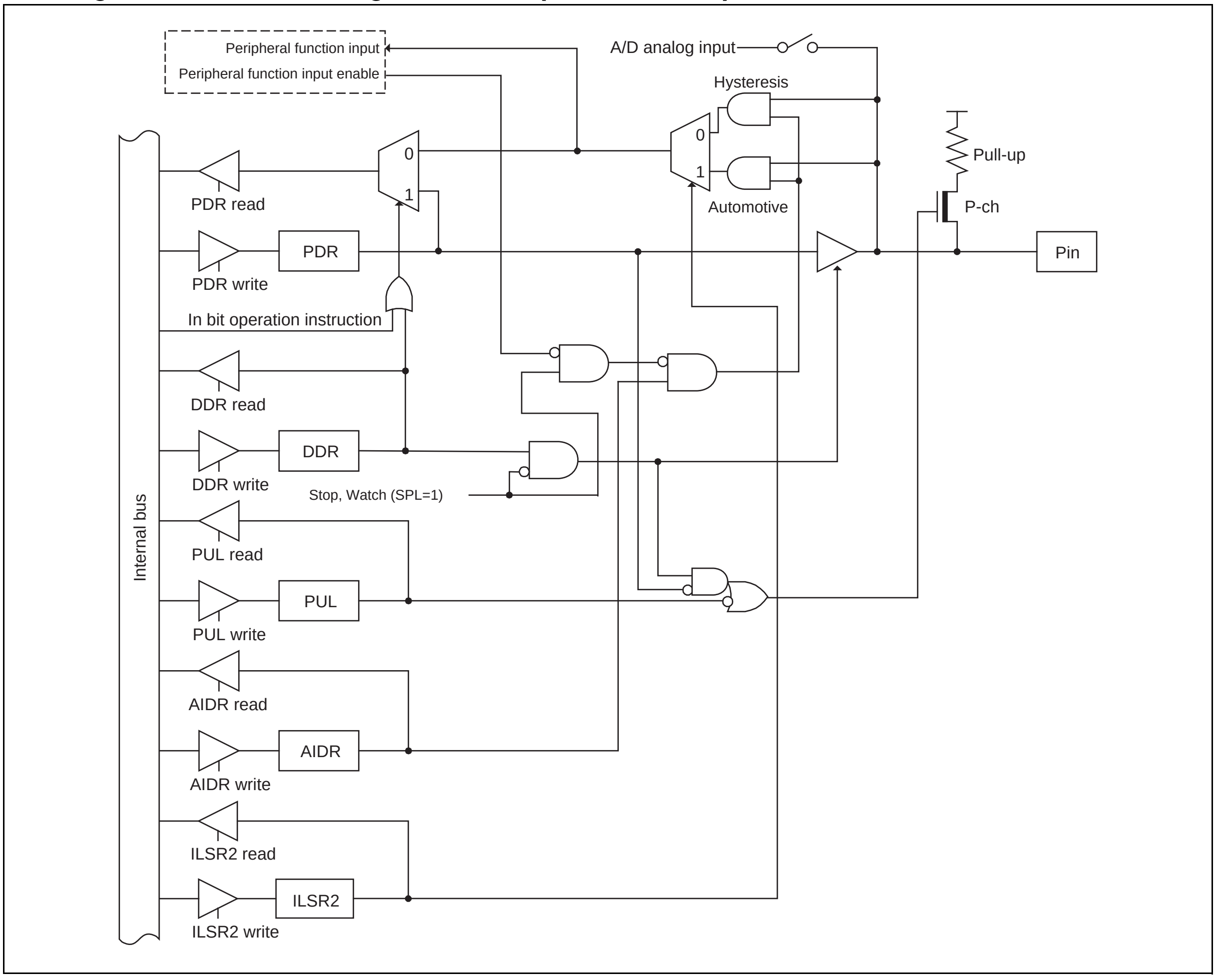
AV_{CC} : This is a 8/10-bit A/D converter power supply pin. Use this at the same potential as V_{CC}. If A/D conversion precision is demanded, you should take measures to ensure that VCC noise does not enter AV_{CC}, or use a separate power source. You should connect this pin to a power source even when the 8/10-bit A/D converter is not being used.

● AVss pin

AV_{SS} : This is a ground pin of the 8/10-bit A/D converter. Use this at the same potential as V_{SS}. When A/D conversion precision is required, take measures to ensure that the VSS noise does not interfere with AV_{SS}. You should connect this pin to a ground (GND) even when the 8/10-bit A/D converter is not being used.

■ Block Diagram of Pins Related to 8/10-bit A/D Converter

Figure 23.3-1 Block Diagram of Pins (AN00 to AN07) Related to 8/10-bit A/D Converter



23.4 Registers of 8/10-bit A/D Converter

The 8/10-bit A/D converter has four registers: A/D converter control register 1 (ADC1), A/D converter control register 2 (ADC2), A/D converter data register upper (ADDH), and A/D converter data register lower (ADDL).

■ List of 8/10-bit A/D Converter Registers

Figure 23.4-1 lists the registers of the 8/10-bit A/D converter.

Figure 23.4-1 Registers of 8/10-bit A/D Converter

8/10-bit A/D converter control register 1 (ADC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$006C_H$	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD	00000000_B
	R/W	R/W	R/W	R/W	R(RM1),W	R/WX	R/W	R0,W	

8/10-bit A/D converter control register 2 (ADC2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$006D_H$	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/10-bit A/D converter data register upper (ADDH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$006E_H$	–	–	–	–	–	–	SAR9	SAR8	00000000_B
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	

8/10-bit A/D converter data register lower (ADDL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$006F_H$	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1	SAR0	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W : Readable/writable (Read value is the same as write value)
R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R0, W : Write only (Writable, "0" is read)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

23.4.1 8/10-bit A/D Converter Control Register 1 (ADC1)

8/10-bit A/D converter control register 1 (ADC1) is used to enable and disable individual functions of the 8/10-bit A/D converter, select an analog input pin, and to check the states.

■ 8/10-bit A/D Converter Control Register 1 (ADC1)

Figure 23.4-2 8/10-bit A/D Converter Control Register 1 (ADC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006C$_H$	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD	00000000$_B$
	R/W	R/W	R/W	R/W	R(RM1),W	R/WX	R/W	R0,W	

AD	A/D conversion start bit
0	Do not start A/D conversion.
1	Start A/D conversion.

ADMVX	Current cut-off analog switch control bit
0	Turn on analog switch only during conversion.
1	Maintain analog switch on.

ADMV	Conversion flag bit
0	Not converting
1	Currently converting

ADI	Interrupt request flag bit	
	Read	Write
0	Conversion not completed	Clear this bit.
1	Conversion completed	Make no changes to the bit with no effect on others.

ANS3	ANS2	ANS1	ANS0	Analog input channel select bits
0	0	0	0	AN00 pin
0	0	0	1	AN01 pin
0	0	1	0	AN02 pin
0	0	1	1	AN03 pin
0	1	0	0	AN04 pin
0	1	0	1	AN05 pin
0	1	1	0	AN06 pin
0	1	1	1	AN07 pin

R/W : Readable/Writable (Read value is the same as write value)
R/WX : Read only (Readable, writing has no effect on operation)
R0,W : Write only (Writable, "0" is read)
R(RM1),W : Readable/Writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
(shaded) : Initial value

Do not select the unusable channel for this series by analog input channel select bits (ANS3 to ANS0).

Table 23.4.-1 Functions of Bits in 8/10-bit A/D Converter Control Register 1 (ADC1)

Bit name		Function
bit7 to bit4	ANS3, ANS2, ANS1, ANS0: Analog input channel select bits	Select the analog input pin to be used from among AN00 to AN07. Note that the number of analog input pins differs depending on the series. When A/D conversion is activated (AD = 1) via software (ADC2: EXT = 0), these bits can be updated at the same time. Note: When the ADMV bit is "1", do not update these bits. The pins not used as analog input pins can be used as general-purpose ports.
bit3	ADI: Interrupt request flag bit	Detects the termination of A/D conversion. • When the A/D conversion function is used, the bit is set "1" upon termination of A/D conversion. • Interrupt requests are output when this bit and the interrupt request enable bit (ADC2: ADIE) are both "1". • When written to this bit, "0" clears it; "1" leaves it unchanged with no affect on others. • When read by a read-modify-write (RMW) instruction, the bit returns "1".
bit2	ADMV: Conversion flag bit	Indicates that conversion is ongoing during execution of the A/D conversion function. The bit contains "1" during conversion. This bit is read only. Any value attempted to be written is meaningless and has no effect on operation.
bit1	ADMVX: Analog switch for shutting down control bit	Controls the analog switch for shutting down the internal reference power supply. When the external impedance of the AVR pin is high, rush current flows immediately after A/D startup and may affect A/D conversion precision. In this kind of situation, this can be avoided by setting this bit to "1" before A/D startup. Set the bit to "0" before switching to standby mode, in order to reduce current consumption. Note that some series do not have AVR pins, and are internally connected to AV_{CC}.
bit0	AD: A/D conversion startup bit	Starts the A/D conversion function via software. Writing "1" to the bit starts the A/D conversion function. Note: Writing "0" to this bit will not stop operation of the A/D conversion function. The value read is always "0". A/D conversion startup by this bit is disabled with EXT=1. A/D converter re-starts by writing "1" to this bit during A/D conversion with EXT = 0.

23.4.2 8/10-bit A/D Converter Control Register 2 (ADC2)

8/10-bit A/D converter control register 2 (ADC2) selects the 8/10-bit A/D converter function, selects the input clock, and performs interrupt and status checking.

■ 8/10-bit A/D Converter Control Register 2 (ADC2)

Figure 23.4-3 8/10-bit A/D Converter Control Register 2 (ADC2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006DH	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CKDIV1	CKDIV0	Clock (CKIN) select bits
0	0	1 MCLK
0	1	2 MCLK
1	0	4 MCLK
1	1	8 MCLK

EXT	Continuous activation enable bit
0	Start using the AD bit in the ADC1 register
1	Continuous activation with the clock selected by the ADCK bit in the ADC2 register

ADIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enable interrupt request output.

ADCK	External start signal select bit
0	Start via ADTG input pin
1	Start via 8/16-bit compound timer (TO00) output

TIM1	TIM0	Sampling time select bits
0	0	CKIN × 4
0	1	CKIN × 7
1	0	CKIN × 10
1	1	CKIN × 16

AD8	Precision select bit
0	10-bit precision
1	8-bit precision

R/W :Readable/writable (Read value is the same as write value)
MCLK :Machine clock
(shaded) :Initial value

Table 23.4.-2 Functions of Bits in 8/10-bit A/D Converter Control Register 2 (ADC2)

Bit name		Function
bit7	AD8: Precision select bit	This bit selects the resolution of A/D conversion. **When set to "0":** The bit selects 10-bit precision. **When set to "1":** The bit selects 8-bit precision, in which case eight bits of data can be read from the ADDL register. Note: The data bits used are different depending on the resolution. Update this bit only with A/D operation stopped before starting conversion.
bit6, bit5	TIM1, TIM0: Sampling time select bits	Set the sampling time. • Change this sampling time setting depending on the operating conditions (voltage and frequency). • The CKIN value is determined by the clock select bits (ADC2:CKDIV1, DKDIV0). Note:Update this bit only with A/D operation stopped.
bit4	ADCK: External start signal select bit	Selects the start signal for external start (ADC2:EXT = 1).
bit3	ADIE: Interrupt request enable bit	Enables or disables output of interrupts to the interrupt controller. Interrupt requests are output with both of this bit and the interrupt request flag bit (ADC1: ADI) set to "1".
bit2	EXT: Continuous activation enable bit	Selects whether to activate the A/D conversion function via software, or continuously upon detection of the rise of the input clock signal.
bit1, bit0	CKDIV1, CKDIV0: Clock select bits	Select the clock to use for A/D conversion. The input clock is generated by the prescaler. See "CHAPTER 6 CLOCK CONTROLLER". • The sampling time can also be changed via this clock selection. • Change this setting depending on the operating conditions (voltage and frequency). Note:Update this bit only with A/D operation stopped.

23.4.3 8/10-bit A/D Converter Data Registers Upper/ Lower (ADDH, ADDL)

The 8/10-bit A/D converter data registers upper/lower (ADDH, ADDL) contain the results of 10-bit A/D conversion.
The high-order two bits of 10-bit data correspond to the ADDH register; the low-order eight bits correspond to the ADDL register.

■ 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)

Figure 23.4-4 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)

ADDH	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	–	–	–	–	–	–	SAR9	SAR8	00000000_B
$006E_H$	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	

ADDL	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
Address	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1	SAR0	00000000_B
$006F_H$	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX : Read only (Readable, writing has no effect on operation)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

The upper two bits of 10-bit A/D data correspond to bits1 and 0 in the ADDH register; the lower eight bits correspond to bit15 to bit8 in the ADDL register.

Set the AD8 bit in the ADC2 register to "1" to select 8-bit precision mode, so that 8-bit data can be read from the ADDL register.

These registers are read-only. Writing has no effect on the operation.

During 8-bit conversion, SAR8 and SAR9 hold "0".

● A/D Conversion Functions

When A/D conversion is started, the results of conversion are finalized and stored in these registers after the conversion time according to the register settings has passed. After A/D conversion finishes, therefore, read the A/D data registers (conversion results), write "0" to the ADI bit (bit3) in the ADC1 register before the next A/D conversion terminates, then after A/D conversion finishes, clear the flag. During A/D conversion, the registers contain the values resulting from the last conversion performed.

23.5 Interrupts of 8/10-bit A/D Converter

An interrupt source of the 8/10-bit A/D converter is Completion of conversion when A/D conversion functions are operating.

■ Interrupts During 8/10-bit A/D Converter Operation

When A/D conversion is completed, the interrupt request flag bit (ADC1: ADI) is set to "1". Then if the interrupt request enable bit is enabled (ADC2: ADIE = 1), an interrupt request is issued to the interrupt controller. Write "0" to the ADI bit using the interrupt service routine to clear the interrupt request.

The ADI bit is set when A/D conversion is completed, irrespective of the value of the ADIE bit.

The CPU cannot return from interrupt processing if the interrupt request flag bit (ADC1: ADI) is 1 with interrupt requests enabled (ADC2: ADIE = 1). Be sure to clear the ADI bit within the interrupt service routine.

■ Register and Vector Table Related to 8/10-bit A/D Converter Interrupts

Table 23.5-1 Register and Vector Table Related to 8/10-bit A/D Converter Interrupts

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
8/10-bit A/D	IRQ18	ILR4	L18	$FFD6_H$	$FFD7_H$

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

23.6 Operations of 8/10-bit A/D Converter and Its Setup Procedure Examples

The EXT bit in the ADC1 register can be used to select the software activation or continuous activation of the 8/10-bit A/D converter.

■ Operations of 8/10-bit A/D Converter's Conversion Function

● Software activation

The settings shown in Figure 23.6-1 are required for software activation of the A/D conversion function.

Figure 23.6-1 Settings for A/D Conversion Function (Software Activation)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD
	◎	◎	◎	◎	◎	◎	◎	1
ADC2	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0
	◎	◎	◎	×	◎	0	◎	◎
ADDH	–	–	–	–	–	–	A/D converted value retained	
ADDL	A/D converted value is retained.							

◎: Used bit
x: Unused bit
0 : Set to "0"
1 : Set to "1"

When A/D conversion is activated, the A/D conversion function starts working. In addition, even during conversion, the A/D conversion function can be reactivated.

● Continuous activation

The settings shown in Figure 23.6-2 are required for continuous activation of the A/D conversion function.

Figure 23.6-2 Settings for A/D Conversion Function (Continuous Activation)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD
	◎	◎	◎	◎	◎	◎	◎	×
ADC2	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0
	◎	◎	◎	◎	◎	1	◎	◎
ADDH	–	–	–	–	–	–	A/D converted value retained	

◎: Used bit
x: Unused bit
1 : Set to "1"

When continuous activation is enabled, A/D conversion is activated at the rising edge of the selected input clock to start the A/D conversion function. Continuous activation is stopped by disabling it (ADC2:EXT = 0).

■ Operations of A/D Conversion Function

This section details the operations of the 8/10-bit A/D converter.

1) When A/D conversion is started, the conversion flag bit is set (ADC1:ADMV = 1) and the selected analog input pin is connected to the sample-and-hold circuit.

2) The voltage at the analog input pin is loaded into the sample-and-hold capacitor in the sample-and-hold circuit during the sampling cycle. This voltage is held until A/D conversion has been completed.

3) The comparator in the control circuit compares the voltage loaded into the sample-and-hold capacitor with the A/D conversion reference voltage, from the most significant bit (MSB) to the least significant bit (LSB), and then sends the results to the ADDH and ADDL registers. After the results have been completely transferred, the conversion flag bit is cleared (ADC1:ADMV = 0) and the interrupt request flag bit is set (ADC1:ADI = 1).

Notes:

- When the A/D conversion function is used, the contents of the ADDH and ADDL registers are retained upon completion of A/D conversion. During A/D conversion, the values resulting from the last conversion are loaded.
- Do not re-select the analog input channel (ADC1: ANS3 to ANS0) while the A/D conversion function is running, in particular, during continuous activation. Disable continuous activation (ADC2: EXT = 0) before re-selecting the analog input channel.
- Starting the reset, stop, or watch mode stops the 8/10-bit A/D converter and initializes each register.

■ Setup Procedure Example

Follow the procedure below to set up the 8/10-bit A/D converter:

● Initial setting

1) Set the port for input (DDR0).
2) Set the interrupt level (ILR4).
3) Enable A/D input (ADC1:ANS0 to ANS3).
4) Set the sampling time (ADC2:TIM1, TIM0).
5) Select the clock (ADC2:CKDIV1, CKDIV0).
6) Set A/D conversion properties (ADC2:AD8).
7) Select the operation mode (ADC2:EXT).
8) Select the startup trigger (ADC2:ADCK).
9) Enable interrupts (ADC2:ADIE=1).
10) Activate A/D (ADC1:AD=1).

● Interrupt processing

1) Clear the interrupt request flag (ADC1:ADI=0).
2) Read converted values (ADDH, ADDL)
3) Activate A/D (ADC1:AD=1).

23.7 Notes on Use of 8/10-bit A/D Converter

This section summarizes notes on using the 8/10-bit A/D converter.

■ Notes on Use of 8/10-bit A/D Converter

● Notes on programmed setup

- When the A/D conversion function is used, the contents of the ADDH and ADDL registers are retained upon completion of A/D conversion. During A/D conversion, the values resulting from the last conversion are loaded.
- Do not re-select the analog input channel (ADC1: ANS3 to ANS0) while the A/D conversion function is running, in particular, during continuous activation. Disable continuous activation (ADC2: EXT = 0) before re-selecting the analog input channel.
- Starting the reset, stop, or watch mode stops the 8/10-bit A/D converter and initializes each register.
- The CPU cannot return from interrupt processing if the interrupt request flag bit (ADC1: ADI) is 1 with interrupt requests enabled (ADC2: ADIE = 1). Be sure to clear the ADI bit within the interrupt processing routine.

● Note on interrupt requests

If A/D conversion is reactivated (ADC1: AD = 1) and terminated at the same time, the interrupt request flag bit (ADC1: ADI) is set.

● Error

As $|AVR\text{-}AV_{SS}|$ decreases, an error increases relatively.

● 8/10-bit A/D converter and analog input power-on/shut-down sequences

Turn on the 8/10-bit A/D converter power supply (AV_{CC}, AV_{SS}) and analog input (AN00 to AN07) at the same as or after turning on the digital power supply (V_{CC},).

In addition, turn off the digital power supply (V_{CC}) either at the same time as or after turning off the 8/10-bit A/D converter power supply (AV_{CC}, AV_{SS}) and analog input (AN00 to AN07).

Be careful not to let the AV_{CC}, AV_{SS}, and analog input exceed the voltage of the digital power supply when turning the 8/10-bit A/D converter on and off.

● Conversion time

The conversion speed of the A/D conversion function is affected by the clock mode, main clock oscillation frequency, and main clock speed switching (gear function).

Example:Sampling time = CKIN × (ADC2: TIM1/TIM0 setting)

Compare time = CKIN × 10 (fixed value) + MCLK

AD start processing time:Min. = MCLK + MCLK

Max. = MCLK + CKIN

Conversion time = A/D start processing time + sampling time + compare time

- The error max. 1 CKIN-1MCLK may occur depending on the timing of AD startup.
- Program the software satisfied with "sampling time" and "compare time" in A/D converter of data sheet.

23.8 Sample Programs for 8/10-bit A/D Converter

Fujitsu provides sample programs to operate the 8/10-bit A/D converter.

■ Sample Programs for 8/10-bit A/D Converter

For sample programs for the 8/10-bit A/D converter, see "■ Sample Programs" in "Preface".

■ Setting Methods not Covered by Sample Programs

● Selecting the operating clock for the 8/10-bit A/D converter

Use the clock select bits (ADC2.CKDIV1/CKDIV0) to select the operating clock.

● Selecting the sampling time of the 8/10-bit A/D converter

Use the sampling time select bits (ADC2.TIM1/TIM0) to select the sampling time.

● Controlling the analog switch for internal reference power shutdown of the 8/10-bit A/D converter

Use the analog switch control bit (ADC1.ADMVX) to control the internal reference power shutdown analog switch.

Control item	Analog switch control bit (ADMVX)
To turn off internal reference power supply	Set the bit to "0".
To turn on internal reference power supply	Set the bit to "1".

● Selecting the 8/10-bit A/D converter activation method

Use the continuous activation enable bit (ADC2.EXT) to select the startup trigger.

A/D startup factor	Continuous activation enable bit (EXT)
To select the software trigger	Set the bit to "0".
To select the input clock rising signal	Set the bit to "1".

- Generating a software trigger

  Use the A/D conversion start bit (ADC1.AD) to generate a software trigger.

Operation	A/D conversion start bit (AD)
To generate a software trigger	Set the bit to "1".

- Activation using the input clock

  A startup trigger is generated at the rise of the input clock signal.

To select the input clock, use the external start signal select bit (ADC2.ADCK).

Input clock	External start signal select bit (ADCK)
To select the ADTG input pin	Set the bit to "0".
To select the 8/16-bit compound timer (TO00)	Set the bit to "1".

● Selecting the A/D conversion precision

To select the precision of conversion results, use the precision select bit (ADC2.AD8).

Operation mode	Precision select bit (AD8)
To select 10-bit precision	Set the bit to "0".
To select 8-bit precision	Set the bit to "1".

● Using analog input pins

To select an analog input pin, use the analog input channel select bits (ADC1:ANS3 to ANS0).

Operation	Analog input channel select bits (ANS3 to ANS0)
To use the AN00 pin	Set the pins to "0000_B".
To use the AN01 pin	Set the pins to "0001_B".
To use the AN02 pin	Set the pins to "0010_B".
To use the AN03 pin	Set the pins to "0011_B".
To use the AN04 pin	Set the pins to "0100_B".
To use the AN05 pin	Set the pins to "0101_B".
To use the AN06 pin	Set the pins to "0110_B".
To use the AN07 pin	Set the pins to "0111_B".

● Checking the completion of conversion

The following two methods can be used to check whether conversion has been completed.

- Checking with the interrupt request flag bit (ADC1.ADI)

Interrupt request flag bit (ADI)	Meaning
The value read is "0".	A/D conversion completed with no interrupt request
The value read is "1".	A/D conversion completed with interrupt request generated

- Checking with the conversion flag bit (ADC1.ADMV)

Conversion flag bit (ADMV)	Setting
The value read is "0".	A/D conversion completed (suspended)
The value read is "1".	A/D conversion in progress

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

Interrupt source	Interrupt level setting register	Interrupt vector
8/10-bit A/D converter	Interrupt level register (ILR4) Address: $0007D_H$	#18 Address: $0FFD6_H$

● Enabling, disabling, and clearing interrupts

To enable interrupts, use the interrupt request enable bit (ADC2.ADIE).

Control item	Interrupt request enable bit (ADIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupt requests, use the interrupt request bit (ADC1.ADI).

Control item	Interrupt request bit (ADI)
To clear an interrupt request	Set the bit to "0". Or activate A/D.

CHAPTER 24

LCD CONTROLLER

This chapter describes the functions and operations of the LCD controller.

24.1 Overview of LCD Controller

The LCD controller contains 8 bytes of display data memory and controls LCD display via 4 common outputs and 16 segment outputs. It offers a choice of three different duty outputs to directly drive the LCD panel (liquid crystal display device).

■ Functions of LCD Controller

The LCD controller uses its segment and common outputs to display the contents of display data memory (display RAM) directly on the LCD panel (liquid crystal display device).

- Step-up power supply circuit integrated (Optional).
- LCD drive voltage divider resistor integrated. Also capable of connecting an external divider resistor.
- Up to 4 common outputs (COM0 to COM3) and 16 segment outputs (SEG00 to SEG15) are available.
- 8 bytes (16 × 4 bits) of display RAM integrated.
- Main clock or sub clock selectable as the operating clock.
- Blinking function (limited to some pins).
- Capable of directly driving the LCD panel.
- Duty selectable from among 1/2, 1/3, and 1/4 (restricted by the bias setting).

Table 24.1-1 lists the bias-duty combinations available.

Table 24.1-1 Bias-duty Combinations

Bias	1/2 Duty	1/3 Duty	1/4 Duty
1/2 bias	❍	×	×
1/3 bias	×	❍	❍
1/2 bias	×	×	×
1/3 bias	×	❍	❍

❍ : Recommended mode

× : Prohibited

24.2 Configuration of LCD Controller

The LCD controller consists of the following blocks, which are divided functionally into a controller section that generates the segment and common signals based on the content of display RAM and a driver section that drives the LCD.

Controller section

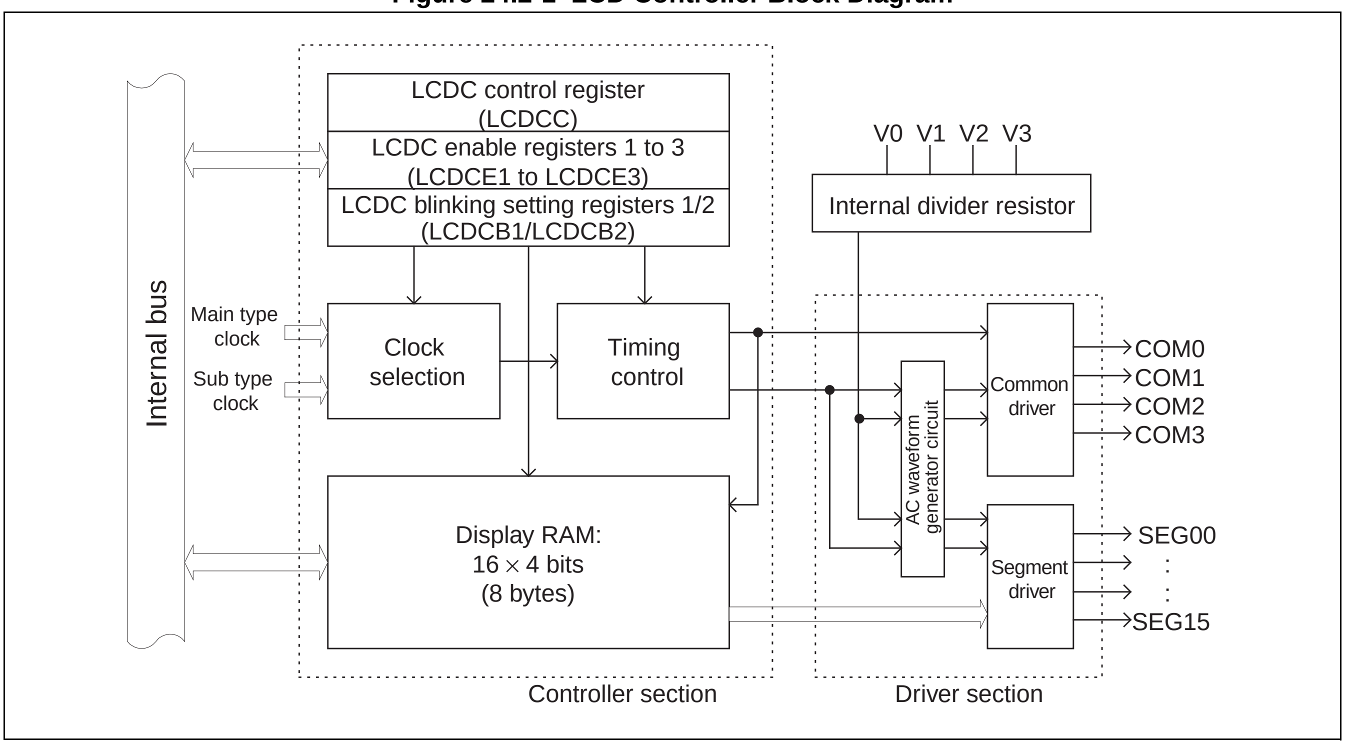
- **LCDC control register (LCDCC)**
- **LCDC enable registers (LCDCE1 to LCDCE3)**
- **LCDC blinking setting registers (LCDCB1/LCDCB2)**
- **Display RAM**
- **Clock selection**
- **Timing control**

Driver section

- **AC waveform generator circuit**
- **Common driver**
- **Segment driver**

■ LCD Controller Block Diagram

Figure 24.2-1 LCD Controller Block Diagram



- **LCDC control register (LCDCC)**

This register is used to select the clock for generating the frame period, select display or display blinking, select the display mode, select the frame period clock, and control the LCD driving power supply.

- **LCDC enable registers 1 to 3 (LCDCE1 to LCDCE3)**

These registers are used to control port inputs, blink interval, and pins.

- **LCDC blinking setting registers 1, 2 (LCDCB1/LCDCB2)**

These registers are used to turn blinking on or off.

- **Display RAM**

16 × 4 bits of RAM for generating segment output signals. The contents of this RAM are read automatically in synchronization with the common signal selection timing and output from the segment output pins.

The contents of VRAM is output from segment output pin when display RAM is re-written.

- **Clock selection**

The frame frequency is generated based on the selection from amongst the eight frequencies generated from the two clocks.

- **Timing control**

The common and segment signals are controlled based on the frame frequency and register settings.

- **AC waveform generator circuit**

This block generates AC waveforms for driving the LCD from timing control signals.

- **Common driver**

This block is the driver of the LCD common pins.

- **Segment driver**

This block is the driver of the LCD segment pins.

- **Divider resistor**

This block is a resistor used to generate the LCD drive voltage. The divider resistor can be connected as an external component. The V0 to V3 pins serve as the divider resistor connection pins.

■ LCD Controller Power Supply Voltage

The power supply voltage for the LCD driver is generated by using the internal divider resistors or by connecting divider resistors to the V0 to V3 pins.

■ Input Clock

The LCD controller uses the output clock of time-base timer or watch prescaler as the input clock (operation clock).

24.2.1 Internal Driver Resistors for LCD Controller

The power supply voltage for the LCD driver is generated by the internal divider resistors (external divider resistors can also be connected).

■ Internal Divider Resistors

Internal divider resistors are included in this series. In addition, external divider resistors can be connected to the LCD driving power pins (V0 to V3).

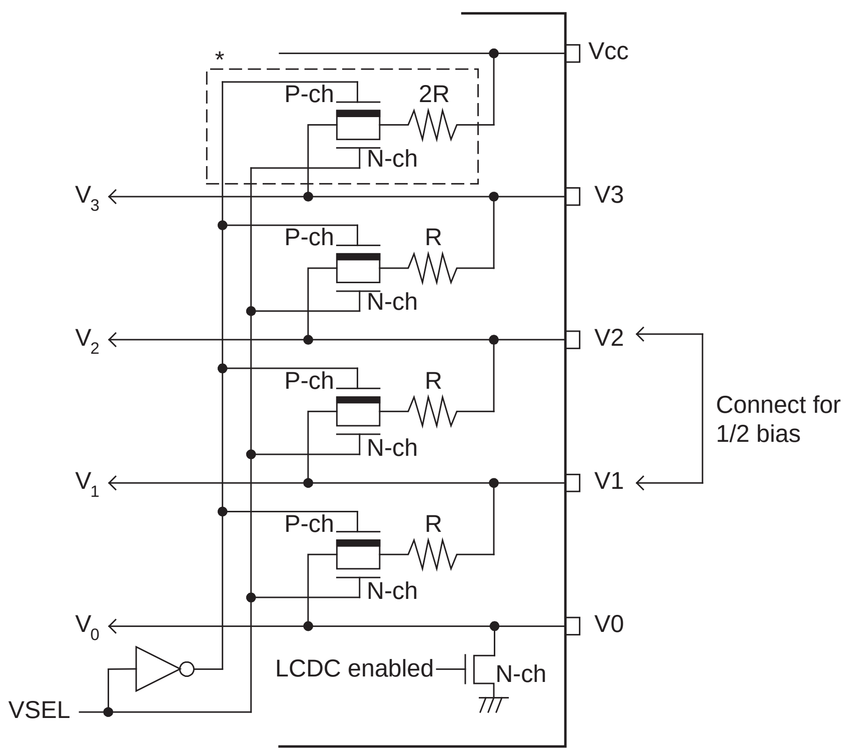
The internal and external divider resistors are selected by the driving power control bit in the LCDC control register (LCDCC: VSEL). Setting the VSEL bit to "1" energizes the internal divider resistors. To use only the internal divider resistors without connecting the external divider resistors, set the VSEL bit to "1".

The LCD controller stops upon transition to main stop or watch mode (STBC:TMD = 1) while operation in main stop and watch modes is disabled (LCDCC:LCDEN = 0) with LCD operation halted (LCDCC:MS1, MS0 = 00_B).

To use the 1/2 bias setting, connect the V2 and V1 pins together.

Figure 24.2-2 shows an equivalent circuit with internal divider resistors used.

Figure 24.2-2 Equivalent Circuit with Internal Divider Resistors Used



V0 to V3: Voltages at V0 to V3 pins
*: Only for 3V products

MB95150/M Series

■ Use of Internal Divider Resistors and Brightness Control

When internal divider resistors are used on a 3V model, the internal 2R resistor reduces the V1, V2, and V3 voltages accordingly. Figure 24.2-3 shows the case when the internal divider resistors are used.

If sufficient brightness is not achieved with the internal divider resistors in use, connect a variable resistor (VR) externally (between the Vcc and V3 pins) to adjust the V3 voltage. Figure 24.2-4 shows an example of connecting a VR to internal divider resistors for brightness control.

Figure 24.2-3 States with Internal Divider Resistors Used

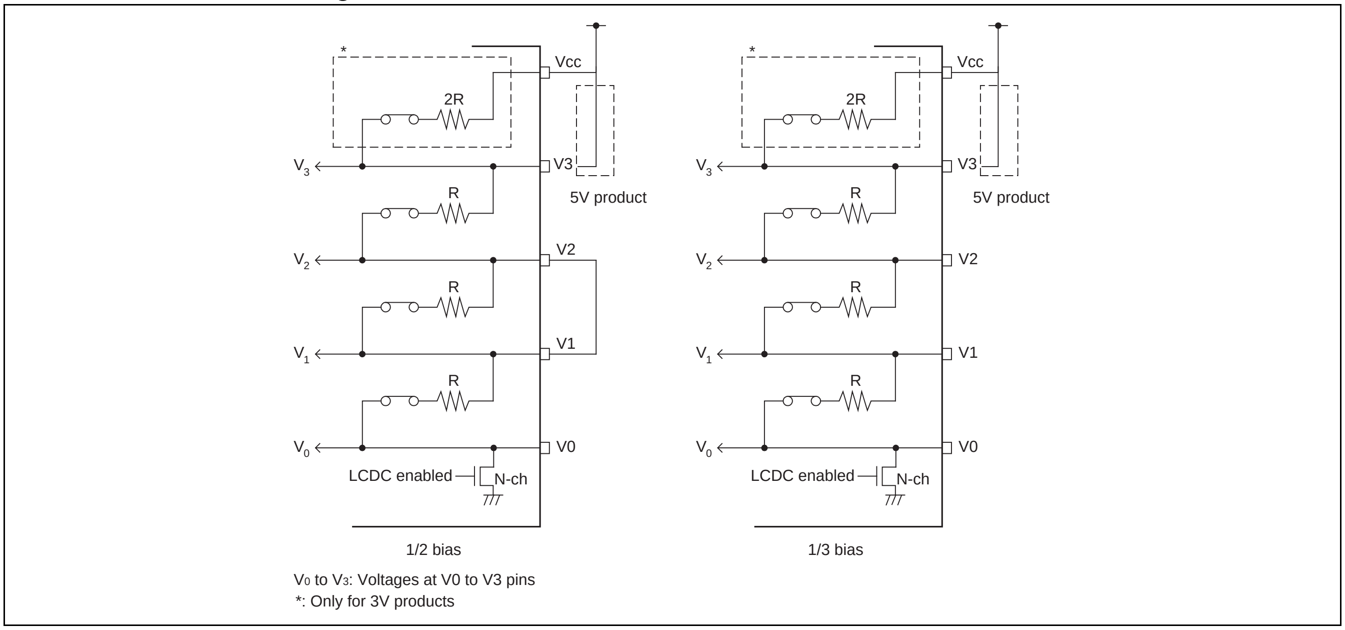
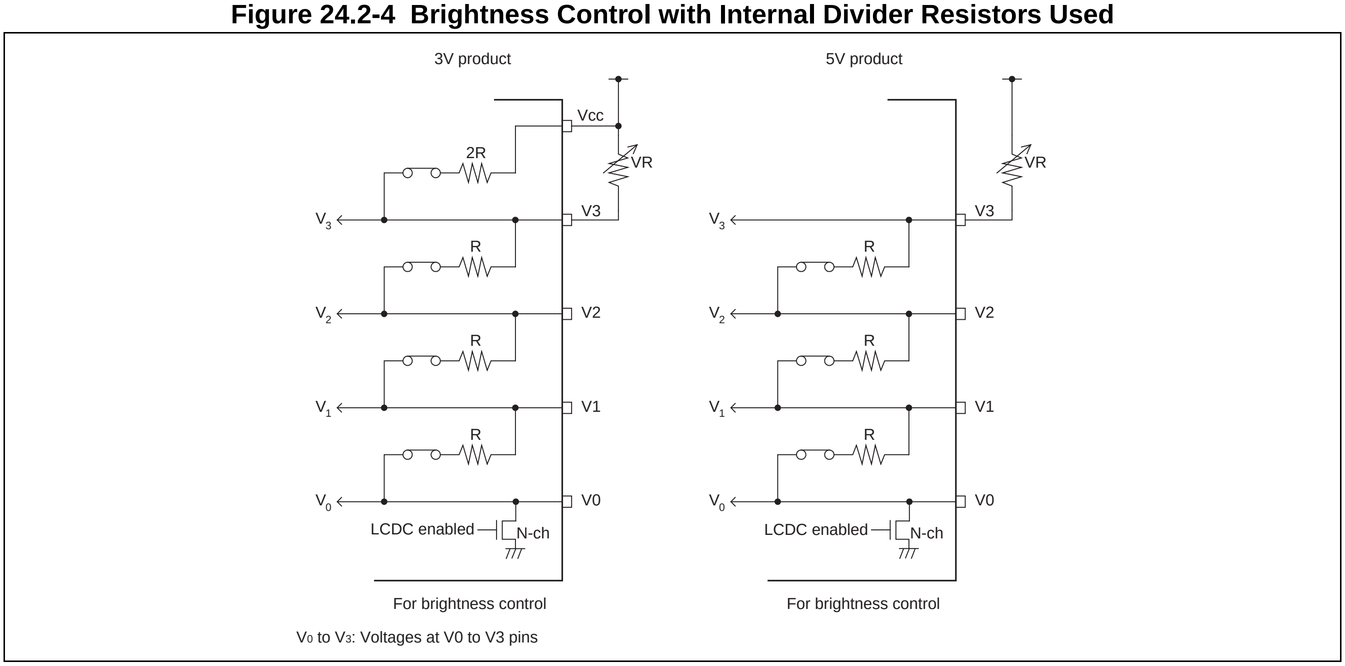


Figure 24.2-4 Brightness Control with Internal Divider Resistors Used



As the internal 2R resistor is enabled during LCD operation, connect the VR resistor in parallel with the 2R resistor.

24.2.2 External Divider Resistors for LCD Controller

This series allow external divider resistors to be connected to the V0 to V3 pins. Also, the brightness can be adjusted by connecting a variable resistor between the V_{CC} and V3 pins.

■ External Divider Resistors

If not using the internal divider resistors, you can connect external divider resistors to the LCD drive power supply pins (V0 to V3) instead. Figure 24.2-5 shows an example of connecting external divider resistors and Table 24.2-1 lists the LCD drive voltage settings for the bias method.

Figure 24.2-5 Example of Connecting External Divider Resistors

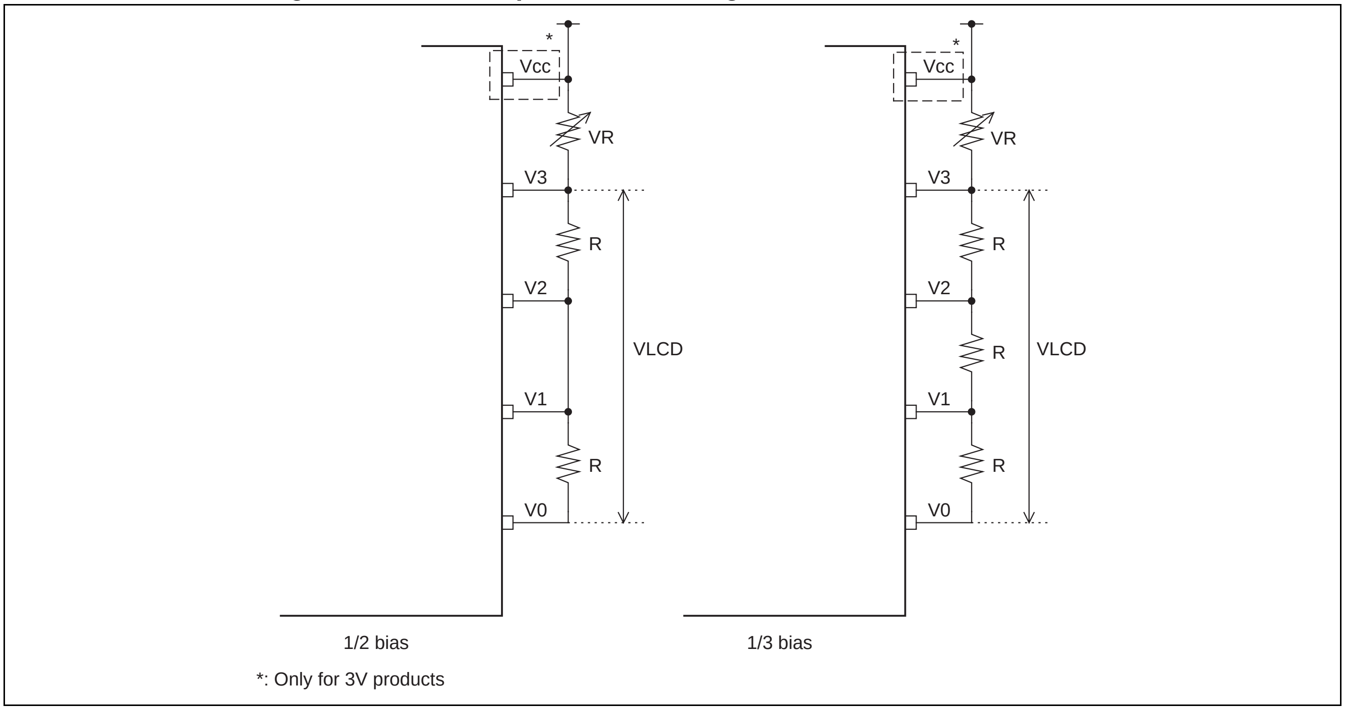


Table 24.2-1 LCD Driving Voltage Settings

	V_3	V_2	V_1	V_0
1/2 bias	VLCD	1/2 VLCD	1/2 VLCD	GND
1/3 bias	VLCD	2/3 VLCD	1/3 VLCD	GND

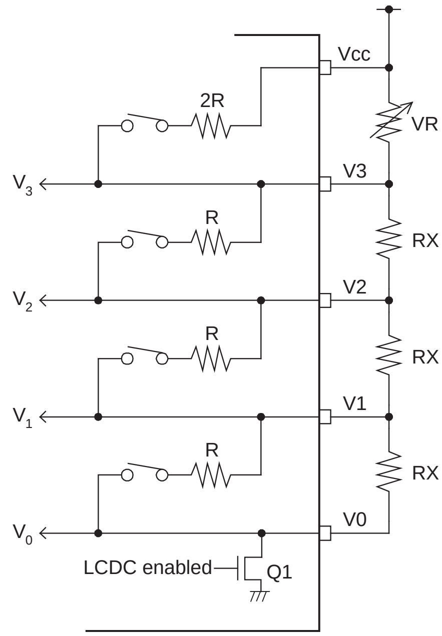
V_0 to V_3: Voltages at V0 to V3 pins

VLCD: LCD operating voltage

■ Use of External Divider Resistors

As the V0 pin is connected to V_{SS} (GND) internally via a transistor, when using external divider resistors, you can shut off the current flowing to the resistors when the LCD controller is halted by connecting the V_{SS} end of the divider resistors to the V0 pin. Figure 24.2-6 shows the states with external divider resistors used.

Figure 24.2-6 States with External Divider Resistors Used



V_0 to V_3: Voltages at V0 to V3 pins

1) To connect the external divider resistors without being affected by the internal divider resistors, you need to write "0" to the drive voltage control bit in the LCDC control register (LCDCC:VSEL) to disconnect all the internal divider resistors.
2) When the internal divider resistors are disconnected, writing a value other than "00_B" to the display mode select bits (MS1 and MS0) in the LCDC control register turns on the LCDC enable transistor (Q1) and current flows in the external divider resistors.
3) Writing "00_B" to the display mode select bits (MS1 and MS0) turns off the LCDC enable transistor (Q1) and this stops the current flow in the external divider resistors.

Note:

The externally connected RX resistors depend on the LCD you are using. Select appropriate resistances to match the LCD.

24.3 Pins of LCD Controller

This section describes the pins of the LCD controller.

■ Pins of LCD Controller

The pins related to the LCD controller are: 4 common output pins (COM0 to COM3), 16 segment output pins (SEG00 to SEG15), and 4 LCD drive power supply pins (V0 to V3). The number of segment output pins differs for each MCU series.

To use these pins for the LCD, set the corresponding bits in the LCDC enable registers (LCDCE1 to LCDCE3) to "1".

To use LCD pins for ports, set the PICTL bit in LCDC enable register 1 (LCDCE1) to "1" and the corresponding select bits (COM/SEG) in LCDC enable registers 1 to 3 to "0".

● COM0 to COM3 pins

The COM0 to COM3 pins are LCD common outputs.

These pins also serve as I/O ports.

● SEG00 to SEG15 pins

The SEG00 to SEG15 pins are LCD segment output pins.

The number of segment output pins depends on each MCU series.

These pins also serve as I/O ports.

● V0 to V3 pins

These pins are used as the power supply pins for driving the LCD.

These pins also serve as I/O ports.

■ Block Diagram of LCD-related Pins

Figure 24.3-1 Block Diagram of LCD-related Pins (V0 to V3)

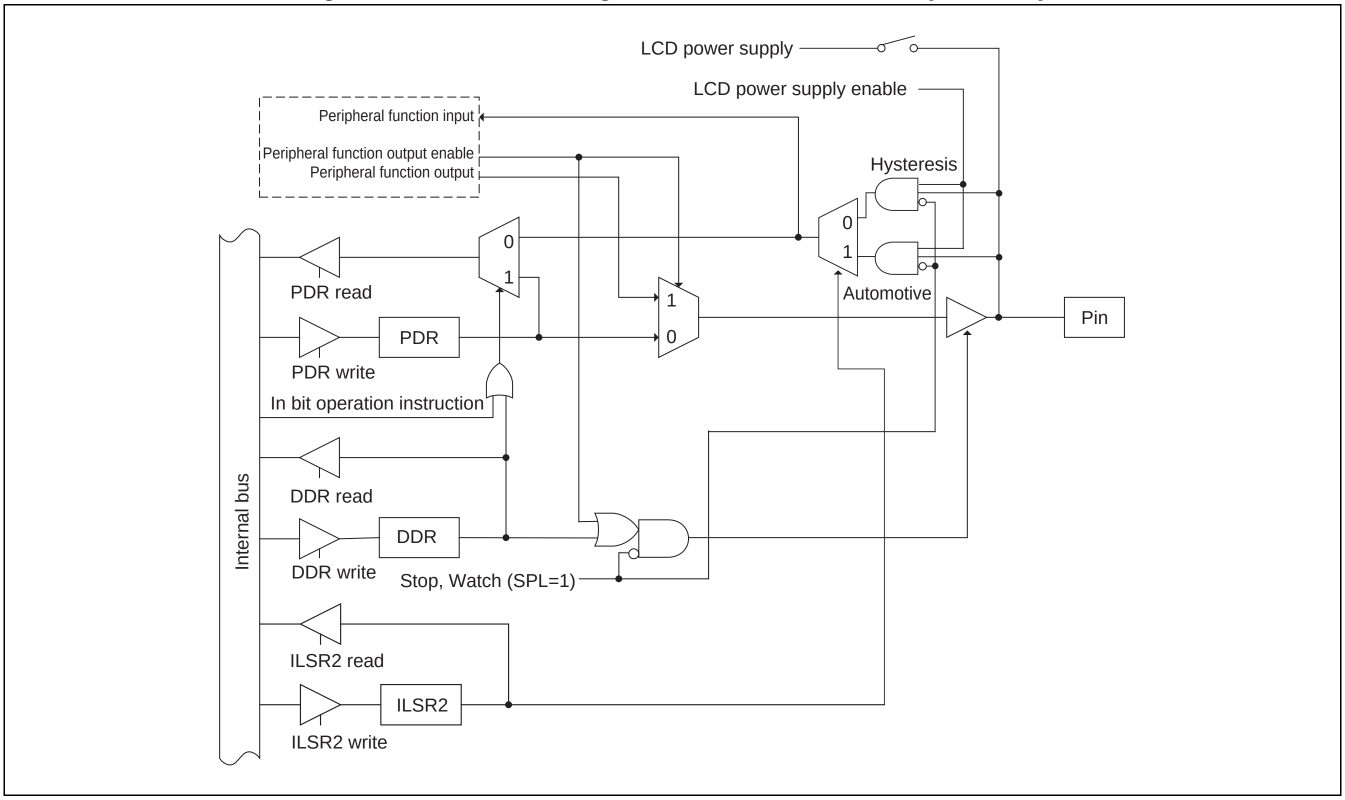


Figure 24.3-2 Block Diagram of LCD-related Pins (COM0 to COM3)

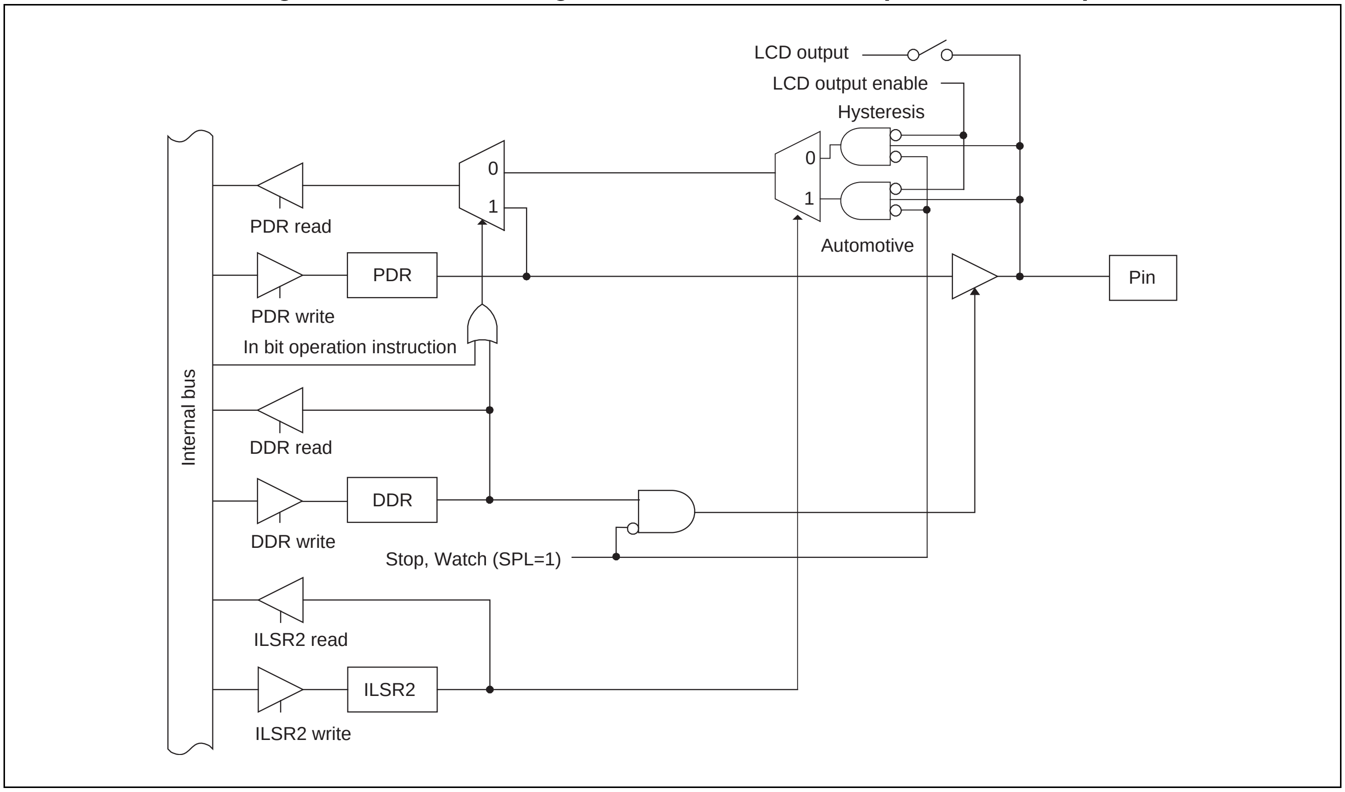


Figure 24.3-3 Block Diagram of LCD-related Pins (SEG00 to SEG07)

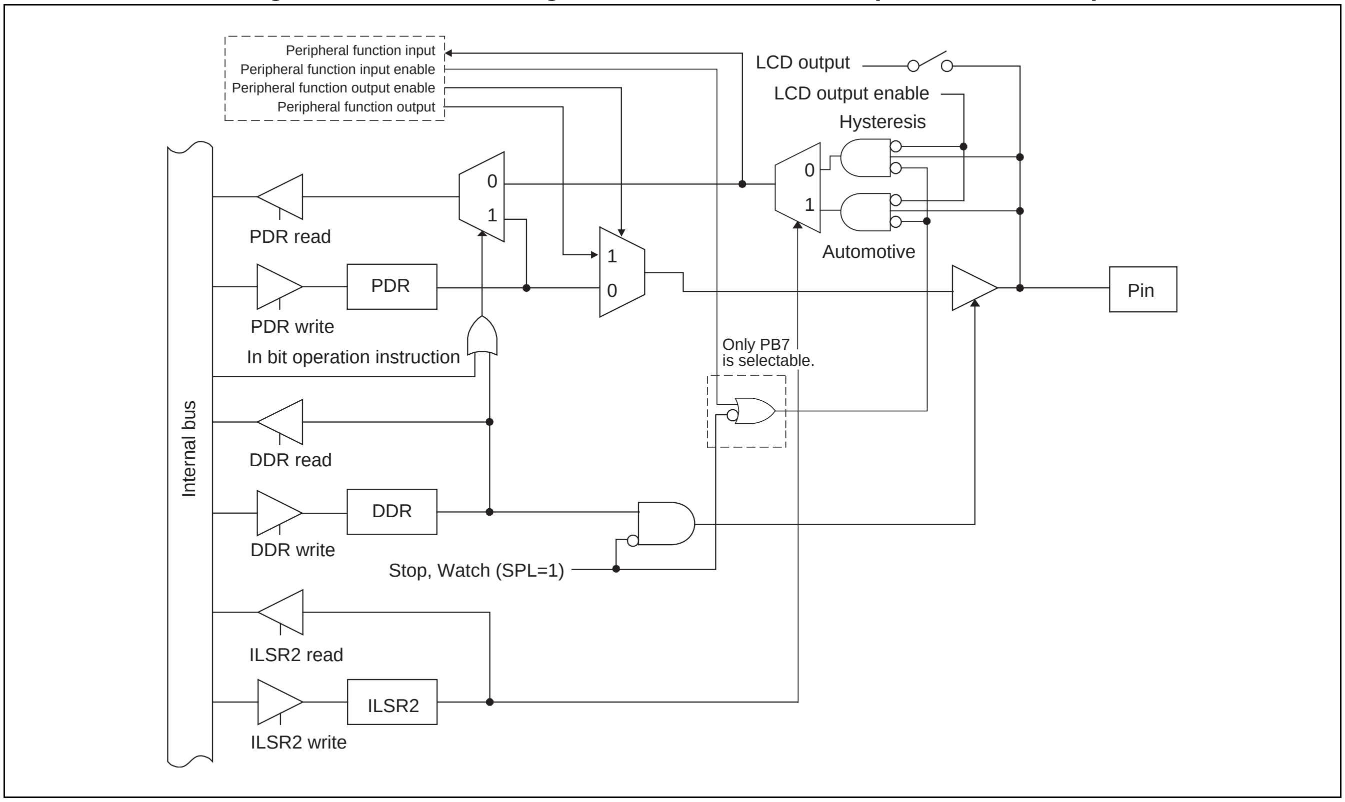
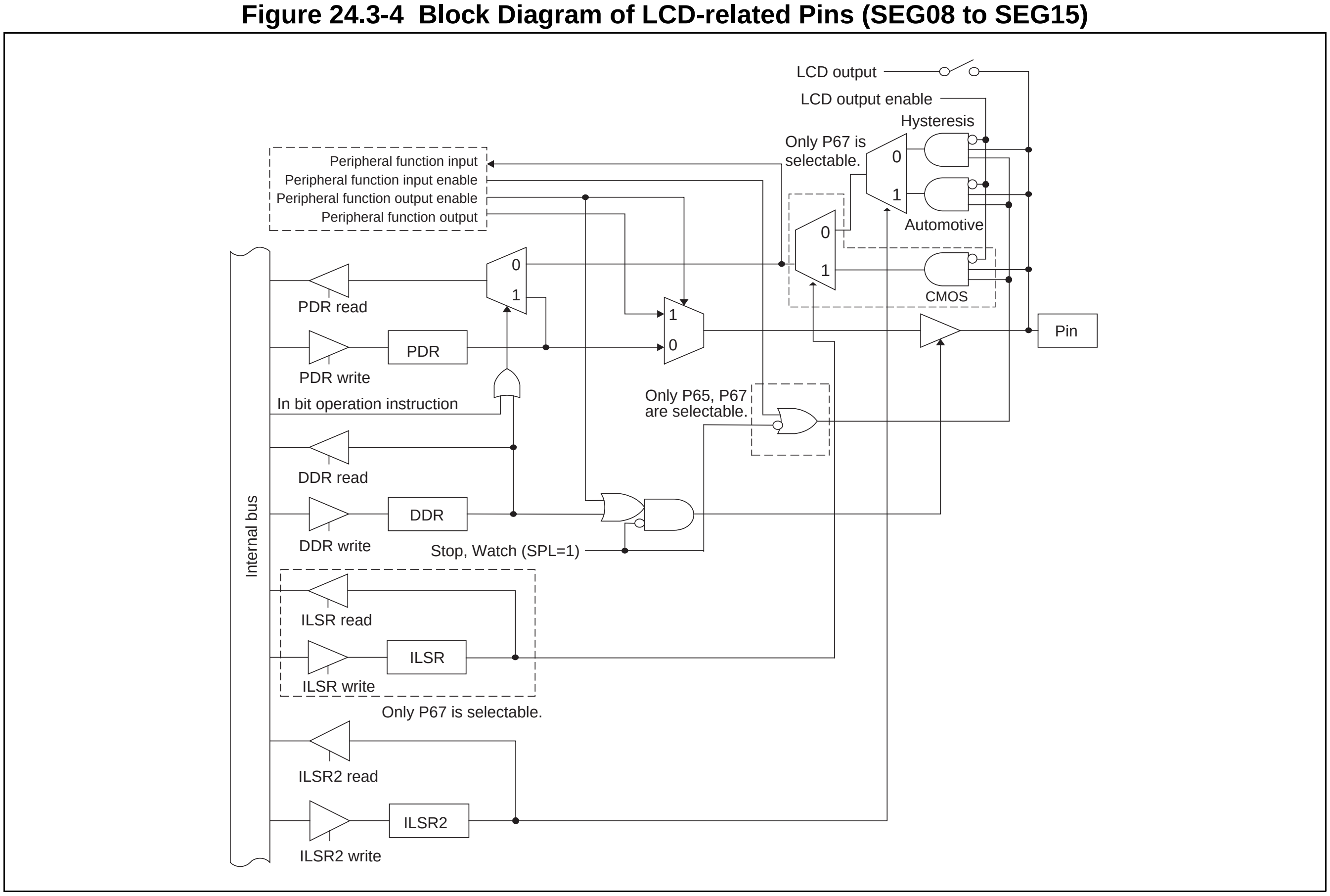


Figure 24.3-4 Block Diagram of LCD-related Pins (SEG08 to SEG15)



24.4 Registers of LCD Controller

This section describes the registers of the LCD controller.

■ LCD Controller Register List

Figure 24.4-1 LCD Controller Registers

LCDC control register (LCDCC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FC4_H$	CSS	LCDEN	VSEL	BK	MS1	MS0	FP1	FP0	00010000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDC enable register 1 (LCDCE1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FC5_H$	PICTL	BLSEL	VE2	VE1	COM3	COM2	COM1	COM0	00110000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDC enable register 2 (LCDCE2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FC6_H$	SEG07	SEG06	SEG05	SEG04	SEG03	SEG02	SEG01	SEG00	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDC enable register 3 (LCDCE3)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FC7_H$	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG09	SEG08	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDC blinking setting register 1 (LCDCB1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FCB_H$	S1C3	S1C2	S1C1	S1C0	S0C3	S0C2	S0C1	S0C0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDC blinking setting register 2 (LCDCB2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FCC_H$	S3C3	S3C2	S3C1	S3C0	S2C3	S2C2	S2C1	S2C0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/Writable (Read value is the same as write value)

24.4.1 LCDC Control Register (LCDCC)

The LCDC control register (LCDCC) is used to set the clock, display mode, and power supply control.

■ LCDC Control Register (LCDCC)

Figure 24.4-2 LCDC Control Register (LCDCC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FC4H	CSS	LCDEN	VSEL	BK	MS1	MS0	FP1	FP0	00010000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

FP1	FP0	Frame period select bits	
		Main clock (CSS = 0)	Sub clock (CSS = 1)
0	0	$2^{14} \times N/F_{CH}$	$2^{6} \times N/F_{CL}$
0	1	$2^{15} \times N/F_{CH}$	$2^{7} \times N/F_{CL}$
1	0	$2^{16} \times N/F_{CH}$	$2^{8} \times N/F_{CL}$
1	1	$2^{17} \times N/F_{CH}$	$2^{9} \times N/F_{CL}$

MS1	MS0	Display mode select bits
0	0	LCD operation halt
0	1	1/2 duty output mode (time division number N = 2)
1	0	1/3 duty output mode (time division number N = 3)
1	1	1/4 duty output mode (time division number N = 4)

BK	Display blanking select bit
0	Display
1	Display blanking

VSEL	LCD drive power supply control bit
0	Use external divider resistors.
1	Use internal divider resistors.

LCDEN	Main stop/watch mode operation enable bit
0	Disable operation.
1	Enable operation.

CSS	Frame period generation clock select bit
0	Main clock
1	Sub clock

F_{CH} : Main oscillation frequency
F_{CL} : Sub oscillation frequency
R/W : Readable/Writable (Read value is the same as write value)
(shaded) : Initial value

Table 24.4-1 Functions of Bits in LCDC Control Register (LCDCC)

Bit name		Function
bit7	CSS: Frame period generation clock select bit	Selects the clock to generate the frame period for LCD display. • When this bit is "0", the LCD controller operates with the output of the time-base timer driven by the main clock oscillation. When the bit is "1", the LCD controller operates with the output of the watch prescaler driven by the sub clock. Note: As the main clock stops oscillation in main stop mode and sub clock mode, the LCD controller cannot operate with the output of the time-base timer in these modes. Shifting the main clock speed (using the gear function) during operation with the time-base timer output does not affect the frame period. LCD display may flicker when the clock speed is being shifted. Before shifting it, therefore, temporarily halt the display, for example, by using blanking (LCDCC:BK = 1).
bit6	LCDEN: Main stop/watch mode operation enable bit	Specifies whether the LCD controller is to continue to operate in main stop and watch (time-base timer) modes. **When the bit is "0"**, LCD display stops. **When this bit is "1"**, LCD display continues even after transition to main stop or watch mode. Note: The sub clock must be selected (CSS = 1) to continue operating in main stop or watch mode.
bit5	VSEL: LCD driving power control bit	In models with internal divider resistors, this bit selects whether to energize the internal divider resistors. **Setting the bit to "0"**: Shuts off the internal divider resistors. **Setting the bit to "1"**: Energizes the internal divider resistors. To connect external divider resistors, set this bit to "0".
bit4	BK: Display blanking select bit	Selects whether to display or blank the LCD. • When display blanking (no display, BK = 1) is selected, the segment output changes to a deselected waveform (waveform not treated as a display condition).
bit3, bit2	MS1, MS0: Display mode select bits	Select one of three output waveform duties. • The common pin to be used is determined depending on the selected duty output mode. • When these bits are "00_B", the LCD controller driver stops the display operation. Note: If the selected frame period generation clock can halt, for example, upon transition to stop mode, halt the display operation (MS1, MS0 = 00) in advance. As LCD display may flicker upon switching, halt the display temporarily, for example, by using blanking (LCDCC:BK = 1) before switching.
bit1, bit0	FP1, FP0: Frame period select bits	Select one of four LCD display frame periods. Note: Set the registers after calculating the optimum frame frequency according to the LCD module to be used. The frame period is affected by the source oscillation frequency. As LCD display may flicker upon switching, halt the display temporarily, for example, by using blanking (LCDCC:BK = 1) before switching.

24.4.2 LCDC Enable Register 1 (LCDCE1)

LCDC enable register 1(LCDCE1) is used to control port input, set the blink cycle, and enable LCD pins.

■ LCDC Enable Register 1 (LCDCE1)

Figure 24.4-3 LCDC Enable Register 1 (LCDCE1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FC5H	PICTL	BLSEL	VE2	VE1	COM3	COM2	COM1	COM0	00110000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

COM0	COM0 select bit
0	General-purpose I/O port
1	Common output

COM1	COM1 select bit
0	General-purpose I/O port
1	Common output

COM2	COM2 select bit
0	General-purpose I/O port
1	Common output

COM3	COM3 select bit
0	General-purpose I/O port
1	Common output

VE1	V2 to V0 select bit
0	General-purpose I/O port
1	V2 to V0 dedicated pins

VE2	V3 select bit
0	General-purpose I/O port
1	V3 dedicated pin

BLSEL	Blinking interval select bit
0	0.5 s
1	1.0 s

PICTL	Port input control bit
0	Disable port input
1	Enable port I/O

R/W : Readable/Writable (Read value is the same as write value)
(shaded) : Initial value

MB95150/M Series

Table 24.4-2 Functions of Bits in LCDC Enable Register 1 (LCDCE1)

Bit name		Function
bit7	PICTL: Port input control bit	Controls the port I/O pins that also serve as segment or common outputs. **Setting the bit to "0"**: Shuts off the port inputs and suppresses shoot-through current during LCD output. Shuts off the port output. **Setting the bit to "1"**: Enables the pins for port I/O. Set the bit to 1 to use the pins as ports. Note: As the port inputs are disconnected at a reset, be sure to set the bit to "1" if you wish to use the pins as port inputs. When used as the segment and common, the port inputs are disconnected regardless of this bit.
bit6	BLSEL: Blinking interval select bit	Selects the blinking interval when blinking is enabled. Blinking is enabled by LCDC blinking setting registers 1 and 2 (LCDCB1, LCDCB2). The setting "1.0s" causes the LCD to remain on for 0.5s and off for 0.5s; the setting "0.5s" causes it to remain on for 0.25s and off for 0.25s.
bit5	VE2: V3 select bit	Selects the status of the V3 pin. **Setting the bit to "0"**: Causes the pin to function as a general-purpose I/O port. **Setting the bit to "1"**: Causes the pin to function as the V3 pin.
bit4	VE1: V2 to V0 select bit	Selects the status of the V2 to V0 pins. **Setting the bit to "0"**: Causes the pins to function as general-purpose I/O ports. **Setting the bit to "1"**: Causes the pins to function as the V2 to V0 pins.
bit3	COM3: COM3 select bit	Selects the status of the COM3 pin. **Setting the bit to "0"**: Causes the pin to function as a general-purpose I/O port. **Setting the bit to "1"**: Causes the pin to function as the COM3 pin.
bit2	COM2: COM2 select bit	Selects the status of the COM2 pin. **Setting the bit to "0"**: Causes the pin to function as a general-purpose I/O port. **Setting the bit to "1"**: Causes the pin to function as the COM2 pin.
bit1	COM1: COM1 select bit	Selects the status of the COM1 pin. **Setting the bit to "0"**: Causes the pin to function as a general-purpose I/O port. **Setting the bit to "1"**: Causes the pin to function as the COM1 pin.
bit0	COM0: COM0 select bit	Selects the status of the COM0 pin. **Setting the bit to "0"**: Causes the pin to function as a general-purpose I/O port. **Setting the bit to "1"**: Causes the pin to function as the COM0 pin.

24.4.3 LCDC Enable Registers 2, 3 (LCDCE2, LCDCE3)

LCDC enable registers 2, 3 (LCDCE2, LCDCE3) are used to enable the individual segment pins for output.

■ LCDC Enable Registers 2, 3 (LCDCE2, LCDCE3)

Figure 24.4-4 LCDC Enable Registers 2, 3 (LCDCE2, LCDCE3)

LCDCE2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	SEG07	SEG06	SEG05	SEG04	SEG03	SEG02	SEG01	SEG00	00000000_B
$0FC6_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDCE3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG09	SEG08	00000000_B
$0FC7_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SEGxx	Segment select bit
0	General-purpose I/O port
1	SEGxx segment output

R/W :Readable/Writable (Read value is the same as write value)

(shaded) :Initial value

24.4.4 LCDC Blinking Setting Registers 1/2 (LCDCB1/LCDCB2)

LCDC blinking setting registers 1/2 (LCDCB1/LCDCB2) are used to turn blinking on or off.

■ LCDC Blinking Setting Registers 1/2 (LCDCB1/LCDCB2)

Figure 24.4-5 LCDC Blinking Setting Registers 1/2 (LCDCB1/LCDCB2)

LCDCB1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	S1C3	S1C2	S1C1	S1C0	S0C3	S0C2	S0C1	S0C0	00000000_B
$0FCB_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCDCB2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	S3C3	S3C2	S3C1	S3C0	S2C3	S2C2	S2C1	S2C0	00000000_B
$0FCC_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SnCm	Blinking select bit
0	Blinking OFF
1	SnCmSEGn/COMm blinking ON

Sn :SEGn (n=0 to 3)
Cm :COMm (m=0 to 3)

R/W: Readable/Writable (Read value is the same as write value)

The blinking function applies to the dots specified by the combinations of SEG0 to SEG3 and COM0 to COM3.

The blinking interval is selected by the BLSEL bit in LCDC enable register 1 (LCDCE1).

All the segments for which blinking has been turned on blink synchronously.

The setting of each blinking select bit remains in effect when the corresponding bit in display RAM holds 1.

24.5 LCD Controller Display RAM

Display RAM is a 16 × 4 bits (8 bytes) of display data memory used to generate segment output signals.

■ Display RAM and Output Pins

The contents of display RAM are read automatically with the common signal selection timing and output from the segment output pins.

Each bit containing "1" is converted to the selected voltage (displayed on the LCD); the one containing "0" is converted to the unselected voltage (undisplayed on the LCD).

As the LCD display operation is performed asynchronously with the CPU operation, display RAM can be read from or written to at any timing. Pins not assigned as segment outputs can be used as I/O ports and the corresponding areas of display RAM can be used as normal RAM. Table 24.5-1 shows the relationships between duty setting/common outputs and bits used in display RAM.

Figure 24.5-1 shows how display RAM addresses are allocated for common output and segment output pins.

Figure 24.5-1 Display RAM and Common/Segment Output Pins

Address					
n	bit3	bit2	bit1	bit0	SEG00
	bit7	bit6	bit5	bit4	SEG01
n+1	bit3	bit2	bit1	bit0	SEG02
	bit7	bit6	bit5	bit4	SEG03
n+2	bit3	bit2	bit1	bit0	SEG04
	bit7	bit6	bit5	bit4	SEG05
:	:	:	:	:	:
:	:	:	:	:	:
n+5	bit3	bit2	bit1	bit0	SEG10
	bit7	bit6	bit5	bit4	SEG11
n+6	bit3	bit2	bit1	bit0	SEG12
	bit7	bit6	bit5	bit4	SEG13
n+7	bit3	bit2	bit1	bit0	SEG14
	bit7	bit6	bit5	bit4	SEG15
	COM3	COM2	COM1	COM0	

Area and common pins used at 1/2 duty (COM1 to COM0)
Area and common pins used at 1/3 duty (COM2 to COM0)
Area and common pins used at 1/4 duty (COM3 to COM0)

Note: The number of segment output pins depends on each MCU series.

Table 24.5-1 Relationships Between Duty Settings/Common Outputs and Display RAM Bits Used

Duty Setting	Common Output Used	Display Data Bits Used							
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
1/2	COM0, COM1 (2 pins)	-	-	❍	❍	-	-	❍	❍
1/3	COM0 to COM2 (3 pins)	-	❍	❍	❍	-	❍	❍	❍
1/4	COM0 to COM3 (4 pins)	❍	❍	❍	❍	❍	❍	❍	❍

❍: Used
- : Unused

24.6 Operations of LCD Controller

This section describes the operations of the LCD controller.

■ Operations of LCD Controller

Figure 24.6-1 shows the settings required for LCD display.

Figure 24.6-1 LCD Controller Settings

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LCDCC	CSS	LCDEN	VSEL	BK	MS1	MS0	FP1	FP0
	◎	◎	◎	◎	Other than 00		◎	◎
LCDCE1	PICTL	BLSEL	VE2	VE1	COM3	COM2	COM1	COM0
	◎	◎	◎	◎	◎	◎	◎	◎
LCDCE2	SEG07	SEG06	SEG05	SEG04	SEG03	SEG02	SEG01	SEG00
	◎	◎	◎	◎	◎	◎	◎	◎
LCDCE3	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG09	SEG08
	◎	◎	◎	◎	◎	◎	◎	◎
LCDCB1	S1C3	S1C2	S1C1	S1C0	S0C3	S0C2	S0C1	S0C0
	◎	◎	◎	◎	◎	◎	◎	◎
LCDCB2	S3C3	S3C2	S3C1	S3C0	S2C3	S2C2	S2C1	S2C0
	◎	◎	◎	◎	◎	◎	◎	◎
Display RAM	Display data							

◎ : Used bit

- When the selected frame period generation clock is oscillating with the settings made as in Figure 24.6-1, the LCD controller outputs the LCD panel drive waveform to the common and segment output pins (COM0 to COM3, SEG00 to SEG15) according to the contents of display RAM and the LCDC register settings.
- The LCD output pins are selected according to LCDCE1 to LCDCE3. The pins not selected as LCD outputs are used as general-purpose I/O ports.
- The frame period generation clock can be changed even during LCD display operation. As the display may flicker when it is changed, however, you should always turn off the display temporarily, for example, using the blanking (LCDCC:BK = 1) function in advance.
- The display drive output is a 2-frame alternating waveform selected by bias and duty settings.
- The COM2 and COM3 pin outputs in 1/2 duty mode and the COM3 pin output in 1/3 duty mode can be used to output the deselected level waveform or as I/O ports.
- To use the blink function, set the corresponding bits in LCDC blinking setting registers 1 and 2 (LCDCB1/LCDCB2) to "1" (ON). The blinking interval can be selected from two options by using the BLSEL bit in the LCDC control register (LCDCC).

Note:

If the selected frame period generation clock halts during LCD display operation, the AC waveform generator circuit also halts and therefore a DC voltage is applied to the liquid crystal elements. In this case, the LCD display operation must be stopped in advance. The conditions under which the main clock (time-base timer) or sub clock (watch prescaler) halts depend on the selected clock mode and standby mode. The frame period is also affected if the time-base timer or watch prescaler is cleared depending on the setting of the frame period generation clock select bit (LCDCC:CSS).

■ LCD Drive Waveform

Due to the characteristics of the LCD, DC driving of the LCD chemically changes and degrades the liquid crystal display elements. Therefore, the LCD controller driver contains an AC waveform generator circuit to drive the LCD using a 2-frame alternating waveform. There are following three types of output waveform:

- 1/2 bias, 1/2 duty output waveforms
- 1/3 bias, 1/3 duty output waveforms
- 1/3 bias, 1/4 duty output waveforms

24.6.1 Output Waveform during LCD Controller Operation (1/2 Duty)

The display drive output is a multiplex drive type of 2-frame alternating waveform.
In 1/2 duty mode, only COM0 and COM1 are used for display. Neither COM2 nor COM3 is used.

■ 1/2 Bias, 1/2 Duty Output Waveform Example

Those liquid crystal elements are turned "ON" for display which has the maximum potential difference between the common and segment outputs.

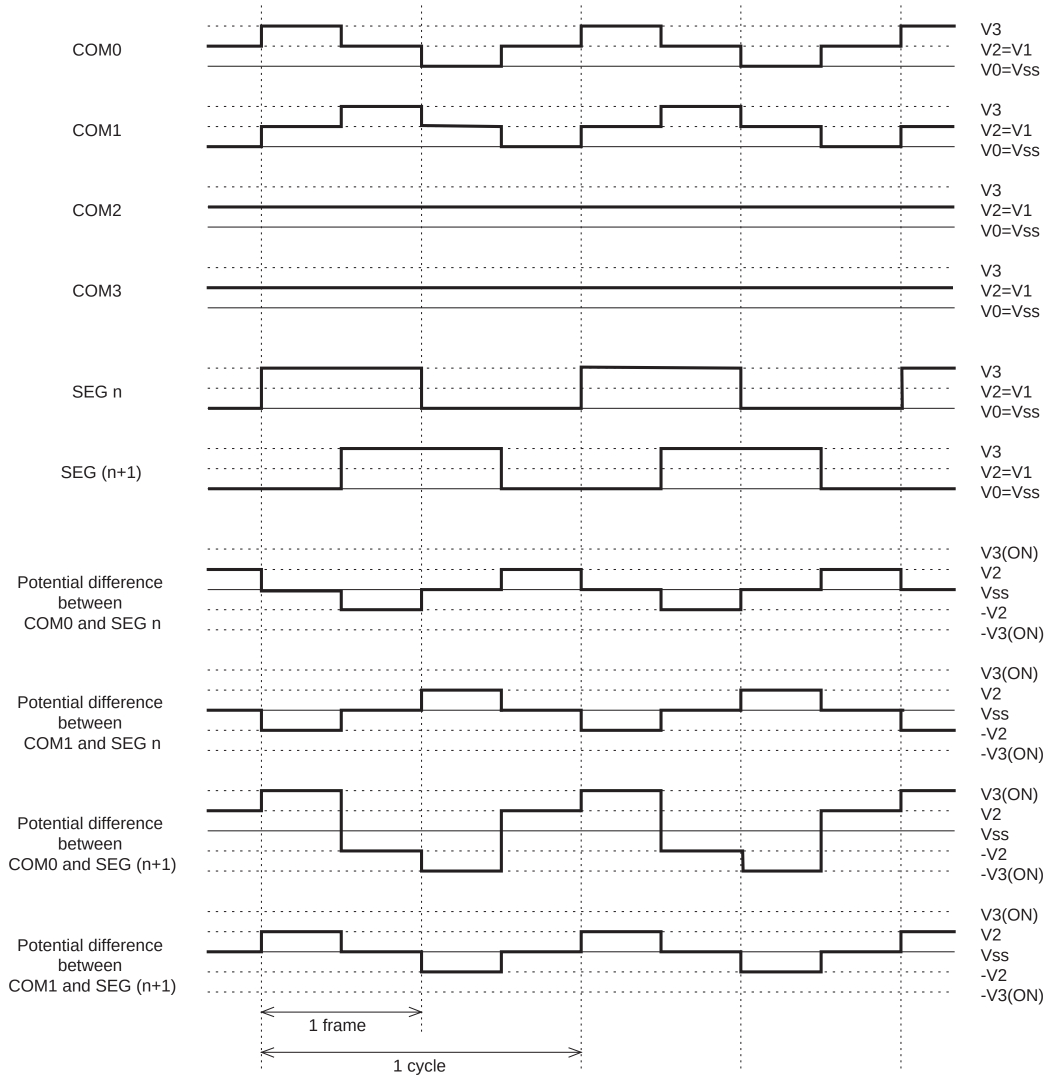
Figure 24.6-2 shows the output waveform when the contents of display RAM are those shown in Table 24.6-1.

Table 24.6-1 Sample Contents of Display RAM

Segment	Contents of Display RAM			
	COM3	COM2	COM1	COM0
SEG n	-	-	0	0
SEG (n+1)	-	-	0	1

-: Unused

Figure 24.6-2 1/2 Bias, 1/2 Duty Output Waveform Example



V_0 to V_3: Voltages at V0 to V3 pins

24.6.2 Output Waveform during LCD Controller Operation (1/3 Duty)

In 1/3 duty mode, COM0, COM1, and COM2 are used for display. COM3 is not used.

■ 1/3 Bias, 1/3 Duty Output Waveform Example

Those liquid crystal elements are turned "ON" for display which has the maximum potential difference between the common and segment outputs.

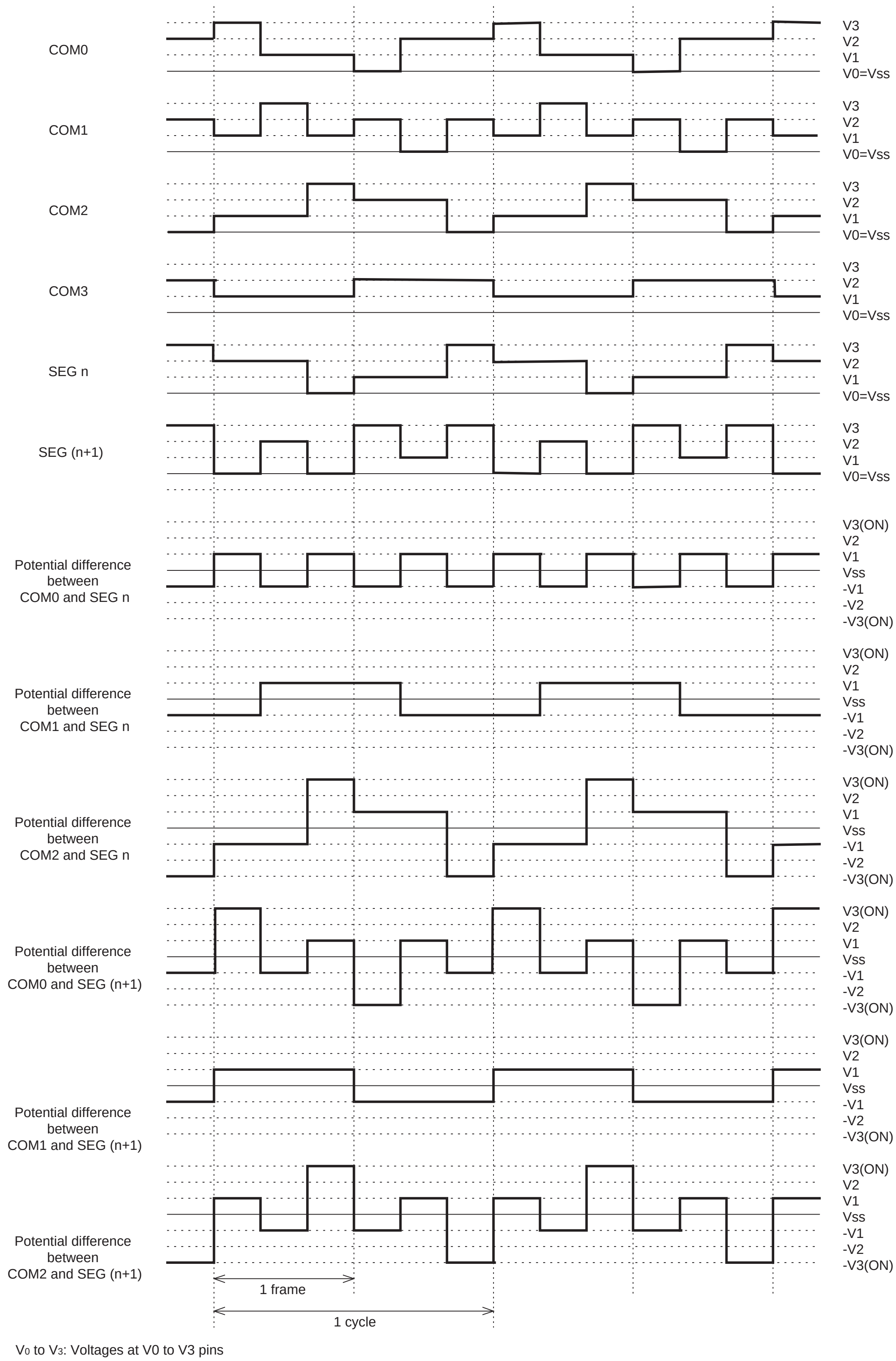
Figure 24.6-3 shows the output waveform when the contents of display RAM are those shown in Table 24.6-2.

Table 24.6-2 Sample Contents of Display RAM

Segment	Contents of Display RAM			
	COM3	COM2	COM1	COM0
SEG n	-	1	0	0
SEG (n+1)	-	1	0	1

-: Unused

Figure 24.6-3 1/3 Bias, 1/3 Duty Output Waveform Example



24.6.3 Output Waveform during LCD Controller Operation (1/4 Duty)

In 1/4 duty mode, all of COM0, COM1, COM2, and COM3 are used for display.

■ 1/3 Bias, 1/4 Duty Output Waveform Example

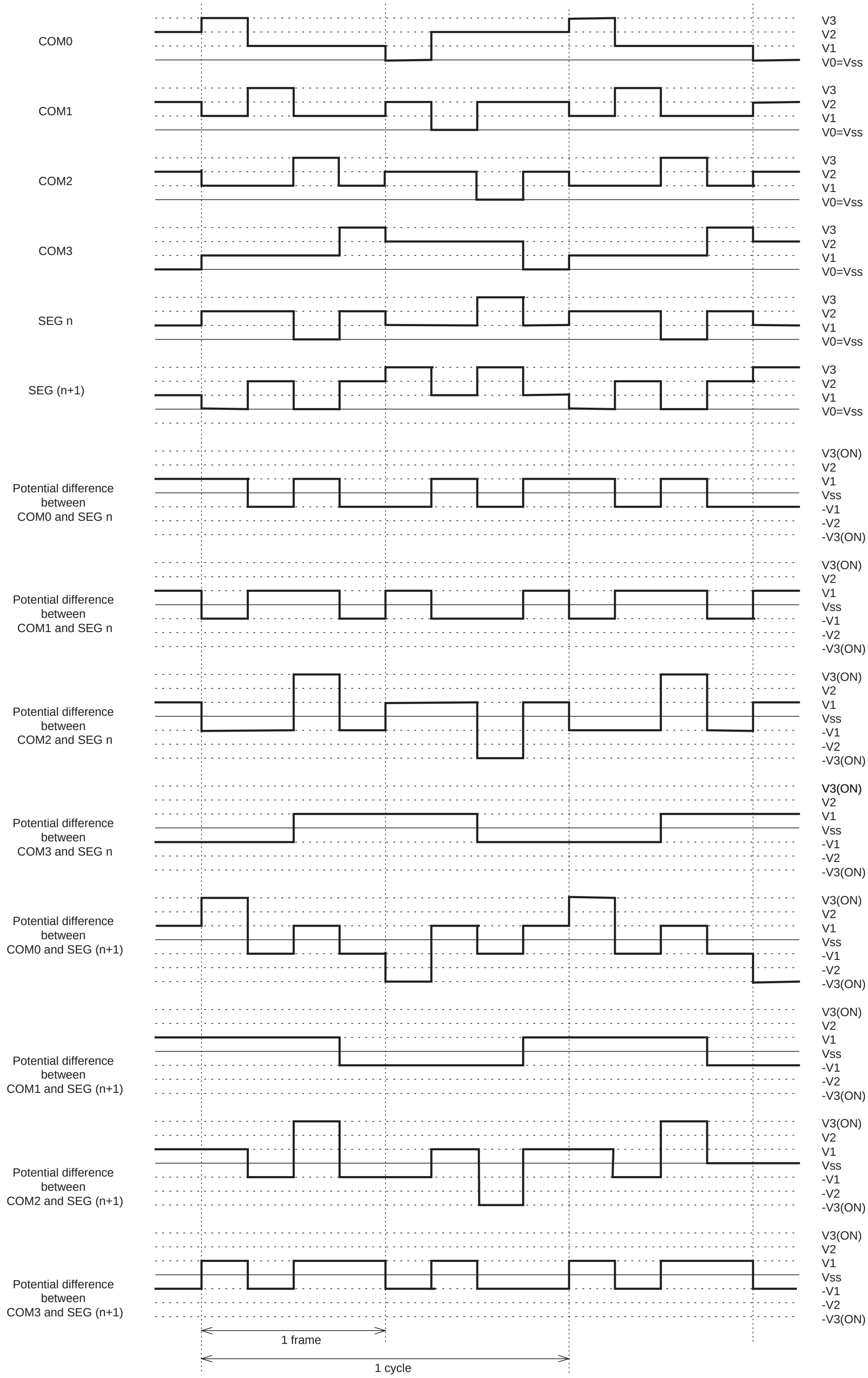
Those liquid crystal elements are turned "ON" for display which has the maximum potential difference between the common and segment outputs.

Figure 24.6-4 shows the output waveform when the contents of display RAM are those shown in Table 24.6-3.

Table 24.6-3 Sample Contents of Display RAM

Segment	Contents of Display RAM			
	COM3	COM2	COM1	COM0
SEG n	0	1	0	0
SEG (n+1)	0	1	0	1

Figure 24.6-4 1/3 Bias, 1/4 Duty Output Waveform Example



V_0 to V_3: Voltages at V0 to V3 pins

24.7 Notes on Use of LCD Controller

This section gives notes on using the LCD controller.

■ Notes on Use of LCD Controller

- To use LCD pins for ports, set the PICTL bit in LCDC enable register 1 (LCDCE1) to "1" and the corresponding select bits (COM/SEG) in LCDC enable registers 1 to 3 to "0".
- If the selected frame period generation clock halts during LCD display operation, the AC waveform generator circuit also halts and therefore a DC voltage is applied to the liquid crystal elements. In this case, the LCD display operation must be stopped in advance. The conditions under which the main clock (time-base timer) or sub clock (watch prescaler) halts depend on the selected clock mode and standby mode. The frame period is also affected if the time-base timer or watch prescaler is cleared depending on the setting of the frame period generation clock select bit (LCDCC:CSS).
- Output operation of display RAM data to the LCD is performed separately from the CPU access operation to display RAM. If the interval for re-writing display RAM is shorter than the LCD cycle, you may experience flickers caused by the different display patterns between frames.

CHAPTER 25

LOW-VOLTAGE DETECTION RESET CIRCUIT

This chapter describes the functions and operations of the low-voltage detection reset circuit.

25.1 Overview of Low-voltage Detection Reset Circuit

The low-voltage detection reset circuit monitors the power supply voltage and generates a reset signal if the voltage drops below the detection voltage level (available as an option to 5-V products only).

■ Low-voltage Detection Reset Circuit

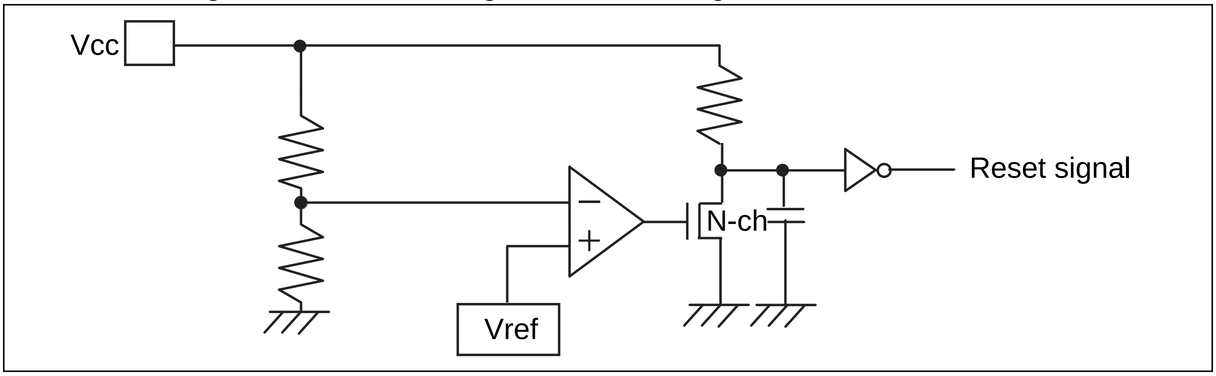
This circuit monitors the power supply voltage and generates a reset signal if the voltage drops below the detection voltage level. The circuit can be selected as an option to 5-V products only. Refer to the data sheet for details of the electrical characteristics.

25.2 Configuration of Low-voltage Detection Reset Circuit

Figure 25.2-1 is a block diagram of the low-voltage detection reset circuit.

■ Block Diagram of Low-voltage Detection Reset Circuit

Figure 25.2-1 Block Diagram of Low-voltage Detection Reset Circuit



25.3 Pins of Low-voltage Detection Reset Circuit

This section explains the pins of the low-voltage detection reset circuit.

■ Pins Related to Low-voltage Detection Reset Circuit

● Vcc pin

The low-voltage detection reset circuit monitors the voltage at this pin.

● Vss pin

This pin is a GND pin serving as the reference for voltage detection.

● $\overline{\text{RST}}$ pin

The low-voltage detection reset signal is output inside the microcontroller and to this pin.

However, for the model equipped with the clock supervisor function (see "1.2 Product Lineup of MB95150/M Series" for details), the low-voltage detection reset signal is generated only in the microcontroller and not output to this pin.

25.4 Operations of Low-voltage Detection Reset Circuit

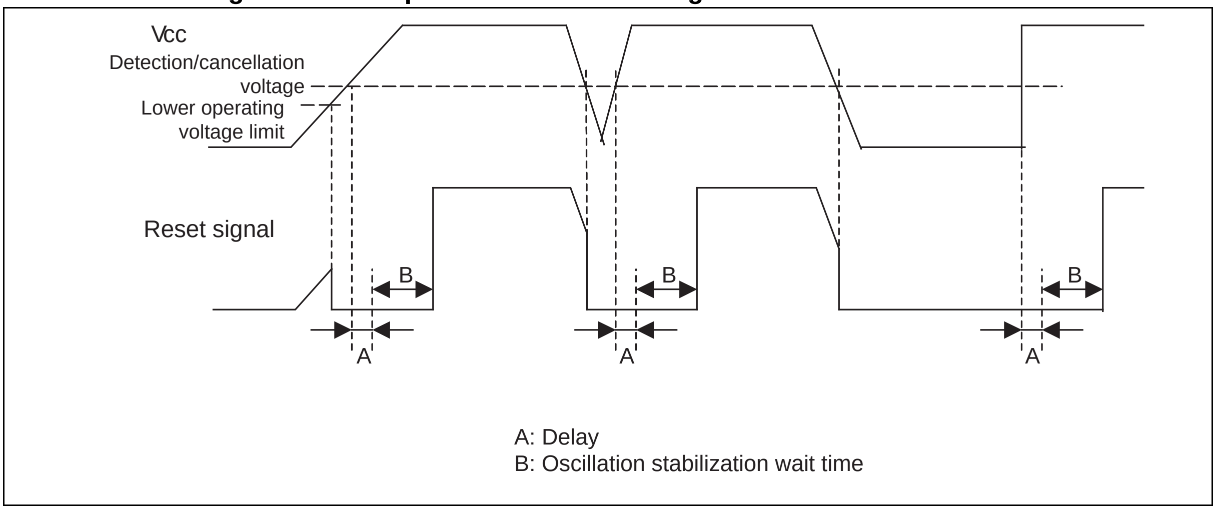
The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage.

■ Operations of Low-voltage Detection Reset Circuit

The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage. If the voltage is subsequently detected to have recovered, the circuit outputs a reset signal for the duration of the oscillation stabilization wait time to cancel the reset.

For details on the electrical characteristics, see the data sheet.

Figure 25.4-1 Operations of Low-voltage Detection Reset Circuit



■ Operations in Standby Mode

The low-voltage detection reset circuit remains operating even in standby modes (stop, sleep, sub clock, and watch modes).

CHAPTER 26
CLOCK SUPERVISOR

This chapter describes the functions and operations of the clock supervisor.

Code: CM26-00112-1E

26.1 Overview of Clock Supervisor

The clock supervisor prevents the situation which is out of control, when main clock and sub clock (only on dual clock products) oscillations have halted. This function switches to an CR clock generated in internal CR oscillator circuit, if main clock and sub clock oscillations have halted (this feature is optional to 5-V products).

■ Overview of Clock Supervisor

- The clock supervisor monitors the main clock and sub clock oscillations and generates an internal reset if it detects that the oscillation has halted. In this case, the clock supervisor switches to the internal CR clock (the clock frequency of the sub clock is equal to the CR clock frequency divided by 2).

  The reset source register (RSRR) can be used to determine whether a reset was triggered by the clock supervisor.
- A main clock oscillation halt is detected if the rising edge of the main clock is not detected for 4 CR clock cycles. The clock supervisor may detect incorrectly, if main clock is longer than 4 CR clock cycles.
- A sub clock oscillation halt is detected if the rising edge of the sub clock is not detected for 32 CR clock cycles. The clock supervisor may detect incorrectly, if sub clock is longer than 32 CR clock cycles.
- The clock supervisor can prohibit to monitor the main clock and sub clock respectively.
- If the sub clock is halted in the main clock mode, a reset does not occur immediately, but does occur after switching to the sub clock mode.
  Setting registers enable to prohibit the reset output.
- While the clock stops in main clock and sub clock stop modes, clock monitoring is disabled.
- This function can be selected as an option on 5-V products only.

Note:

Refer to the data sheet for the period and other details about the CR clock.

26.2 Configuration of Clock Supervisor

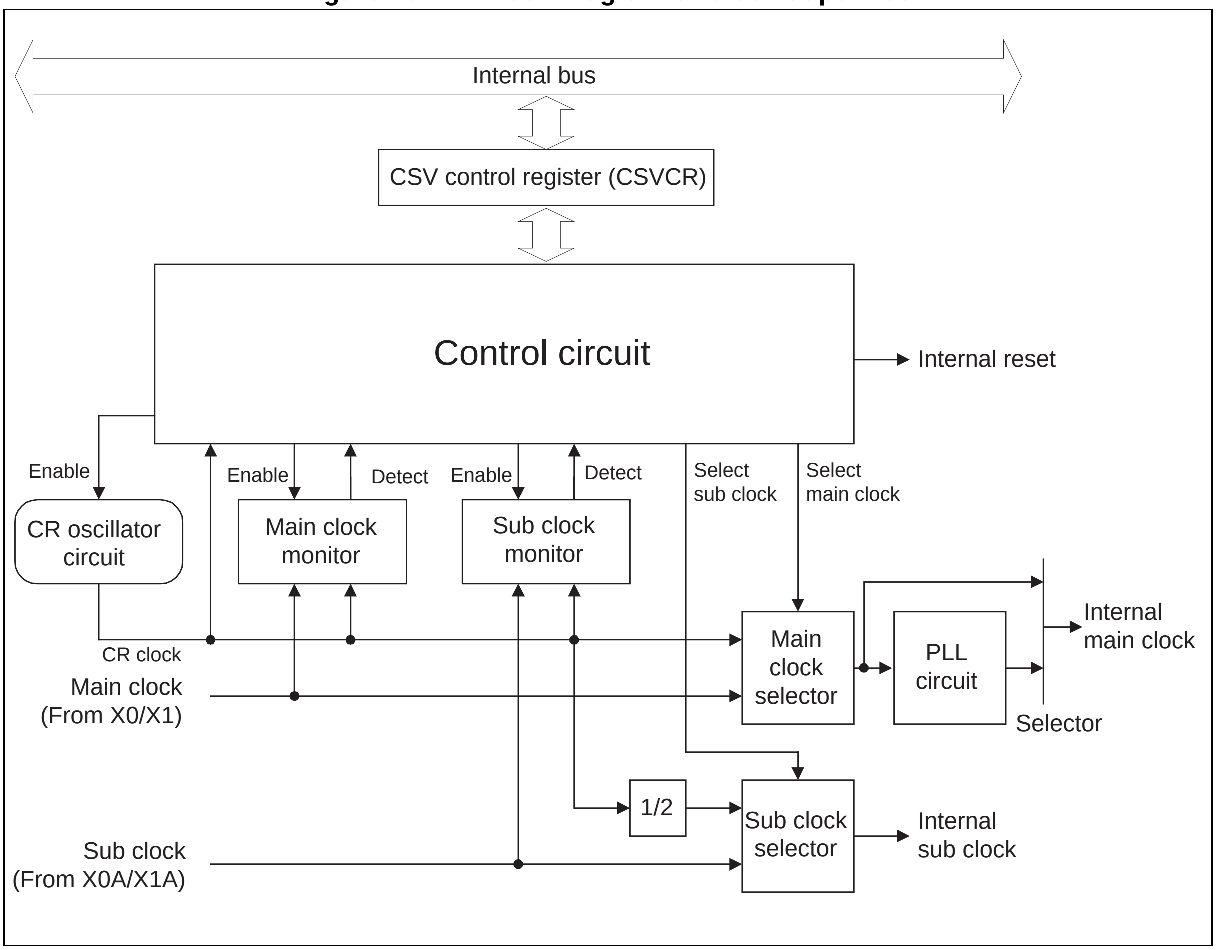
The clock supervisor consists of the following blocks:

- **Control circuit**
- **CR oscillator circuit**
- **Main clock monitor**
- **Sub clock monitor**
- **Main clock selector**
- **Sub clock selector**
- **CSV control register (CSVCR)**

■ Block Diagram of Clock Supervisor

Figure 26.2-1 shows a block diagram of the clock supervisor.

Figure 26.2-1 Block Diagram of Clock Supervisor



● Control circuit

This block controls the clocks, resets, and other settings based on the information in the CSV control register (CSVCR).

● CR oscillator circuit

This block is a internal CR oscillator circuit. The oscillation can be turned on or off via a control signal from the control circuit.

This also serves as an internal clock after a clock halt is detected.

● Main clock monitor

This block monitors whether the main clock halts.

● Sub clock monitor

This block monitors whether the sub clock halts.

● Main clock selector

This block outputs the CR clock as the internal main clock upon detection of a main clock halt.

● Sub clock selector

This block outputs the clock obtained by dividing the CR clock as the internal sub clock upon detection of a sub clock halt.

● CSV control register (CSVCR)

This block is used to control clock monitoring and CR clock and to check information on halt detection.

26.3 Register of Clock Supervisor

This section describes the clock supervisor registers.

■ Register of Clock Supervisor

Figure 26.3-1 shows the register of the clock supervisor.

Figure 26.3-1 Clock Supervisor Register

Clock supervisor control register (CSVCR)

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
Address 000FEA$_H$	Reserved	MM	SM	RCE	MSVE	SSVE	SRST	Reserved	Initial value 00011100$_B$
	R/W	R	R	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable
R : Read only

26.3.1 Clock Supervisor Control Register (CSVCR)

The clock supervisor control register (CSVCR) is used to enable the various functions and to check the status.

■ Clock Supervisor Control Register (CSVCR)

Figure 26.3-2 Clock Supervisor Control Register (CSVCR)

Address 000FEA$_H$ — Initial value 00011100$_B$

bit	7	6	5	4	3	2	1	0
	Reserved	MM	SM	RCE	MSVE	SSVE	SRST	Reserved
	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Reserved	Reserved bit
0	Be sure to set this bit to "0".

SRST	Reset generation enable bit *
0	Disables reset generation.
1	Enables reset generation.

*: Assuming that a sub clock halt has been already detected at transition from main clock mode to sub clock mode.

SSVE	Sub clock monitoring enable bit
0	Disables sub clock monitoring.
1	Enables sub clock monitoring.

MSVE	Main clock monitoring enable bit
0	Disables main clock monitoring.
1	Enables main clock monitoring.

RCE	CR clock oscillation enable bit
0	Disables CR clock oscillation.
1	Enables CR clock oscillation.

SM	Sub clock halt detection bit
0	Sub clock halt not detected.
1	Sub clock halt detected.

MM	Main clock halt detection bit
0	Main clock halt not detected.
1	Main clock halt detected.

Reserved	Reserved bit
0	Be sure to set this bit to "0".

R/W : Readable/writable
R : Read only
Reserved: Reserved bit
▭ : Initial value

Table 26.3-1 Functions of Bits in Clock Supervisor Control Register (CSVCR)

Bit name		Function
bit7	Reserved bit	This bit is reserved. Write "0" to this bit. The read value is always "0".
bit6	MM: Main clock halt detection bit	This bit is read-only, and this bit indicates that a main clock oscillation halt has been detected. **When set to "0":** The bit indicates that no main clock oscillation halt has been detected. **When set to "1":** The bit indicates that main clock oscillation halt has been detected. Writing "1" to this bit does not affect the operation.
bit5	SM: Sub clock halt detection bit	This bit is read-only, and this bit indicates that a sub clock oscillation halt has been detected. **When set to "0":** The bit indicates that no sub clock oscillation halt has been detected. **When set to "1":** The bit indicates that sub clock oscillation halt has been detected. Writing "1" to this bit does not affect the operation.
bit4	RCE: CR clock oscillation enable bit	This bit enables CR oscillation. **When set to "0":**The bit disables oscillation. **When set to "1":** The bit enables oscillation (initial value). Before writing "0" to this bit, make sure that the clock monitor function has been disabled with the MM and SM bits set to "0".
bit3	MSVE: Main clock monitoring enable bit	This bit enables the monitoring of main clock oscillation. **When set to"0":** The bit disables main clock monitoring. **When set to"1":** The bit enables main clock monitoring. This bit is set to "1" only when a power-on reset occurs.
bit2	SSVE: Sub clock monitoring enable bit	This bit enables the monitoring of sub clock oscillation. **When set to"0":** The bit disables sub clock monitoring. **When set to"1":** The bit enables sub clock monitoring. This bit is set to "1" only when a power-on reset occurs.
bit1	SRST: Reset generation enable bit	This bit enables reset output upon transition to sub mode. **When set to "0":** The bit prevents a reset upon transition to sub clock mode with the sub clock halted in main clock mode. **When set to "1":** The bit causes a reset upon transition to sub clock mode with the sub clock halted in main clock mode.
bit0	Reserved bit	This bit is reserved. Write "0" to this bit.The read value is always "0".

Note:

When the power is turned on, the clock supervisor starts monitoring after the oscillation stabilization wait time for the main clock elapses. The oscillation stabilization wait time of the main clock must therefore be longer than the time required for the clock supervisor to start operating.

26.4 Operations of Clock Supervisor

This section describes the operations of the clock supervisor.

■ Operations of Clock Supervisor

The clock supervisor monitors the main clock and sub clock oscillations. If main clock and sub clock oscillations have halted, the device switches to an CR clock and generates a reset.

The following describes the operation in each clock mode.

● Main clock oscillation halt in main clock mode

The clock supervisor detect that main clock oscillation has halted, if no rising edge is detected on the main clock for 4 CR clock cycles in main clock mode.

If a main clock halt is detected, a reset is generated and the main clock switches to the CR clock.

The clock supervisor may detect incorrectly, if main clock is a low speed (longer than 4 CR clock cycles). It results from using the CR clock for detecting that main clock oscillation have halted.

The clock supervisor does not detect the main clock during stop mode.

● Sub clock oscillation halt in main clock mode (only on dual clock products)

In main clock mode, the condition used to detect the sub clock oscillation as having halted is that no rising edge is detected on the sub clock for 32 CR clock cycles.

Although no reset is generated immediately if a sub clock halt is detected in main clock mode, the sub clock switches to CR clock divided by two.

A reset can be generated when the device switches from main clock mode to sub clock mode with a sub clock oscillation halt detected, by setting the SRST bit in the clock supervisor control register (CSVCR).

As the CR clock is used to detect whether the sub clock has halted, a sub clock halt may be detected if the sub clock is set to a low speed (period longer than 32 CR clock cycles).

The clock supervisor does not detect the sub clock during the stop mode.

● Sub clock oscillation halt in sub clock mode (only on dual clock products)

In sub clock mode, the condition used to detect the sub clock oscillation as having halted is that no rising edge is detected on the sub clock for 34 CR clock cycles.

If a sub clock halt is detected, a reset is generated and the device enters main clock mode. In this case, the sub clock switches to CR clock divided by two.

As the CR clock is used to detect whether the sub clock has halted, a sub clock halt may be detected if the sub clock is set to a low speed (period longer than 32 CR clock cycles).

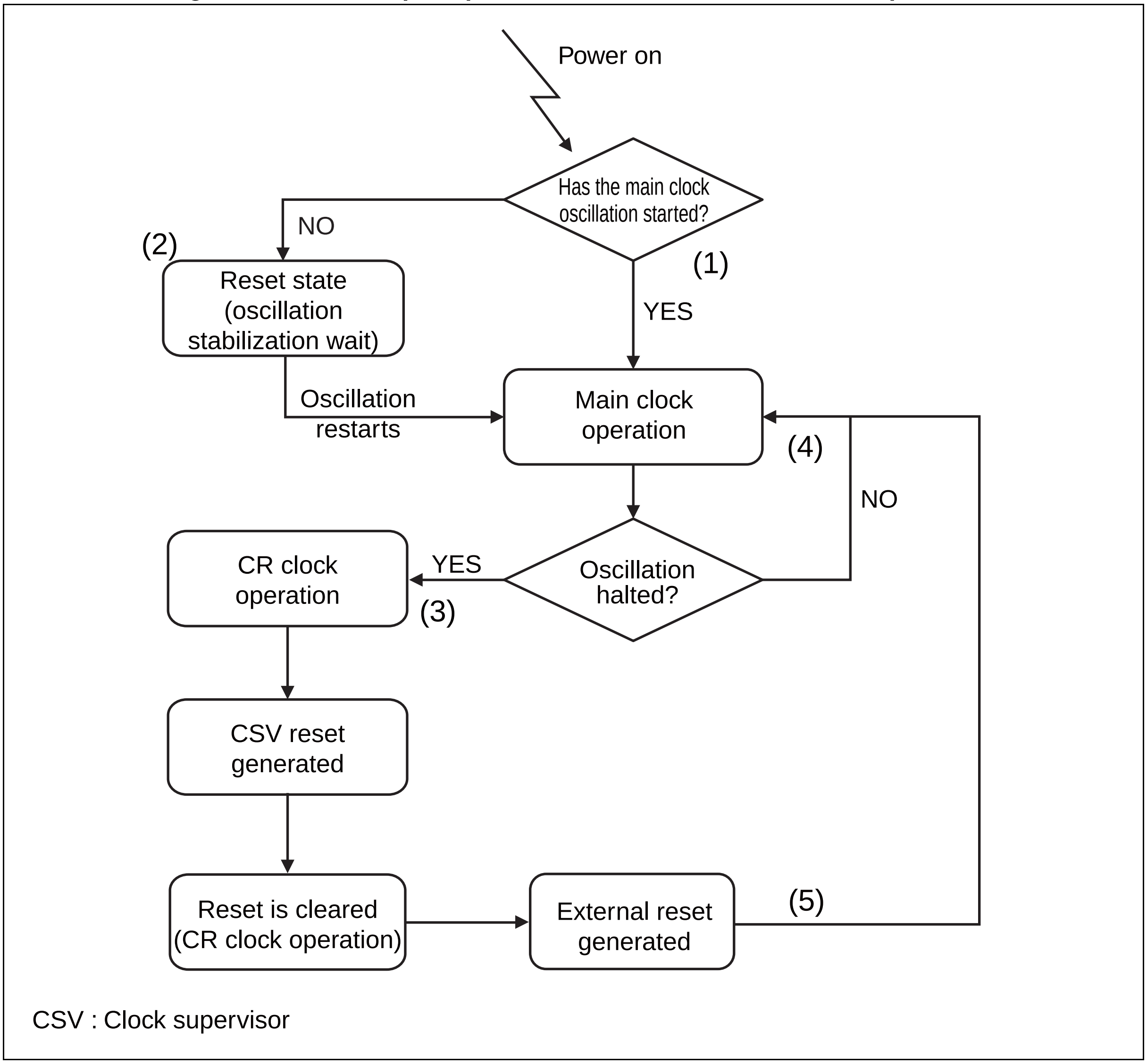
The clock supervisor does not detect the sub clock during the stop mode.

● Main clock oscillation halt in sub clock mode (only on dual clock products)

In sub clock mode, the main clock oscillation remains halted and is therefore not detected by the clock supervisor.

■ Example Operation Flowchart for the Clock Supervisor

Figure 26.4-1 Example Operation Flowchart for the Clock Supervisor



(1) After the power is turned on, the main clock operation starts after the oscillation stabilization wait time generated by the main clock oscillation has elapsed.

(2) If the main clock halts at power on, the device remains in the reset state (oscillation stabilization wait state). The operation changes to the main clock, after the oscillation restarts and the oscillation stabilization wait time elapsed.

(3) If an oscillation halt is detected during main clock operation, the operating clock is switched to the CR clock and a reset is generated.

(4) If the main oscillation continues (oscillation does not halt), the device continues to run using the main clock.

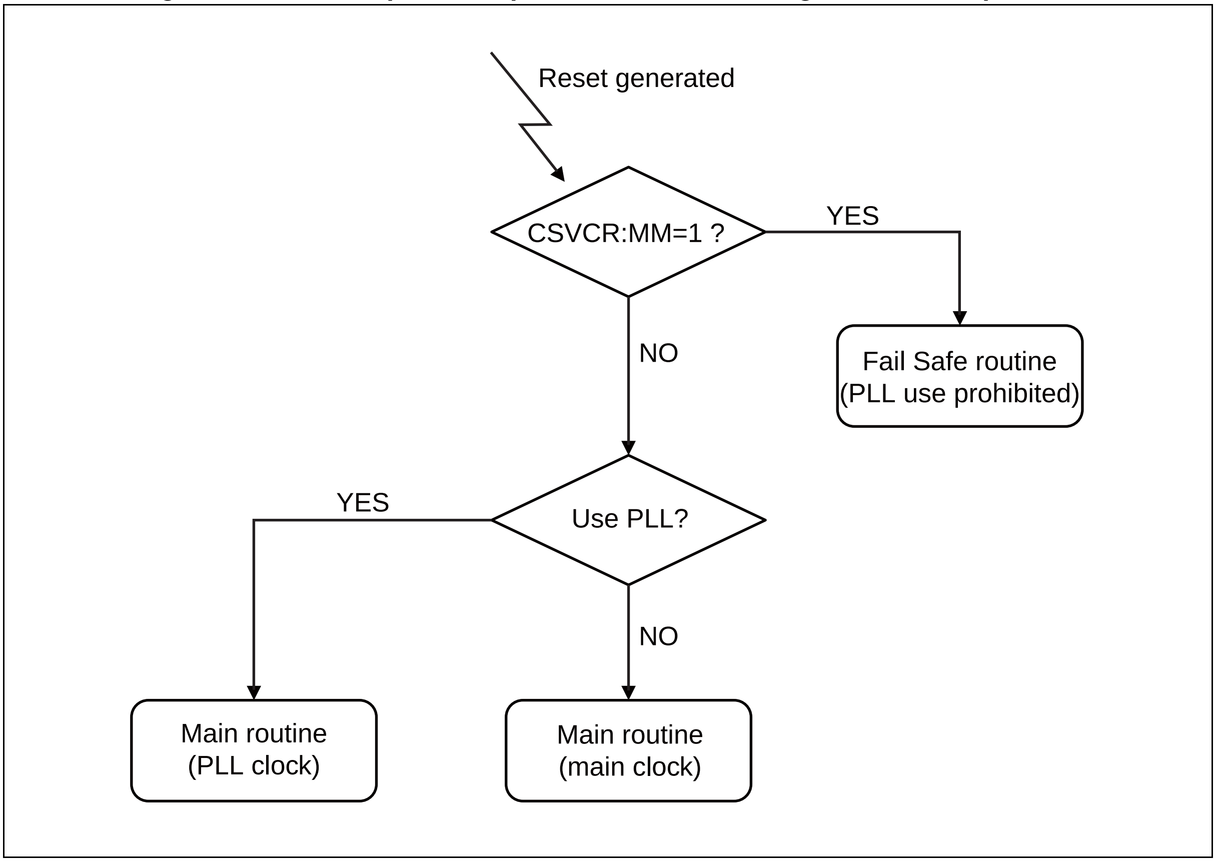
(5) If an external reset occurs during the CR clock operation, operation changes to the main clock. However, if the oscillation is halted at this time, another CSV reset is generated and the device returns to CR clock operation.

■ Example Startup Flowchart when using the Clock Supervisor

Inserting checking process of the main clock stop detection bit (CSVCR:MM) enables user programs to control the Fail Safe routine.

Figure 26.4-2 shows the example startup flowchart when using the clock supervisor.

Figure 26.4-2 Example Startup Flowchart when using the Clock Supervisor



26.5 Notes on Using Clock Supervisor

Take note of the following points when using the clock supervisor.

■ Notes on Using Clock Supervisor

Points to Note when using the Clock Supervisor

- Operation of the clock supervisor at power on

  When the power is turned on, the clock supervisor starts monitoring after the oscillation stabilization wait time for the main clock has elapsed. Therefore, unless the operation continues for longer than the oscillation stabilization wait time for the main clock, the clock supervisor will not operate.

- Transition to CR clock mode

  Do not turn on the PLL after changing to CR clock mode.

  As the frequency is below the lower limit for the input frequency of the PLL circuit, the PLL operation will not be guaranteed.

- Disabling the CR oscillation

  Do not use the CR oscillation enable bit (CSVCR:RCE) to disable the CR oscillation during CR clock mode.

  As this halts the internal clock, it may result in deadlock.

- Initializing the main clock halt detection bit

  The main clock halt detection bit (CSVCR:MM) is initialized by a power-on reset or external reset only. The bit is not initialized by the watchdog timer reset/software reset/ CSV reset. Accordingly, the device remains in CR clock mode if one of these resets occurs during CR clock mode.

CHAPTER 27
256-Kbit FLASH MEMORY

This chapter describes the functions and operations of 256-Kbit flash memory.

27.1 Overview of 256-Kbit Flash Memory

256-Kbit flash memory is located from 8000_H to $FFFF_H$ on the CPU memory map. The function of the flash memory interface circuit provides read access and program access from the CPU to flash memory.

■ Overview of 256-Kbit Flash Memory

The following methods can be used to program (write) and erase data into/from flash memory:

- Programming/erasing using a parallel writer
- Programming/erasing using a dedicated serial writer
- Programming/erasing by program execution

As flash memory by program execution can be programmed and erased by the instructions from the CPU via the flash memory interface circuit, you can efficiently reprogram (update) program code and data in flash memory with the device mounted on a circuit board.

■ Features of 256-Kbit Flash Memory

- 32K bytes × 8 bits sector configuration
- Automatic program algorithm (Embedded Algorithm)
- Detection of completion of programming/erasing using the data polling or toggle bit function
- Detection of completion of programming/erasing by CPU interrupts
- Compatible with JEDEC standard commands
- Programming/erase count (minimum): 10,000 times

■ Programming and Erasing Flash Memory

- It is not possible to write to and read from flash memory at the same time.
- To program/erase data into/from flash memory, first copy the program on the flash memory to RAM, and then execute the copied program on RAM, so that writing to the flash memory can be performed.

MB95150/M Series

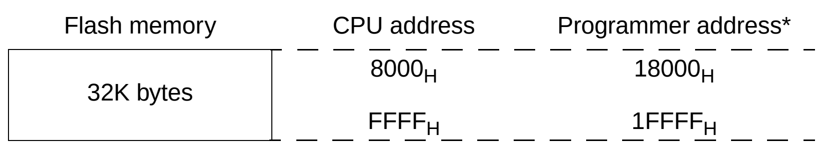
27.2 Sector Configuration of Flash Memory

This section explains the sector configuration of flash memory.

■ Sector Configuration of 256-Kbit Flash Memory

Figure 27.2-1 shows the sector configuration of the 256-Kbit flash memory. The upper and lower addresses of each sector are given in the figure.

Figure 27.2-1 Sector Configuration of 256-Kbit Flash Memory



*: The programmer address is equivalent to the CPU address, which is used when data is written to the flash memory using parallel programmer.
When a parallel programmer is used for programming/erasing, the programmer address is used for programming/erasing.

27.3 Register of Flash Memory

This section shows the register of the flash memory.

■ Register of the Flash Memory

Figure 27.3-1 Register of the Flash Memory

Flash Memory Status Register (FSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072_H	–	–	RDYIRQ	RDY	Reserved	IRQEN	WRE	Reserved	$000X0000_B$
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W0	

R/W : Readable/writable (Read value is the same as write value)
R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R/W0 : Reserved bit (Write value is "0", read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
X : Indeterminate

27.3.1 Flash Memory Status Register (FSR)

Figure 27.3-2 lists the functions of the flash memory status register (FSR).

■ Flash Memory Status Register (FSR)

Figure 27.3-2 Flash Memory Status Register (FSR)

Address	bit7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0072H	-	-	RDYIRQ	RDY	Reserved	IRQEN	WRE	Reserved	000X0000B
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W0	

Reserved	Reserved bit
0	Be sure to set the bit to "0".

WRE	Flash memory program/erase enable bit
0	Disables flash memory area programming/erasing.
1	Enables flash memory area programming/erasing.

IRQEN	Flash memory program/erase interrupt enable bit
0	Disables interrupts upon completion of programming/erasing.
1	Enables interrupts upon completion of programming/erasing.

Reserved	Reserved bit
0	Be sure to set the bit to "0".

RDY	Flash memory program/erase status bit
0	Data is being programmed/erased (not ready to program/erase next data).
1	Data has been programmed/erased (ready to program/erase next data).

RDYIRQ	Flash memory operation flag bit	
	Read	Write
0	Programming/erasing is being performed.	Clears this bit.
1	Programming/erasing has been completed.	No effect.

Undefined bit
The value read is always "0". Writing has no effect on the operation.

Undefined bit
The value read is always "0". Writing has no effect on the operation.

R/W : Readable/writable (Read value is the same as write value)
R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX : Read only (Readable, writing has no effect on operation)
R/W0 : Reserved bit (Write value is "0", read value is the same as write value)
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)
X : Indeterminate
(shaded box) : Initial value

Table 27.3-1 Functions of Flash Memory Status Register (FSR)

Bit name		Function
bit7, bit6	-: Undefined bits	The value read is always "0". Writing has no effect on the operation.
bit5	RDYIRQ: Flash memory operation flag bit	This bit shows the operating state of flash memory. The RDYIRQ bit is set to "1" upon completion of the flash memory automatic algorithm when flash memory programming/erasing is completed. • An interrupt request occurs when the RDYIRQ bit is set to "1" if interrupts triggered by the completion of flash memory programming/erasing have been enabled (FSR: IRQEN=1). • If the RDYIRQ bit is set to "0" when flash memory programming/erasing is completed, further flash memory programming/erasing is disabled. **Setting the bit to "0":** Clears the bit. **Setting the bit to "1":** Has no effect on the operation. "1" is read from the bit whenever a read-modify-write (RMW) instruction is used.
bit4	RDY: Flash memory program/erase status bit	This bit shows the programming/erasing status of flash memory. • Flash memory programming/erasing cannot be performed with the RDY bit set to "0". • A read/reset command can be accepted even when the RDY bit contains "0". The RDY bit is set to "1" upon completion of programming/erasing. • It takes a delay of two machine clock (MCLK) cycles after the issuance of a program/erase command for the RDY bit to be set to "0". Read this bit after, for example, inserting NOP twice after issuing the program/erase command.
bit3	Reserved: Reserved bit	Be sure to set this bit to "0".
bit2	IRQEN: Flash memory program/erase interrupt enable bit	This bit enables or disables the generation of interrupt requests in response to the completion of flash memory programming/erasing. **Setting the bit to "0":** Prevents an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1). **Setting the bit to "1":** Causes an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1).
bit1	WRE: Flash memory program/erase enable bit	This bit enables or disables the programming/erasing of data into/from the flash memory area. Set the WRE bit before invoking a flash memory program/erase command. **Setting the bit to "0":** Prevents a program/erase signal from being generated even when a program/erase command is input. **Setting the bit to "1":** Allows flash memory programming/erasing to be performed after a program/erase command is input. When flash memory is not to be programmed or erased, set the WRE bit to "0" to prevent it from being accidentally programmed or erased.
bit0	Reserved: Reserved bit	Be sure to set this bit to "0".

27.4 Starting the Flash Memory Automatic Algorithm

There are three types of commands that invoke the flash memory automatic algorithm: read/reset, write (program), and chip-erase.

■ Command Sequence Table

Table 27.4-1 shows commands used for programming/erasing data on flash memory.

Table 27.4-1 Command Sequence Table

Command sequence	Bus write cycle	1st bus write cycle		2nd bus write cycle		3rd bus write cycle		4th bus write cycle		5th bus write cycle		6th bus write cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/ reset*	1	F_XXX_H	$F0_H$	-	-	-	-	-	-	-	-	-	-
	4	$UAAA_H$	AA_H	$U554_H$	55_H	$UAAA_H$	$F0_H$	RA	RD	-	-	-	-
Programming	4	$UAAA_H$	AA_H	$U554_H$	55_H	$UAAA_H$	$A0_H$	PA	PD	-	-	-	-
Chip erasing	6	$XAAA_H$	AA_H	$X554_H$	55_H	$XAAA_H$	80_H	$XAAA_H$	AA_H	$X554_H$	55_H	$XAAA_H$	10_H

- RA : Read address
- PA : Write (program) address
- RD : Read data
- PD : Write (program) data
- U : Upper 4 bits same as RA and PA
- F_X : FF/FE
- X : Arbitrary address

*: Both of the two types of read/reset command can reset the flash memory to read mode.

Notes:

- Addresses in the table are the values in the CPU memory map. All addresses and data are hexadecimal values. However, "X" is an arbitrary value.
- Address "U" in the table is not arbitrary, whose four bits (bit15 to bit12) must have the same value as RA and PA.

Example: If RA = $C48E_H$, U = C; If PA = 1024_H, U=1

■ Notes on Issuing Commands

Pay attention to the following points when issuing commands in the command sequence table:

The upper address U bits (bit15 to bit12) used when commands are issued must have the same value as RA and PA, from the first command on.

If the above measures are not followed, commands are not recognized normally. Execute a reset to initialize the command sequencer in the flash memory.

27.5 Checking the Automatic Algorithm Execution Status

As the flash memory uses the automatic algorithm for a process flow for programming/erasing, you can check its internal operating status with hardware sequence flags.

■ Hardware Sequence Flag

● Overview of hardware sequence flag

The hardware sequence flag consists of the following 3-bit outputs:

- Data Polling Flag (DQ7)
- Toggle Bit Flag (DQ6)
- Execution Time-out Flag (DQ5)

The hardware sequence flags tell whether the write (program) or chip-erase command has been terminated and whether an erase code write can be performed.

You can reference hardware sequence flags by read access to the address of each relevant sector in flash memory after setting a command sequence.

Table 27.5-1 shows the bit allocation of the hardware sequence flags.

Table 27.5-1 Bit Allocation of Hardware Sequence Flags

Bit No.	7	6	5	4	3	2	1	0
Hardware sequence flag	DQ7	DQ6	DQ5	–	–	–	–	–

- To know whether the automatic write, chip-erase, command is being executed or has been terminated, check the hardware sequence flags or the flash memory program/erase status bit in the flash memory status register (FSR:RDY). After programming/erasing is terminated, flash memory returns to the read/reset state.
- When creating a write/erase program, read data after checking the termination of automatic writing/erasing with the DQ7, DQ6, and DQ5 flags.

● Explanation of hardware sequence flag

Table 27.5-2 lists the functions of the hardware sequence flag.

Table 27.5-2 List of Hardware Sequence Flag Functions

State		DQ7	DQ6	DQ5
State transition during normal operation	Programming → Programming completed (when write address has been specified)	$\overline{DQ7}$ → DATA: 7	Toggle → DATA: 6	0 → DATA: 5
	Chip erasing → Erasing completed	0 → 1	Toggle → Stop	0 → 1
Abnormal operation	Write	$\overline{DQ7}$	Toggle	1
	Chip erase	0	Toggle	1

27.5.1 Data Polling Flag (DQ7)

The data polling flag (DQ7) is a hardware sequence flag used to indicate that the automatic algorithm is being executing or has been completed using the data polling function.

■ Data Polling Flag (DQ7)

Table 27.5-3 and Table 27.5-4 show the state transition of the data polling flag.

Table 27.5-3 State Transition of Data Polling Flag (During Normal Operation)

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ7	$\overline{DQ7}$ → DATA: 7	0 → 1

Table 27.5-4 State Transition of Data Polling Flag (During Abnormal Operation)

Operating state	Programming	Chip erase
DQ7	$\overline{DQ7}$	0

● At programming

When read access takes place during execution of the automatic write algorithm, the flash memory outputs the inverted value of bit7 in the last data written to DQ7.

If read access takes place on completion of the automatic write algorithm, the flash memory outputs bit7 of the value read from the read-accessed address to DQ7.

● At chip erasing

When read access is made to the sector currently being erased during execution of the chip erase automatic algorithm, bit7 of flash memory outputs "0". Bit7 of flash memory outputs "1" upon completion of chip erasing.

Note:

Once the automatic algorithm has been started, read access to the specified address is ignored. Data reading is allowed after the data polling flag (DQ7) is set to "1". Data reading after the end of the automatic algorithm should be performed following read access made to confirm the completion of data polling.

27.5.2 Toggle Bit Flag (DQ6)

The toggle bit flag (DQ6) is a hardware sequence flag used to indicate that the automatic algorithm is being executed or has been completed using the toggle bit function.

■ Toggle Bit Flag (DQ6)

Table 27.5-5 and Table 27.5-6 show the state transition of the toggle bit flag.

Table 27.5-5 State Transition of Toggle Bit Flag (During Normal Operation)

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ6	Toggle → DATA: 6	Toggle → Stop

Table 27.5-6 State Transition of Toggle Bit Flag (During Abnormal Operation)

Operating state	Programming	Chip erase
DQ6	Toggle	Toggle

● At programming and chip erasing

- When read access is made continuously during execution of the automatic write algorithm or chip-erase/sector-erase algorithm, the flash memory toggles the output between "1" and "0" at each read access.
- When read access is made continuously after the automatic write algorithm or chip-erase/ sector-erase algorithm is terminated, the flash memory outputs bit6 (DATA:6) of the value read from the read address at each read access.

27.5.3 Execution Time-out Flag (DQ5)

The execution time-out flag (DQ5) is a hardware sequence flag indicating that the automatic algorithm has been executed beyond the specified time (required for programming/erasing) internal to the flash memory.

■ Execution Time-out Flag (DQ5)

Table 27.5-7 and Table 27.5-8 show the state transition of the execution time-out flag.

Table 27.5-7 State Transition of Execution Time-out Flag (During Normal Operation)

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ5	0 → DATA: 5	0 → 1

Table 27.5-8 State Transition of Execution Time-out Flag (During Abnormal Operation)

Operating state	Programming	Chip erase
DQ5	1	1

● At programming and chip erasing

When read access is made with the write or chip-erase automatic algorithm invoked, the flag outputs "0" when the algorithm execution time is within the specified time (required for programming/erasing) or "1" when it exceeds that time.

The execution time-out flag (DQ5) can be used to check whether programming/erasing has succeeded or failed regardless of whether the automatic algorithm has been running or terminated. When the execution time-out flag (DQ5) outputs "1", it indicates that programming has failed if the automatic algorithm is still running for the data polling or toggle bit function.

If an attempt is made to write "1" to a flash memory address holding "0", for example, the flash memory is locked, preventing the automatic algorithm from being terminated and valid data from being output from the data polling flag (DQ7). As the toggle bit flag (DQ6) does not stop toggling, the time limit is exceeded and the execution time-out flag (DQ5) outputs "1". The state in which the execution time-out flag (DQ5) outputs "1" means that the flash memory has not been used correctly; it does not mean that the flash memory is defective. When this state occurs, execute the reset command.

27.6 Flash Memory Program/Erase

This section describes the individual procedures for flash memory reading/ resetting, programming, and chip-erasing by entering their respective commands to invoke the automatic algorithm.

■ Details of Programming/Erasing Flash Memory

The automatic algorithm can be invoked by writing the read/reset, program, and chip-erase command sequence to flash memory from the CPU. Writing command sequence to flash memory from the CPU must always be performed continuously. The termination of the automatic algorithm can be checked by the data polling function. After the automatic algorithm terminates normally, the flash memory returns to the read/reset state.

The individual operations are explained in the following order:

- Enter read/reset state.
- Program data.
- Erase all data (chip-erase).

27.6.1 Placing Flash Memory in the Read/Reset State

This section explains the procedure for entering the read/reset command to place flash memory in the read/reset state.

■ Placing Flash Memory in the Read/Reset State

- To place flash memory in the read/reset state, send the read/reset command in the command sequence table continuously from the CPU to flash memory.
- The read/reset command is available in two different command sequences: one involves a single bus operation and the other involves four bus operations, which are essentially the same.
- Since the read/reset state is the initial state of flash memory, the flash memory always enters this state after the power is turned on and at the normal termination of a command. The read/reset state is also described as the wait state for command input.
- In the read/reset state, read access to flash memory enables data to be read. As is the case with masked ROM, program access from the CPU can be made.
- Read access to flash memory does not require the read/reset command. If a command is not terminated normally, use the read/reset command to initialize the automatic algorithm.

27.6.2 Programming Data into Flash Memory

This section explains the procedure for entering the write (program) command to program data into flash memory.

■ Programming Data into Flash Memory

- To start the automatic algorithm for programming data into flash memory, send the program command in the command sequence table continuously from the CPU to flash memory.
- Upon completion of data programming to a target address in the fourth cycle, the automatic algorithm starts automatic programming.

● How to specify addresses

Programming (writing) can be performed even in any order of addresses or across a sector boundary. Data written by a single program command is only one byte.

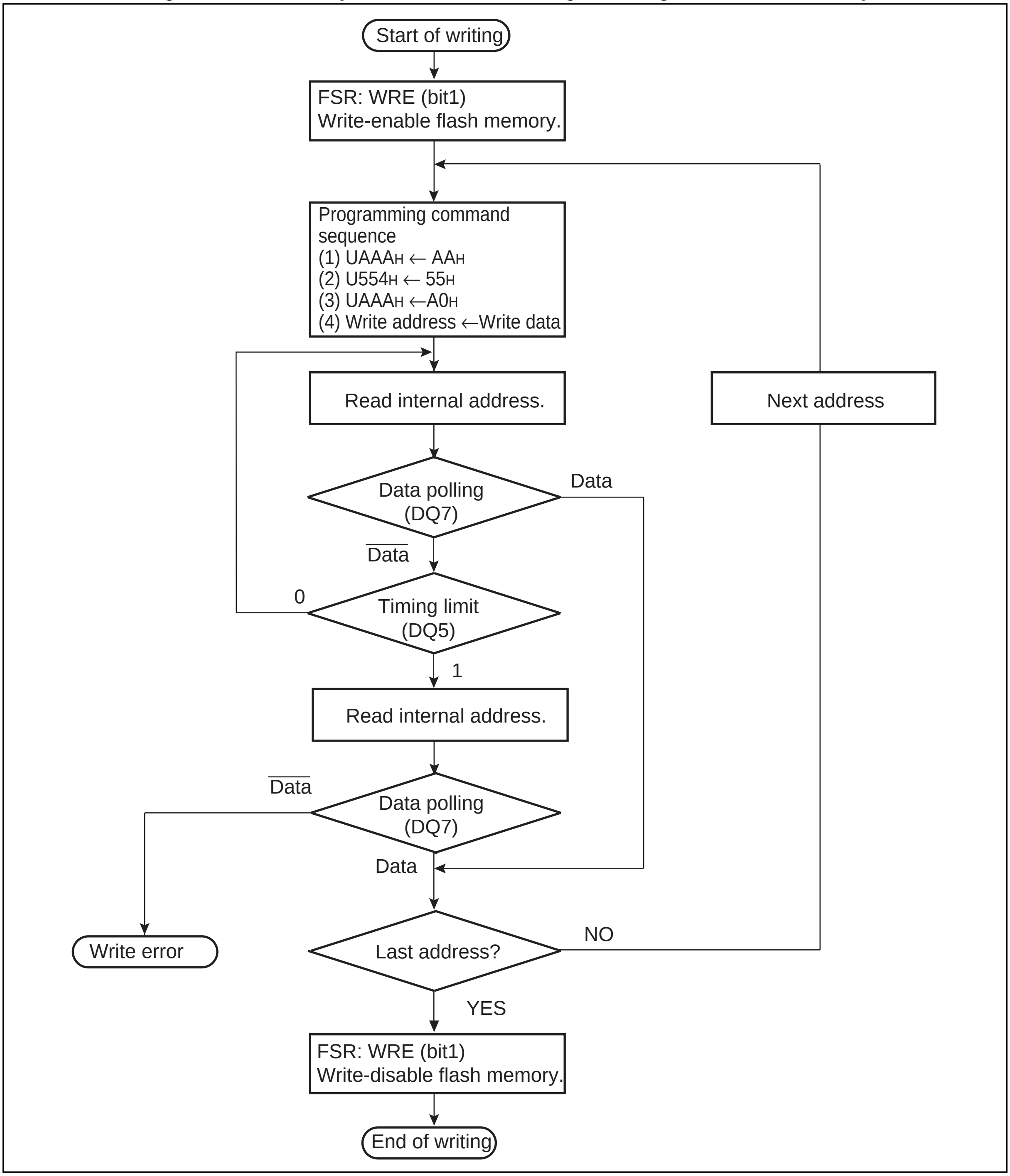
● Notes on programming data

- Bit data cannot be returned from "0" to "1" by programming. When bit data "1" is programmed to bit data "0", the data polling function (DQ7) or toggle operation (DQ6) is not terminated, the flash memory element is determined to be defective, and the execution time-out flag (DQ5) detects an error to indicate that the specified programming time has been exceeded. When data is read in the read/reset state, the bit data remains "0". To return the bit data from "0" to "1", erase flash memory.
- All commands are ignored during automatic programming.
- If a hardware reset occurs during programming, the data being programmed to the current address is not guaranteed. Retry from the chip-erase command.

■ Flash Memory Programming Procedure

- Figure 27.6-1 shows the sample procedure for programming into flash memory. The hardware sequence flags can be used to check the operating state of the automatic algorithm in flash memory. The data polling flag (DQ7) is used for checking the completion of programming into flash memory in this example.
- Flag check data should be read from the address where data was last written.
- Because the data polling flag (DQ7) and execution time-out flag (DQ5) are updated at the same time, the data polling flag (DQ7) must be checked even when the execution time-out flag (DQ5) is "1".
- Similarly, the toggle bit flag (DQ6) must be checked as it stops toggling at the same time as when the execution time-out flag (DQ5) changes to "1".

Figure 27.6-1 Sample Procedure for Programming into Flash Memory



27.6.3 Erasing All Data from Flash Memory (Chip Erase)

This section describes the procedure for issuing the chip erase command to erase all data from flash memory.

■ Erasing Data from Flash Memory (Chip Erase)

- To erase all data from flash memory, send the chip erase command in the command sequence table continuously from the CPU to flash memory.
- The chip erase command is executed in six bus operations. Chip erasing is started upon completion of the sixth programming cycle.
- Before chip erasing, the user need not perform programming into flash memory. During execution of the automatic erase algorithm, flash memory automatically programs "0" before erasing all cells automatically.

■ Notes on Chip Erase

If a hardware reset occurs during erasure, the data being erased from flash memory is not guaranteed.

MB95150/M Series

27.7 Flash Security

The flash security controller function prevents the contents of flash memory from being read through external pins.

■ Flash Security

Writing protection code "01_H" to a flash memory address (8000_H) restricts access to flash memory, barring read/write access to flash memory from any external pin. Once flash memory has been protected, the function cannot be unlocked until the chip erase command is executed.

Note that only addresses 5554_H and $2AAA_H$ can be read as exceptions.

It is advisable to code the protection code at the end of flash programming. This is to avoid unnecessary protection during programming.

Once flash memory has been protected, the chip erase operation is required before it can be reprogrammed.

CHAPTER 28
EXAMPLE OF SERIAL PROGRAMMING CONNECTION

This chapter describes the example of a serial programming connection.

Code: CM26-00124-1E

Page: 530, 531

28.1 Basic Configuration of Serial Programming Connection for Flash Memory Products

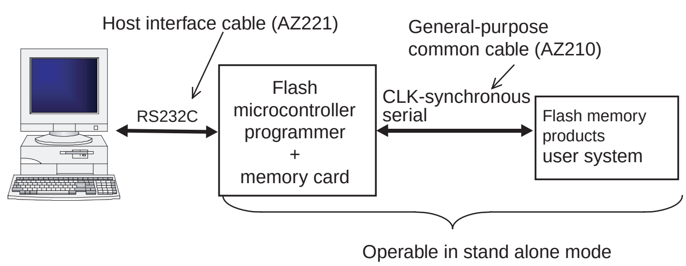
The MB95F156M/N/J supports flash ROM serial onboard programming (Fujitsu standard). This section describes the specifications.

■ Basic Configuration of Serial Programming Connection for Flash Memory Products

The AF220/AF210/AF120/AF110 flash microcontroller programmer manufactured by YokogawaDigital Computer Co., Ltd. is used for Fujitsu standard serial onboard programming.

Figure 28.1-1 shows the basic configuration of serial programming connection for flash memory products.

Figure 28.1-1 Basic Configuration of Serial Programming Connection for Flash Memory Products



Note:

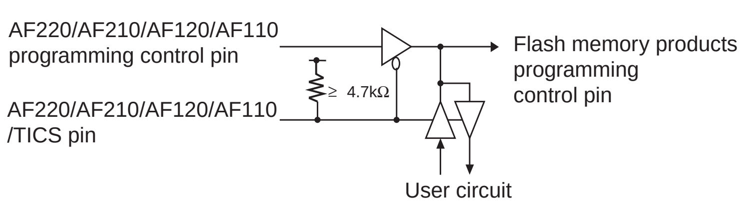
For the function and operation method of the AF220/AF210/AF120/AF110 flash microcontroller programmer and the general-purpose common cable (AZ210) and connector, contact Yokogawa Digital Computer Co., Ltd.

Table 28.1-1 Pins Used for Fujitsu Standard Serial Onboard Programming

Pin	Function	Description
MOD, P13	Mode pin	Setting MOD=High and P13=Low sets serial write mode.
X0, X1	Oscillation pins	The CPU's internal operating clock during serial write mode is the oscillator frequency divided by two. Note that a 1MHz or higher oscillator frequency must be input when performing serial writing.
RST	Reset pin	-
P10/UI0	Serial data input pin	Setting P10/UI0=Low specifies that serial write mode uses clock synchronous communications. As this low input is handled by the TTXD pin of the flash microcontroller programmer, you do not need to provide a pull-down for the P10/UI0 pin.
P11/UO0	Serial data output pin	-
P12/UCK0	Serial clock input pin	Setting P12/UCK0=High sets serial write mode. As this high input is handled by the TCK pin of the flash microcontroller programmer, you do not need to provide a pull-up for the P12/UCK0 pin.
V_{CC}	Power supply voltage supply pin	On the 3V products, the write voltage (Vcc = 2.7V to 3.6V) is supplied from the user system. On the 5V products, the write voltage (Vcc = 4.5V to 5.5V) is supplied from the user system.
V_{SS}	GND pin	Common to the GND of the flash microcontroller programmer.

As the UI0, UO0, and UCK0 pins are also used by the user system, you need to provide a control circuit as shown in Figure 28.1-2 if you want to disconnect from the user circuit during serial programming.
(The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 28.1-2 for details.)

Figure 28.1-2 Control Circuit



● Oscillation Clock Frequency and Serial Clock Input Frequency

The permitted frequency for the input serial clock on the flash memory products is calculated from the following formula. Accordingly, modify the serial clock input frequency by setting the flash microcontroller programmer, according to the oscillation clock frequency used.

Permitted frequency for the input serial clock = 0.125 × Oscillation clock frequency

Example:

Oscillation clock frequency	Maximum serial clock frequency that can be input to the microcontroller	Maximum serial clock frequency that can be set on the AF220, AF210, AF120, and AF110	Maximum serial clock frequency that can be set on the AF200
at 4MHz	500kHz	500kHz	500kHz
at 8MHz	1MHz	850kHz	500kHz
at 10MHz	1.25MHz	1.25MHz	500kHz

Table 28.1-2 System Configuration of the Flash Microcontroller Program (Yokogawa Digital Computer Co., Ltd.)

Product type		Function
Main unit	AF220/AC4P	Model with built-in Ethernet interface /100V to 220V power adapter
	AF210/AC4P	Standard model /100V to 220V power adapter
	AF120/AC4P	Single-key model with built-in Ethernet interface /100V to 220V power adapter
	AF110/AC4P	Single-key model /100V to 220V power adapter
AZ221		Writer-dedicated RS232C cable for PC/AT
AZ210		Standard target probe (a) Length: 1 m
FF201		Fujitsu control module for F2MC-16LX flash microcontroller
AZ290		Remote controller
/P2		2Mbytes PC Card (Option) Flash memory capacity: up to 128 Kbytes
/P4		4Mbytes PC Card (Option) Flash memory capacity: up to 512 Kbytes

Contact: Yokogawa Digital Computer Co., Ltd. Tell: +81-042-333-6222

Note:

Although the AF200 flash microcontroller programmer is an old model, this can be handled using the FF201 control module. The connection examples shown in the next chapter can also be used as example connections for serial writing.

28.2 Example of Serial Programming Connection

Inputting MOD="H" from TAUX3 on the AF220, AF210, AF120, or AF110 to the mode pin, which is set to MOD="L" by the user system, sets the mode to serial write mode (serial write mode: MOD="H", P12="H", P13="L").

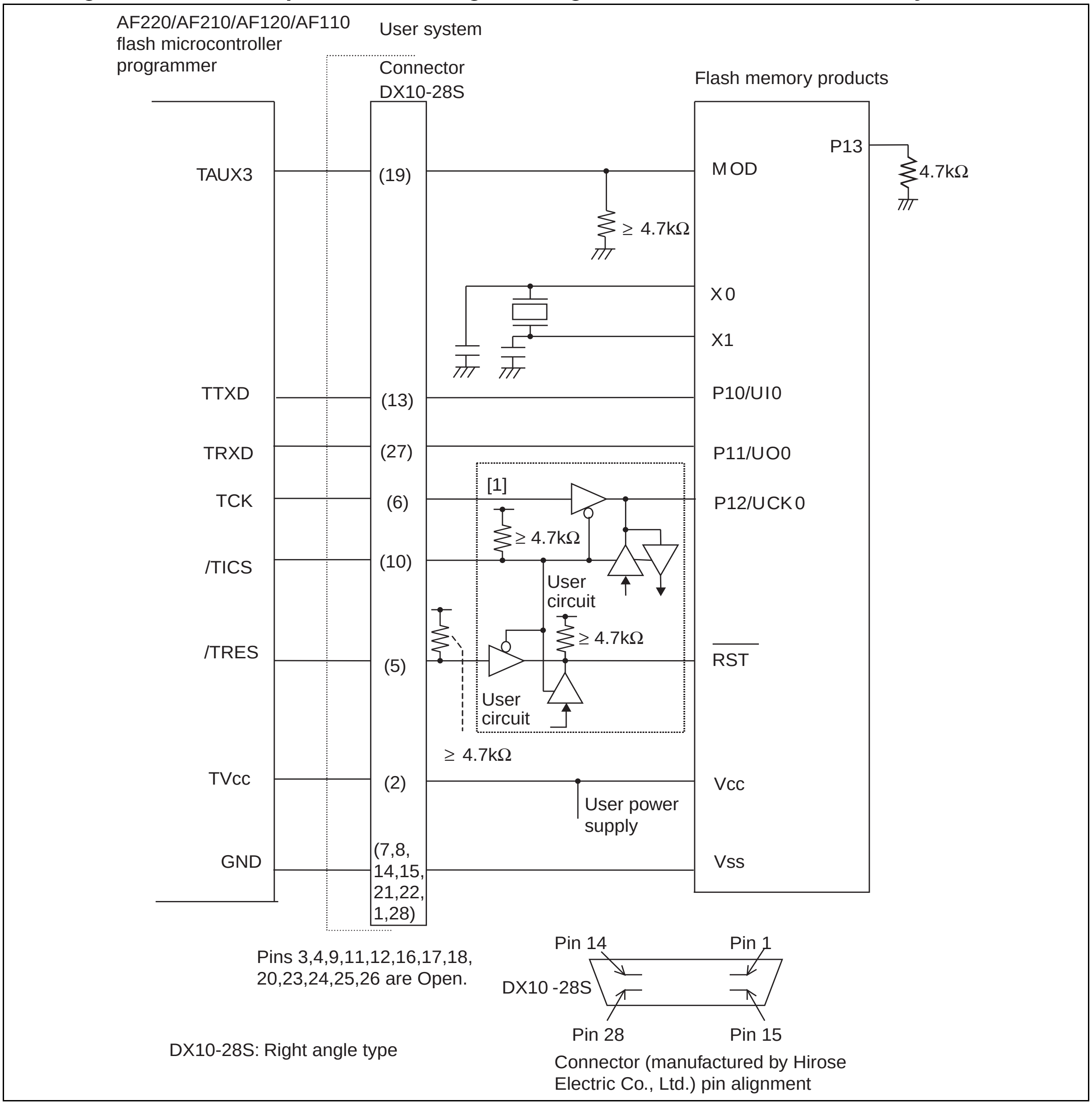
■ Example of Serial Programming Connection

Figure 28.2-1 shows an example connection for serial writing.

The TTXD pin on the flash microcontroller programmer is connected to P10/UI0 and outputs low until data transfer starts. Setting P10/UI0=Low in this way specifies that serial write mode uses clock synchronous communications.

Note that a user power supply is required for serial programming.

Figure 28.2-1 Example of Serial Programming Connection for Flash Memory Products

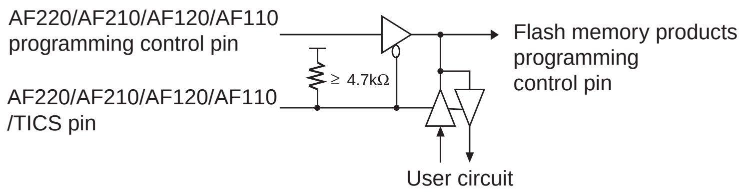


The circuit [1] shown in Figure 28.2-1 is required if you want to disconnect the UCK0 and RST pins from the user circuit during serial programming (The /TICS signal of the flash microcontroller programmer outputs low during serial writing and this disconnects the user circuit).

If it is not necessary to disconnect from the user circuit, the connection to /TICS and circuit [1] are not required. See the connection example in Figure 28.3-1.

The UI0 and UO0 pins are also used by the user system and the control circuit shown below like that used for the UCK0 pin is required if you want to disconnect from the user circuit during serial programming (The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 28.2-1 for details).

Figure 28.2-2 Control Circuit



Only connect to the AF220, AF210, AF120, or AF110 while the user power supply is turned off.

Note:

The pull-up and pull-down resistances in the above example connection are examples only and may be adjusted to suit your system.If variation in the input level to the MOD pin is possible due to noise or other factors, it is also recommended that you use a capacitor or other method to minimize noise.

28.3 Example of Minimum Connection to Flash Microcontroller Programmer

The connection between MOD and the flash microcontroller programmer is not required if the pins are set as shown in Figure 28.3-1 during serial writing (serial write mode: MOD="H", P12="H", P13="L").

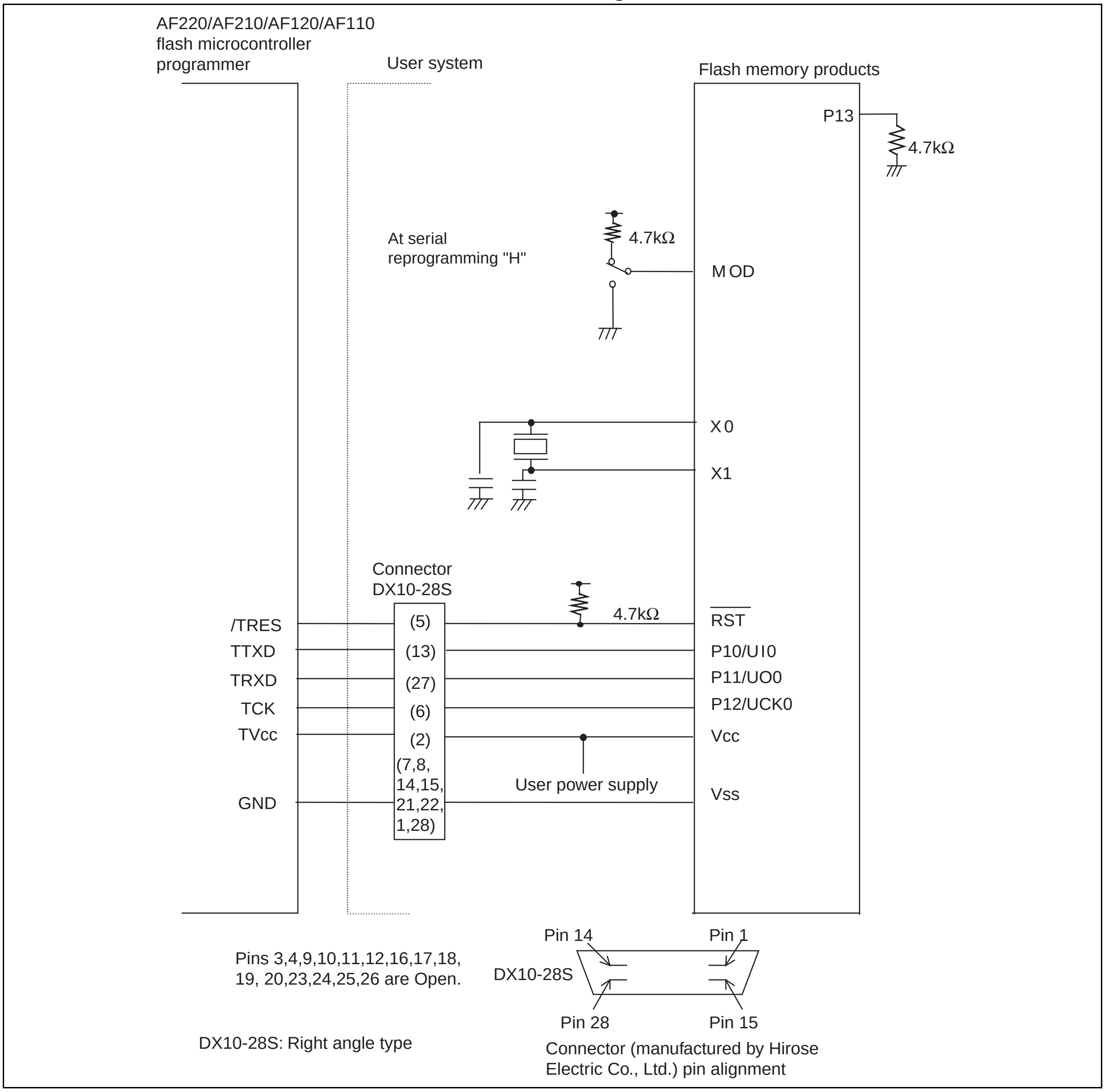
■ Example of Minimum Connection to Flash Microcontroller Programmer

Figure 28.3-1 shows an example of the minimum connection between the flash memory products and flash microcontroller programmer.

The TTXD pin on the flash microcontroller programmer is connected to P10/UI0 and outputs low until data transfer starts. Setting P10/UI0=Low in this way specifies that serial write mode uses clock synchronous communications.

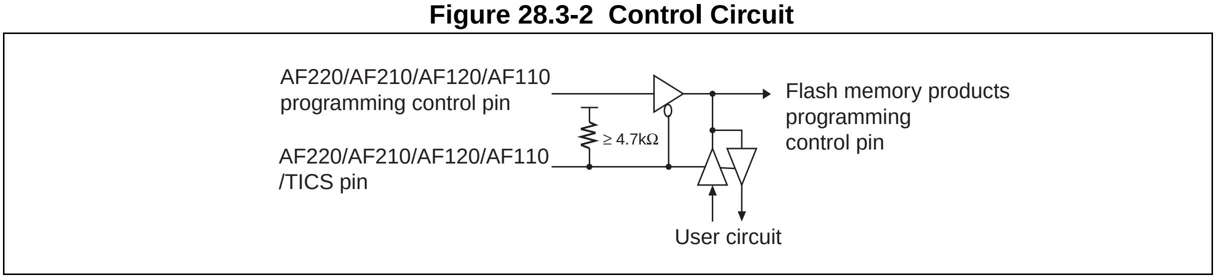
Note that a user power supply is required for serial writing.

Figure 28.3-1 Example of Minimum Connection between Flash Memory Products and Flash Microcontroller Programmer



As the UI0, UO0, and UCK0 pins are also used by the user system, you need to provide a control circuit as shown below if you want to disconnect from the user circuit during serial writing (The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 28.2-1 for details).

Figure 28.3-2 Control Circuit



Only connect to the AF220, AF210, AF120, or AF110 while the user power supply is turned off.

Note:

The pull-up and pull-down resistances in the above example connection are examples only and may be adjusted to suit your system. If variation in the input level to the MOD pin is possible due to noise or other factors, it is also recommended that you use a capacitor or other method to minimize noise.

APPENDIX

This appendix explains I/O map, interrupt list, memory map, pin status, instruction overview, mask option and writing to Flash microcontroller using parallel writer.

APPENDIX A I/O Map

APPENDIX A I/O Map

This section explains I/O map that is used on MB95150/M series.

■ I/O Map

Table A-1 MB95150/M Series (1 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0000_H	PDR0	Port 0 data register	R/W	00000000_B
0001_H	DDR0	Port 0 direction register	R/W	00000000_B
0002_H	PDR1	Port 1 data register	R/W	00000000_B
0003_H	DDR1	Port 1 direction register	R/W	00000000_B
0004_H	(Disabled)			
0005_H	WATR	Oscillation stabilization wait time setting register	R/W	11111111_B
0006_H	PLLC	PLL control register	R/W	00000000_B
0007_H	SYCC	System clock control register	R/W	$1010X011_B$
0008_H	STBC	Standby control register	R/W	00000000_B
0009_H	RSRR	Reset source register	R/W	$XXXXXXXX_B$
$000A_H$	TBTC	Time-base timer control register	R/W	00000000_B
$000B_H$	WPCR	Watch prescaler control register	R/W	00000000_B
$000C_H$	WDTC	Watchdog timer control register	R/W	00000000_B
$000D_H$ to 0015_H	(Disabled)			
0016_H	PDR6	Port 6 data register	R/W	00000000_B
0017_H	DDR6	Port 6 direction register	R/W	00000000_B
0018_H to $001B_H$	(Disabled)			
$001C_H$	PDR9	Port 9 data register	R/W	00000000_B
$001D_H$	DDR9	Port 9 direction register	R/W	00000000_B
$001E_H$	PDRA	Port A data register	R/W	00000000_B
$001F_H$	DDRA	Port A direction register	R/W	00000000_B
0020_H	PDRB	Port B data register	R/W	00000000_B
0021_H	DDRB	Port B direction register	R/W	00000000_B
0022_H to 0029_H	(Disabled)			
$002A_H$	PDRG	Port G data register	R/W	00000000_B
$002B_H$	DDRG	Port G direction register	R/W	00000000_B
$002C_H$	PUL0	Port 0 pull-up register	R/W	00000000_B

APPENDIX A I/O Map

MB95150/M Series

Table A-1 MB95150/M Series (2 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
$002D_H$	PUL1	Port 1 pull-up register	R/W	00000000_B
$002E_H$ to 0034_H	(Disabled)			
0035_H	PULG	Port G pull-up register	R/W	00000000_B
0036_H	T01CR1	8/16-bit compound timer 01 control status register 1 ch.0	R/W	00000000_B
0037_H	T00CR1	8/16-bit compound timer 00 control status register 1 ch.0	R/W	00000000_B
0038_H	T11CR1	8/16-bit compound timer 11 control status register 1 ch.1	R/W	00000000_B
0039_H	T10CR1	8/16-bit compound timer 10 control status register 1 ch.1	R/W	00000000_B
$003A_H$	PC01	8/16-bit PPG1 control register ch.0	R/W	00000000_B
$003B_H$	PC00	8/16-bit PPG0 control register ch.0	R/W	00000000_B
$003C_H$	PC11	8/16-bit PPG1 control register ch.1	R/W	00000000_B
$003D_H$	PC10	8/16-bit PPG0 control register ch.1	R/W	00000000_B
$003E_H$ to 0041_H	(Disabled)			
0042_H	PCNTH0	16-bit PPG status control register upper ch.0	R/W	00000000_B
0043_H	PCNTL0	16-bit PPG status control register lower ch.0	R/W	00000000_B
0044_H to 0047_H	(Disabled)			
0048_H	EIC00	External interrupt circuit control register ch.0, ch.1	R/W	00000000_B
0049_H	EIC10	External interrupt circuit control register ch.2, ch.3	R/W	00000000_B
$004A_H$	EIC20	External interrupt circuit control register ch.4, ch.5	R/W	00000000_B
$004B_H$	EIC30	External interrupt circuit control register ch.6, ch.7	R/W	00000000_B
$004C_H$ to $004F_H$	(Disabled)			
0050_H	SCR	LIN-UART serial control register	R/W	00000000_B
0051_H	SMR	LIN-UART serial mode register	R/W	00000000_B
0052_H	SSR	LIN-UART serial status register	R/W	00001000_B
0053_H	RDR/TDR	LIN-UART reception/transmission data register	R/W	00000000_B
0054_H	ESCR	LIN-UART extended status control register	R/W	00000100_B
0055_H	ECCR	LIN-UART extended communication control register	R/W	$000000XX_B$
0056_H	SMC10	UART/SIO serial mode control register 1 ch.0	R/W	00000000_B
0057_H	SMC20	UART/SIO serial mode control register 2 ch.0	R/W	00100000_B
0058_H	SSR0	UART/SIO serial status register ch.0	R/W	00000001_B
0059_H	TDR0	UART/SIO serial output data register ch.0	R/W	00000000_B
$005A_H$	RDR0	UART/SIO serial input data register ch.0	R	00000000_B

APPENDIX A I/O Map

Table A-1 MB95150/M Series (3 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
$005B_H$ to $006B_H$	(Disabled)			
$006C_H$	ADC1	8/10 bit A/D control register 1	R/W	00000000_B
$006D_H$	ADC2	8/10 bit A/D control register 2	R/W	00000000_B
$006E_H$	ADDH	8/10 bit A/D data register upper	R/W	00000000_B
$006F_H$	ADDL	8/10 bit A/D data register lower	R/W	00000000_B
0070_H	WCSR	Watch counter control register	R/W	00000000_B
0071_H	(Disabled)			
0072_H	FSR	Flash memory status register	R/W	$000X0000_B$
0073_H to 0075_H	(Disabled)			
0076_H	WREN	Wild register address compare enable register	R/W	00000000_B
0077_H	WROR	Wild register data test setting register	R/W	00000000_B
0078_H	-	Register bank pointer (RP), Mirror of direct bank pointer (DP)	-	-
0079_H	ILR0	Interrupt level setting register 0	R/W	11111111_B
$007A_H$	ILR1	Interrupt level setting register 1	R/W	11111111_B
$007B_H$	ILR2	Interrupt level setting register 2	R/W	11111111_B
$007C_H$	ILR3	Interrupt level setting register 3	R/W	11111111_B
$007D_H$	ILR4	Interrupt level setting register 4	R/W	11111111_B
$007E_H$	ILR5	Interrupt level setting register 5	R/W	11111111_B
$007F_H$	(Disabled)			
$0F80_H$	WRARH0	Wild register address setup register upper ch.0	R/W	00000000_B
$0F81_H$	WRARL0	Wild register address setup register lower ch.0	R/W	00000000_B
$0F82_H$	WRDR0	Wild register data setup register ch.0	R/W	00000000_B
$0F83_H$	WRARH1	Wild register address setup register upper ch.1	R/W	00000000_B
$0F84_H$	WRARL1	Wild register address setup register lower ch.1	R/W	00000000_B
$0F85_H$	WRDR1	Wild register data setup register ch.1	R/W	00000000_B
$0F86_H$	WRARH2	Wild register address setup register upper ch.2	R/W	00000000_B
$0F87_H$	WRARL2	Wild register address setup register lower ch.2	R/W	00000000_B
$0F88_H$	WRDR2	Wild register data setup register ch.2	R/W	00000000_B
$0F89_H$ to $0F91_H$	(Disabled)			
$0F92_H$	T01CR0	8/16-bit compound timer 01 control status register 0 ch.0	R/W	00000000_B
$0F93_H$	T00CR0	8/16-bit compound timer 00 control status register 0 ch.0	R/W	00000000_B
$0F94_H$	T01DR	8/16-bit compound timer 01 data register ch.0	R/W	00000000_B
$0F95_H$	T00DR	8/16-bit compound timer 00 data register ch.0	R/W	00000000_B
$0F96_H$	TMCR0	8/16-bit compound timer 00/01 timer mode control register ch.0	R/W	00000000_B

APPENDIX A I/O Map

MB95150/M Series

Table A-1 MB95150/M Series (4 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
$0F97_H$	T11CR0	8/16-bit compound timer 11 control status register 0 ch.1	R/W	00000000_B
$0F98_H$	T10CR0	8/16-bit compound timer 10 control status register 0 ch.1	R/W	00000000_B
$0F99_H$	T11DR	8/16-bit compound timer 11 data register ch.1	R/W	00000000_B
$0F9A_H$	T10DR	8/16-bit compound timer 10 data register ch.1	R/W	00000000_B
$0F9B_H$	TMCR1	8/16-bit compound timer 10/11 timer mode control register ch.1	R/W	00000000_B
$0F9C_H$	PPS01	8/16-bit PPG01 cycle setting buffer register ch.1	R/W	11111111_B
$0F9D_H$	PPS00	8/16-bit PPG00 cycle setting buffer register ch.0	R/W	11111111_B
$0F9E_H$	PDS01	8/16-bit PPG01 duty setting buffer register ch.1	R/W	11111111_B
$0F9F_H$	PDS00	8/16-bit PPG00 duty setting buffer register ch.0	R/W	11111111_B
$0FA0_H$	PPS11	8/16-bit PPG01 cycle setting buffer register ch.1	R/W	11111111_B
$0FA1_H$	PPS10	8/16-bit PPG00 cycle setting buffer register ch.0	R/W	11111111_B
$0FA2_H$	PDS11	8/16-bit PPG01 duty setting buffer register ch.1	R/W	11111111_B
$0FA3_H$	PDS10	8/16-bit PPG00 duty setting buffer register ch.0	R/W	11111111_B
$0FA4_H$	PPGS	8/16-bit PPG startup register	R/W	00000000_B
$0FA5_H$	REVC	8/16-bit PPG output reverse register	R/W	00000000_B
$0FA6_H$ to $0FA9_H$	(Disabled)			
$0FAA_H$	PDCRH0	16-bit PPG down counter register upper ch.0	R	00000000_B
$0FAB_H$	PDCRL0	16-bit PPG down counter register lower ch.0	R	00000000_B
$0FAC_H$	PCSRH0	16-bit PPG cycle setting buffer register upper ch.0	R/W	11111111_B
$0FAD_H$	PCSRL0	16-bit PPG cycle setting buffer register lower ch.0	R/W	11111111_B
$0FAE_H$	PDUTH0	16-bit PPG duty setting buffer register upper ch.0	R/W	11111111_B
$0FAF_H$	PDUTL0	16-bit PPG duty setting buffer register lower ch.0	R/W	11111111_B
$0FB0_H$ to $0FBB_H$	(Disabled)			
$0FBC_H$	BGR1	LIN-UART baud rate generator register 1	R/W	00000000_B
$0FBD_H$	BGR0	LIN-UART baud rate generator register 0	R/W	00000000_B
$0FBE_H$	PSSR0	UART/SIO dedicated baud rate generator prescaler select register	R/W	00000000_B
$0FBF_H$	BRSR0	UART/SIO dedicated baud rate generator baud rate setting register	R/W	00000000_B
$0FC0_H$ to $0FC2_H$	(Disabled)			
$0FC3_H$	AIDRL	A/D input disable register lower	R/W	00000000_B
$0FC4_H$	LCDCC	LCDC control register	R/W	00010000
$0FC5_H$	LCDCE1	LCDC enable register 1	R/W	00110000
$0FC6_H$	LCDCE2	LCDC enable register 2	R/W	00000000_B
$0FC7_H$	LCDCE3	LCDC enable register 3	R/W	00000000_B

APPENDIX A I/O Map

Table A-1 MB95150/M Series (5 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
$0FC8_H$ to $0FCA_H$	(Disabled)			
$0FCB_H$	LCDCB1	LCDC blinking set register 1	R/W	00000000_B
$0FCC_H$	LCDCB2	LCDC blinking set register 2	R/W	00000000_B
$0FCD_H$ to $0FD4_H$	LCDRAM	LCDC display RAM	R/W	00000000_B
$0FD5_H$ to $0FE2_H$	(Disabled)			
$0FE3_H$	WCDR	Watch counter data register	R/W	00111111_B
$0FE4_H$ to $0FE6_H$	(Disabled)			
$0FE7_H$	ILSR2	Input level selection register 2	R/W	00000000_B
$0FE8_H$, $0FE9_H$	(Disabled)			
$0FEA_H$	CSVCR	Clock supervisor control register	R/W	00011100_B
$0FEB_H$ to $0FED_H$	(Disabled)			
$0FEE_H$	ILSR	Input level selection register	R/W	00000000_B
$0FEF_H$	WICR	Interrupt pin selection circuit control register	R/W	01000000_B
$0FF0_H$ to $0FFF_H$	(Disabled)			

- R/W access symbols
  R/W : Readable/Writable
  R : Read only
  W : Write only
- Initial value symbols
  0 : The initial value of this bit is "0".
  1 : The initial value of this bit is "1".
  X : The initial value of this bit is undefined.

APPENDIX B Table of Interrupt Causes

APPENDIX B Table of Interrupt Causes

This section describes the table of interrupt causes used in MB95150/M series.

■ Table of Interrupt Causes

Refer to "CHAPTER 8 INTERRUPTS" for interrupt operation.

Table B-1 MB95150/M Series

Interrupt causes	Interrupt request number	Address of vector table		Bit name of interrupt level setting register	The same level priority (Concurrence)
		Upper	Lower		
(reset vector)	-	$FFFE_H$	$FFFF_H$	-	High
(mode data)	-	-	$FFFD_H$	-	↑
External interrupt ch.0	IRQ0	$FFFA_H$	$FFFB_H$	L00 [1:0]	
External interrupt ch.4					
External interrupt ch.1	IRQ1	$FFF8_H$	$FFF9_H$	L01 [1:0]	
External interrupt ch.5					
External interrupt ch.2	IRQ2	$FFF6_H$	$FFF7_H$	L02 [1:0]	
External interrupt ch.6					
External interrupt ch.3	IRQ3	$FFF4_H$	$FFF5_H$	L03 [1:0]	
External interrupt ch.7					
UART/SIO ch.0	IRQ4	$FFF2_H$	$FFF3_H$	L04 [1:0]	
8/16-bit compound timer ch.0 (lower)	IRQ5	$FFF0_H$	$FFF1_H$	L05 [1:0]	
8/16-bit compound timer ch.0 (upper)	IRQ6	$FFEE_H$	$FFEF_H$	L06 [1:0]	
LIN-UART (reception)	IRQ7	$FFEC_H$	$FFED_H$	L07 [1:0]	
LIN-UART (transmission)	IRQ8	$FFEA_H$	$FFEB_H$	L08 [1:0]	
8/16-bit PPG ch.1 (lower)	IRQ9	$FFE8_H$	$FFE9_H$	L09 [1:0]	
8/16-bit PPG ch.1 (upper)	IRQ10	$FFE6_H$	$FFE7_H$	L10 [1:0]	
(Not used)	IRQ11	$FFE4_H$	$FFE5_H$	L11 [1:0]	
8/16-bit PPG ch.0 (lower)	IRQ12	$FFE2_H$	$FFE3_H$	L12 [1:0]	
8/16-bit PPG ch.0 (upper)	IRQ13	$FFE0_H$	$FFE1_H$	L13 [1:0]	
8/16-bit compound timer ch.1 (upper)	IRQ14	$FFDE_H$	$FFDF_H$	L14 [1:0]	
16-bit PPG ch.0	IRQ15	$FFDC_H$	$FFDD_H$	L15 [1:0]	
(Not used)	IRQ16	$FFDA_H$	$FFDB_H$	L16 [1:0]	
(Not used)	IRQ17	$FFD8_H$	$FFD9_H$	L17 [1:0]	
8/10-bit A/D	IRQ18	$FFD6_H$	$FFD7_H$	L18 [1:0]	
Time-base timer	IRQ19	$FFD4_H$	$FFD5_H$	L19 [1:0]	
Watch prescaler/counter	IRQ20	$FFD2_H$	$FFD3_H$	L20 [1:0]	
(Not used)	IRQ21	$FFD0_H$	$FFD1_H$	L21 [1:0]	↓
8/16-bit compound timer ch.1 (lower)	IRQ22	$FFCE_H$	$FFCF_H$	L22 [1:0]	
Flash memory	IRQ23	$FFCC_H$	$FFCD_H$	L23 [1:0]	Low

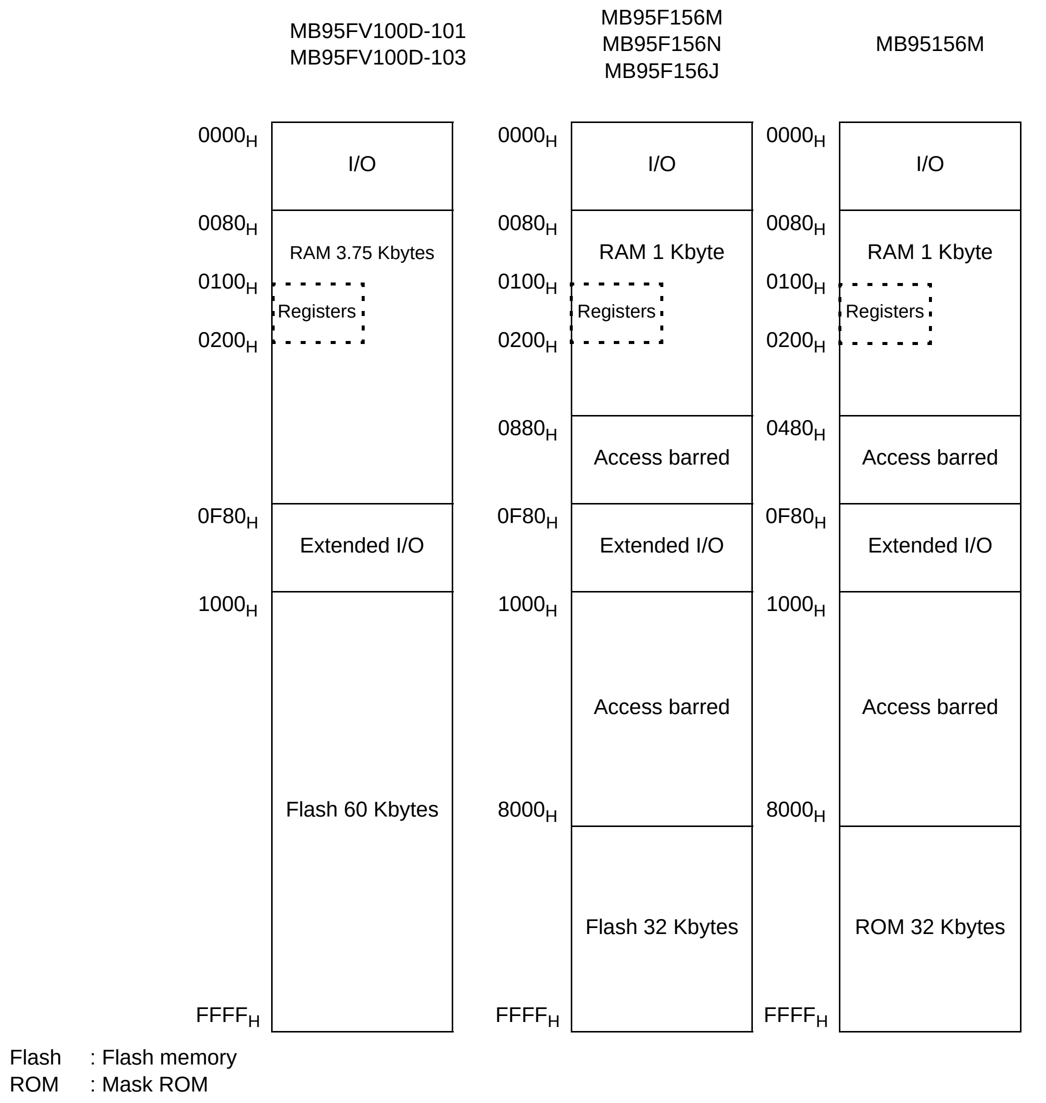
APPENDIX C Memory Map

APPENDIX C   Memory Map

This section shows the memory map of MB95150/M series.

■ Memory Map

Figure C-1 Memory Map



APPENDIX D Pin Status of MB95150/M series

APPENDIX D Pin Status of MB95150/M series

The state of the terminal of the MB95150/M series in each mode is shown in Table D-1.

■ Pin Status in Each Mode

Table D-1 Pin Status in Each Mode (1 / 3)

<table>
<tr><th rowspan="2">Pin name</th><th rowspan="2">Normal operation</th><th rowspan="2">Sleep mode</th><th colspan="2">Stop mode</th><th colspan="2">Watch mode</th><th rowspan="2">While resetting</th></tr>
<tr><th>SPL=0</th><th>SPL=1</th><th>SPL=0</th><th>SPL=1</th></tr>
<tr><td>X0</td><td>Oscillation circuit input</td><td>Oscillation circuit input</td><td>Hi-Z</td><td>Hi-Z</td><td>Hi-Z</td><td>Hi-Z</td><td>Oscillation circuit input</td></tr>
<tr><td>X1</td><td>Oscillation circuit output</td><td>Oscillation circuit input</td><td>"H"</td><td>"H"</td><td>"H"</td><td>"H"</td><td>Oscillation circuit output</td></tr>
<tr><td>MOD</td><td>Mode input</td><td>Mode input</td><td>Mode input</td><td>Mode input</td><td>Mode input</td><td>Mode input</td><td>Mode input</td></tr>
<tr><td>$\overline{RST}$</td><td>Reset input</td><td>Reset input</td><td>Reset input</td><td>Reset input</td><td>Reset input</td><td>Reset input</td><td>Reset input</td></tr>
<tr><td>P00/INT00/ AN00</td><td rowspan="8">I/O port/ peripheral function I/O/ analog input</td><td rowspan="8">I/O port/ peripheral function I/O/ analog input</td><td rowspan="8">I/O port/ peripheral function I/O/ analog input</td><td rowspan="8">Hi-Z (However, the setting of the pull-up is effective.) Input interception (However, an external interrupt can be input when the external interrupt is enable.)</td><td rowspan="8">I/O port/ peripheral function I/O/ analog input</td><td rowspan="8">Hi-Z Input interception (However, an external interrupt can be input when the external interrupt is enable.)</td><td rowspan="8">Hi-Z Input disable[*2]</td></tr>
<tr><td>P01/INT01/ AN01</td></tr>
<tr><td>P02/INT02/ AN02</td></tr>
<tr><td>P03/INT03/ AN03</td></tr>
<tr><td>P04/INT04/ AN04</td></tr>
<tr><td>P05/INT05/ AN05</td></tr>
<tr><td>P06/INT06/ AN06</td></tr>
<tr><td>P07/INT07/ AN07</td></tr>
<tr><td>P10/UI0</td><td rowspan="5">I/O port/ peripheral function I/O</td><td rowspan="5">I/O port/ peripheral function I/O</td><td rowspan="5">I/O port/ peripheral function I/O</td><td rowspan="5">Hi-Z (However, the setting of the pull-up is effective.) Input interception</td><td rowspan="5">I/O port/ peripheral function I/O</td><td rowspan="5">Hi-Z (However, the setting of the pull-up is effective.) Input interception</td><td rowspan="5">Hi-Z Input enable[*1] (However, it doesn't function.)</td></tr>
<tr><td>P11/UO0</td></tr>
<tr><td>P12/UCK0</td></tr>
<tr><td>P13/TRG0/ ADTG</td></tr>
<tr><td>P14/PPG0</td></tr>
</table>

APPENDIX D Pin Status of MB95150/M series

Table D-1 Pin Status in Each Mode (2 / 3)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		While resetting
			SPL=0	SPL=1	SPL=0	SPL=1	
P60/PPG10/SEG08	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z Input interception	I/O port/ peripheral function I/O	Hi-Z Input interception	Hi-Z Input disable*2
P61/PPG11/SEG09							
P62/TO10/SEG10							
P63/TO11/SEG11							
P64/EC1/SEG12							
P65/SCK/SEG13							
P66/SOT/SEG14							
P67/SIN/SEG15							
P90/V3	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z Input interception	I/O port/ peripheral function I/O	Hi-Z Input interception	Hi-Z Input enable*1 (P95 and P94 input is enable. However, it doesn't function.)
P91/V2							
P92/V1							
P93/V0							
P94							
P95							
PA0/COM0	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z Input interception	I/O port/ peripheral function I/O	Hi-Z Input interception	Hi-Z Input disable*2
PA1/COM1							
PA2/COM2							
PA3/COM3							

APPENDIX D Pin Status of MB95150/M series

Table D-1 Pin Status in Each Mode (3 / 3)

<table>
<tr><th rowspan="2">Pin name</th><th rowspan="2">Normal operation</th><th rowspan="2">Sleep mode</th><th colspan="2">Stop mode</th><th colspan="2">Watch mode</th><th rowspan="2">While resetting</th></tr>
<tr><th>SPL=0</th><th>SPL=1</th><th>SPL=0</th><th>SPL=1</th></tr>
<tr><td>PB0/SEG00</td><td rowspan="8">I/O port/ peripheral function I/O</td><td rowspan="8">I/O port/ peripheral function I/O</td><td rowspan="8">I/O port/ peripheral function I/O</td><td rowspan="8">Hi-Z
Input interception</td><td rowspan="8">I/O port/ peripheral function I/O</td><td rowspan="8">Hi-Z
Input interception</td><td rowspan="8">Hi-Z
Input enable*1 (However, it does not function.)</td></tr>
<tr><td>PB1/SEG01</td></tr>
<tr><td>PB2/SEG02</td></tr>
<tr><td>PB3/ SEG03/ PPG00</td></tr>
<tr><td>PB4/ SEG04/ PPG01</td></tr>
<tr><td>PB5/ SEG05/ TO00</td></tr>
<tr><td>PB6/ SEG06/ TO01</td></tr>
<tr><td>PB7/ SEG07/ EC0</td></tr>
<tr><td>PG0/C *3</td><td>I/O port</td><td>I/O port</td><td>I/O port</td><td>Hi-Z
Input interception</td><td>I/O port</td><td>Hi-Z
Input interception</td><td>Hi-Z
Input enable*1 (However, it does not function.)</td></tr>
</table>

SPL: Pin status specification bit of standby control register (STBC: SPL)

Hi-Z: High impedance

*1: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pullup or output pin is recommended.

*2: "Input disable" means direct input gate operation from the terminal is disable status.

*3: 5V product uses C pin.

APPENDIX E Instruction Overview

APPENDIX E Instruction Overview

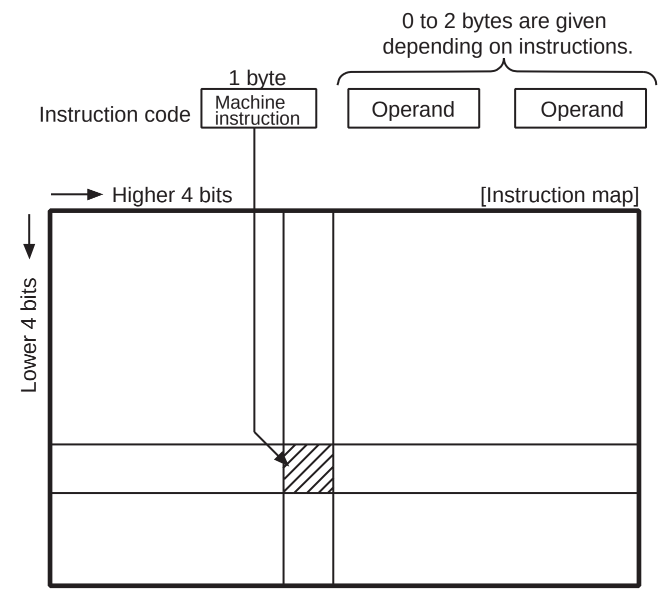
This section explains the instructions used in F^{2}MC-8FX.

■ Instruction Overview of F^{2}MC-8FX

In F^{2}MC-8FX, there are 140 kinds of one byte machine instructions (as the map, 256 bytes), and the instruction code is composed of the instruction and the operand following it.

Figure E-1 shows the correspondence of the instruction code and the instruction map.

Figure E-1 Instruction Code and Instruction Map



- The instruction is classified into following four types; forwarding system, operation system, branch system and others.
- There are various methods of addressing, and ten kinds of addressing can be selected by the selection and the operand specification of the instruction.
- This provides with the bit operation instruction, and can operate the read modification write.
- There is an instruction that directs special operation.

APPENDIX E Instruction Overview

■ Explanation of Display Sign of Instruction

Table E-1 shows the explanation of the sign used by explaining the instruction code of this APPENDIX E.

Table E-1 Explanation of Sign in Instruction Table

Sign	Signification
dir	Direct address (8-bit length)
off	Offset (8-bit length)
ext	Extended address (16-bit length)
#vct	Vector table number (3-bit length)
#d8	Immediate data (8-bit length)
#d16	Immediate data (16-bit length)
dir:b	Bit direct address (8-bit length: 3-bit length)
rel	Branch relative address (8-bit length)
@	Register indirect (Example: @A, @IX, @EP)
A	Accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
AH	Upper 8-bit of accumulator (8-bit length)
AL	Lower 8-bit of accumulator (8-bit length)
T	Temporary accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
TH	Upper 8-bit of temporary accumulator (8-bit length)
TL	Lower 8-bit of temporary accumulator (8-bit length)
IX	Index register (16-bit length)
EP	Extra pointer (16-bit length)
PC	Program counter (16-bit length)
SP	Stack pointer (16-bit length)
PS	Program status (16-bit length)
dr	Either of accumulator or index register (16-bit length)
CCR	Condition code register (8-bit length)
RP	Register bank pointer (5-bit length)
DP	Direct bank pointer (3-bit length)
Ri	General-purpose register (8-bit length, i = 0 to 7)
x	This shows that x is immediate data. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
(x)	This shows that contents of x are objects of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
((x))	This shows that the address that contents of x show is an object of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)

APPENDIX E Instruction Overview

■ Explanation of Item in Instruction Table

Table E-2 Explanation of Item in Instruction Table

Item	Description
MNEMONIC	It shows the assembly description of the instruction.
~	It shows the number of cycles of the instruction. One instruction cycle is a machine cycle. Note: The number of cycles of the instruction can be delayed by 1 cycle by the immediately preceding instruction. Moreover, the number of cycles of the instruction might be extended in the access to the I/O area.
#	It shows the number of bytes for the instruction.
Operation	It shows the operations for the instruction.
TL, TH, AH	They show the change (auto forwarding from A to T) in the content when each TL, TH, and AH instruction is executed. The sign in the column indicates the followings respectively. • -: No change • dH: upper 8 bits of the data described in operation. • AL and AH: the contents become those of the immediately preceding instruction's AL and AH. • 00: Become 00
N, Z, V, C	They show the instruction into which the corresponding flag is changed respectively. The sign in the column shows the followings respectively. • -: No change • +: Change • R: Become "0" • S: Become "1"
OP CODE	It shows the code of the instruction. When a pertinent instruction occupies two or more codes, it follows the following description rules. [Example] 48 to 4F: This shows 48, 49....4F.

APPENDIX E Instruction Overview

E.1 Addressing

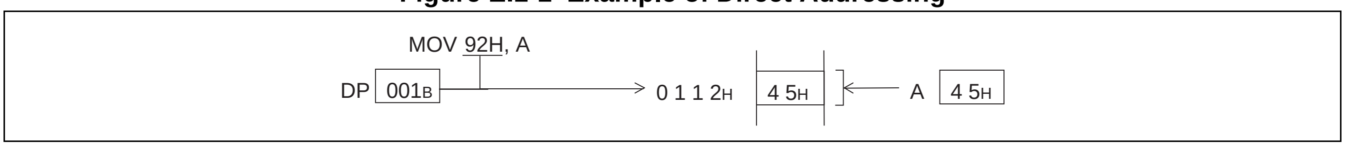
F²MC-8FX has the following ten types of addressings:
- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

■ Explanation of Addressing

● Direct addressing

This is used when accessing the direct area of "0000_H" to "$047F_H$" with addressing indicated "dir" in instruction table. In this addressing, when the operand address is "00_H" to "$7F_H$", it is accessed into "0000_H" to "$007F_H$". Moreover, when the operand address is "80_H" to "FF_H", the access can be mapped in "0080_H" to "$047F_H$" by setting of direct bank pointer DP. Figure E.1-1 shows an example.

Figure E.1-1 Example of Direct Addressing

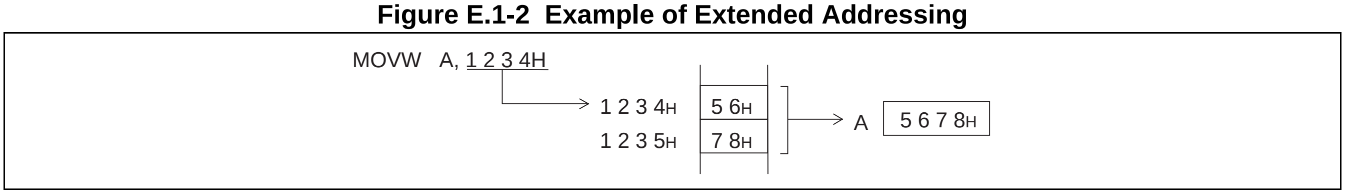


● Extended addressing

This is used when the area of the entire 64 Kbytes is accessed by addressing shown "ext" in the instruction table. In this addressing, the first operand specifies one high rank byte of the address and the second operand specifies one subordinate position byte of the address.

Figure E.1-2 shows an example.

Figure E.1-2 Example of Extended Addressing



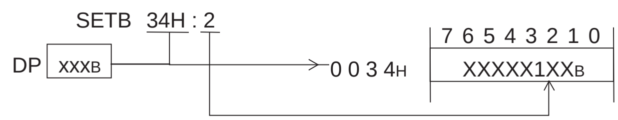
APPENDIX E Instruction Overview

● Bit direct addressing

This is used when accessing the direct area of "0000_H" to "$047F_H$" in bit unit with addressing indicated "dir:b" in instruction table. In this addressing, when the operand address is "00_H" to "$7F_H$", it is accessed into "0000_H" to "$007F_H$". Moreover, when the operand address is "80_H" to "FF_H", the access can be mapped in "0080_H" to "$047F_H$" by setting of direct bank pointer DP. The position of the bit in the specified address is specified by the values of the instruction code of three subordinate position bits.

Figure E.1-3 shows an example.

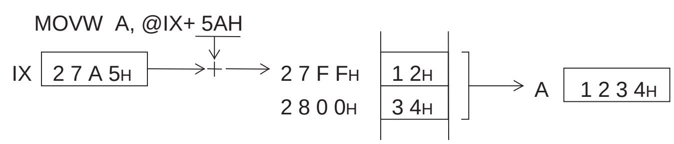
Figure E.1-3 Example of Bit Direct Addressing



● Index addressing

This is used when the area of the entire 64 Kbytes is accessed by addressing shown "@IX+off" in the instruction table. In this addressing, the content of the first operand is sign extended and added to IX (index register) to the resulting address. Figure E.1-4 shows an example.

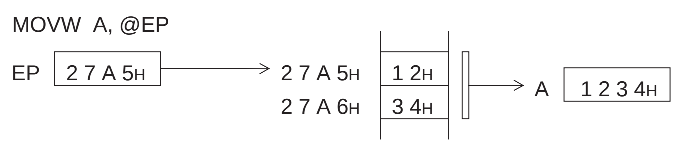
Figure E.1-4 Example of Index Addressing



● Pointer addressing

This is used when the area of the entire 64 Kbytes is accessed by addressing shown "@EP" in the instruction table. In this addressing, the content of EP (extra pointer) is assumed to be an address. Figure E.1-5 shows an example.

Figure E.1-5 Example of Pointer Addressing



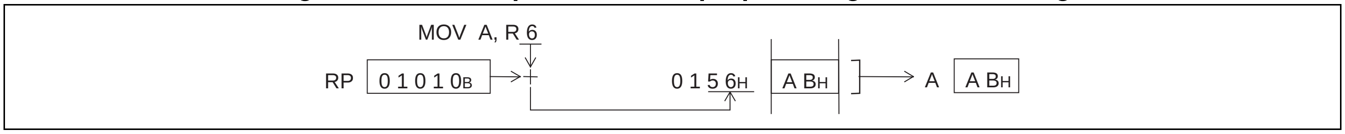
● General-purpose register addressing

This is used when accessing the register bank in general-purpose register area with the addressing shown "Ri" in instruction table. In this addressing, fix one high rank byte of the address to "01" and create one subordinate position byte from the contents of RP (register bank pointer) and three subordinate bits of the operation code to access to this address. Figure E.1-6 shows an example.

APPENDIX E Instruction Overview

MB95150/M Series

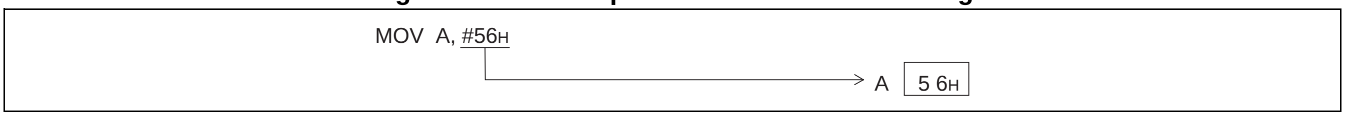
Figure E.1-6 Example of General-purpose Register Addressing



● Immediate addressing

This is used when immediate data is needed in addressing shown "#d8" in the instruction table. In this addressing, the operand becomes immediate data as it is. The specification of byte/word depends on the operation code. Figure E.1-7 shows an example.

Figure E.1-7 Example of Immediate Addressing



● Vector addressing

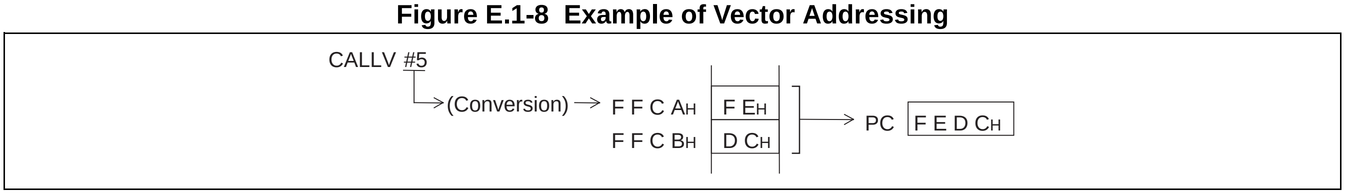
This is used when branching to the subroutine address registered in the table with the addressing shown "#vct" in the instruction table. In this addressing, information on "#vct" is contained in the operation code, and the address of the table is created using the combinations shown in Table E.1-1.

Table E.1-1 Vector Table Address Corresponding to "#vct"

#vct	Vector table address (jump destination high-ranking address: subordinate address)
0	FFC0$_H$: FFC1$_H$
1	FFC2$_H$: FFC3$_H$
2	FFC4$_H$: FFC5$_H$
3	FFC6$_H$: FFC7$_H$
4	FFC8$_H$: FFC9$_H$
5	FFCA$_H$: FFCB$_H$
6	FFCC$_H$: FFCD$_H$
7	FFCE$_H$: FFCF$_H$

Figure E.1-8 shows an example.

Figure E.1-8 Example of Vector Addressing

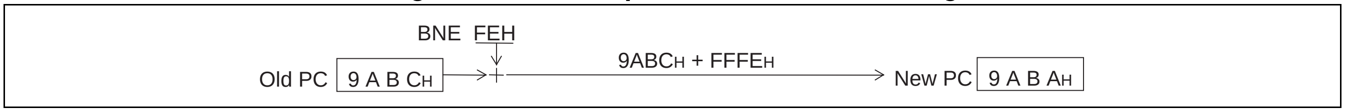


APPENDIX E Instruction Overview

● Relative addressing

This is used when branching to the area in 128 bytes before and behind PC (program counter) with the addressing shown "rel" in the instruction table. In this addressing, add the content of the operand to PC with the sign and store the result in PC. Figure E.1-9 shows an example.

Figure E.1-9 Example of Relative Addressing

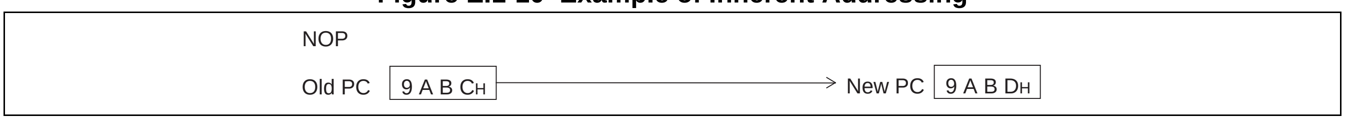


In this example, by jumping to the address where the operation code of BNE is stored, it results in an infinite loop.

● Inherent addressing

This is used when doing the operation decided by the operation code with the addressing that does not have the operand in the instruction table. In this addressing, the operation depends on each instruction. Figure E.1-10 shows an example.

Figure E.1-10 Example of Inherent Addressing



APPENDIX E Instruction Overview

E.2 Special Instruction

This section explains special instructions other than the addressings.

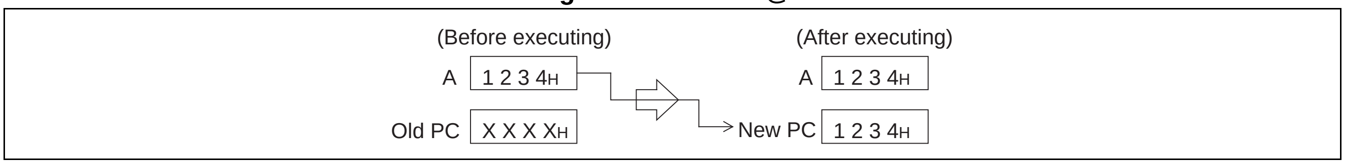
■ Special Instruction

● JMP @A

This instruction is to branch the content of A (accumulator) to PC (program counter) as an address. N pieces of the jump destination is arranged on the table, and one of the contents is selected and transferred to A. N branch processing can be done by executing this instruction.

Figure E.2-1 shows a summary of the instruction.

Figure E.2-1 JMP @A

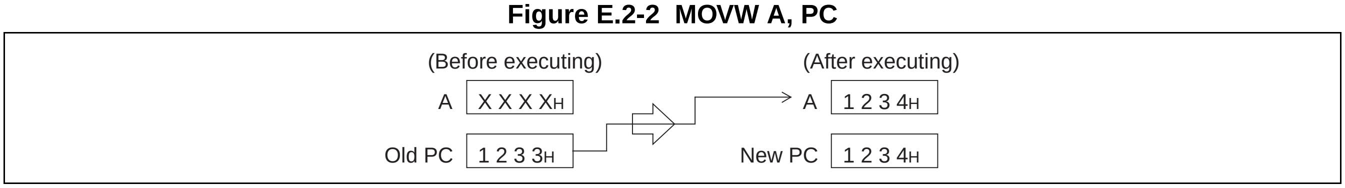


● MOVW A, PC

This instruction works as the opposite of "JMP @A". That is, it stores the content of PC to A. When you have executed this instruction in the main routine and set it to call a specific subroutine, you can make sure that the content of A is the specified value in the subroutine. Also, you can identify that the branch is not from the part that cannot be expected, and use it for the reckless driving judgment.

Figure E.2-2 shows a summary of the instruction.

Figure E.2-2 MOVW A, PC



When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-2, the value "1234_H" stored in A corresponds to the address where the following operation code of "MOVW A, PC" is stored.

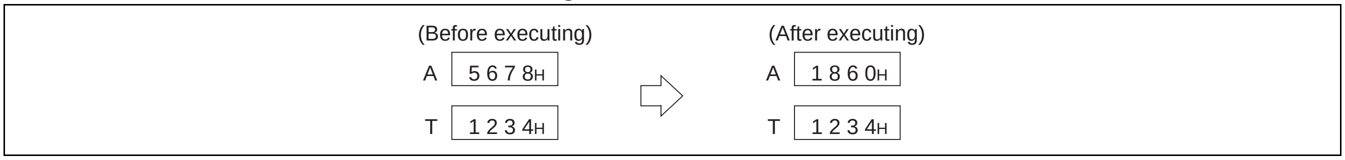
● MULU A

This instruction performs an unsigned multiplication of AL (lower 8-bit of the accumulator) and TL (lower 8-bit of the temporary accumulator), and stores the 16-bit result in A. The contents of T (temporary accumulator) do not change. The contents of AH (higher 8-bit of the accumulator) and TH (higher 8-bit of the temporary accumulator) before execution of the instruction are not used for the operation. The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a multiplication.

APPENDIX E Instruction Overview

Figure E.2-3 shows a summary of the instruction.

Figure E.2-3 MULU A

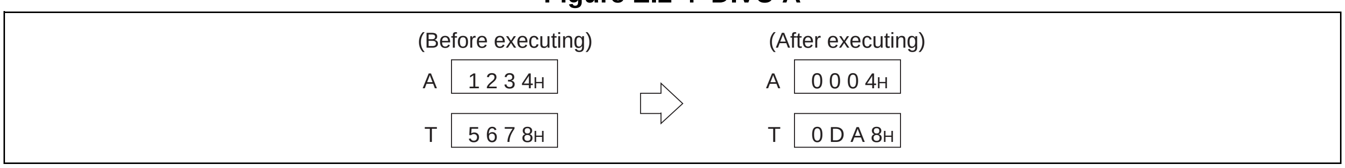


● DIVU A

This instruction divides the 16-bit value in T by the unsigned 16-bit value in A, and stores the 16-bit result and the 16-bit remainder in A and T, respectively. When the value in A before execution of instruction is "0", the Z flag becomes "1" to indicate zero-division is executed. The instruction does not change other flags, and therefore care must be taken when a branch may occur depending on the result of a division.

Figure E.2-4 shows a summary of the instruction.

Figure E.2-4 DIVU A

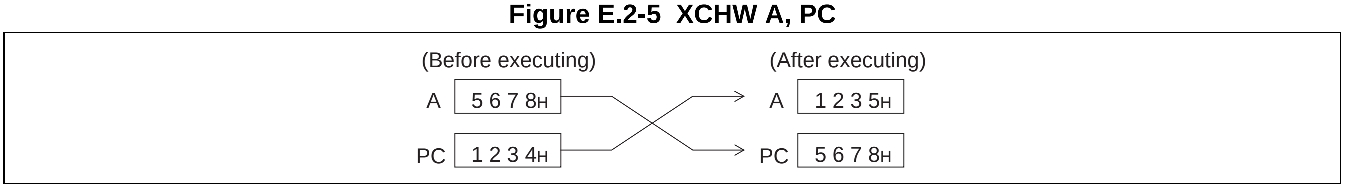


● XCHW A, PC

This instruction swaps the contents of A and PC, resulting in a branch to the address contained in A before execution of the instruction. After the instruction is executed, A becomes the address that follows the address where the operation code of "XCHW A, PC" is stored. This instruction is effective especially when it is used in the main routine to specify a table for use in a subroutine.

Figure E.2-5 shows a summary of the instruction.

Figure E.2-5 XCHW A, PC

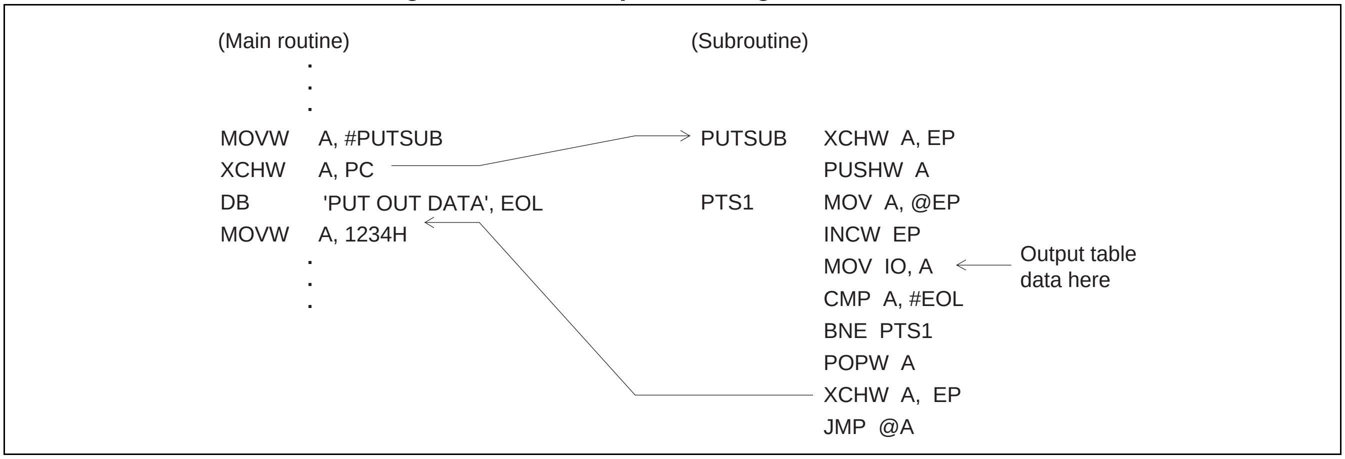


When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-5, the value "1235_H" stored in A corresponds to the address where the following operation code of "XCHW A, PC" is stored. This is why "1235_H" is stored instead of "1234_H".

APPENDIX E Instruction Overview

Figure E.2-6 shows an assembler language example.

Figure E.2-6 Example of Using "XCHW A, PC"

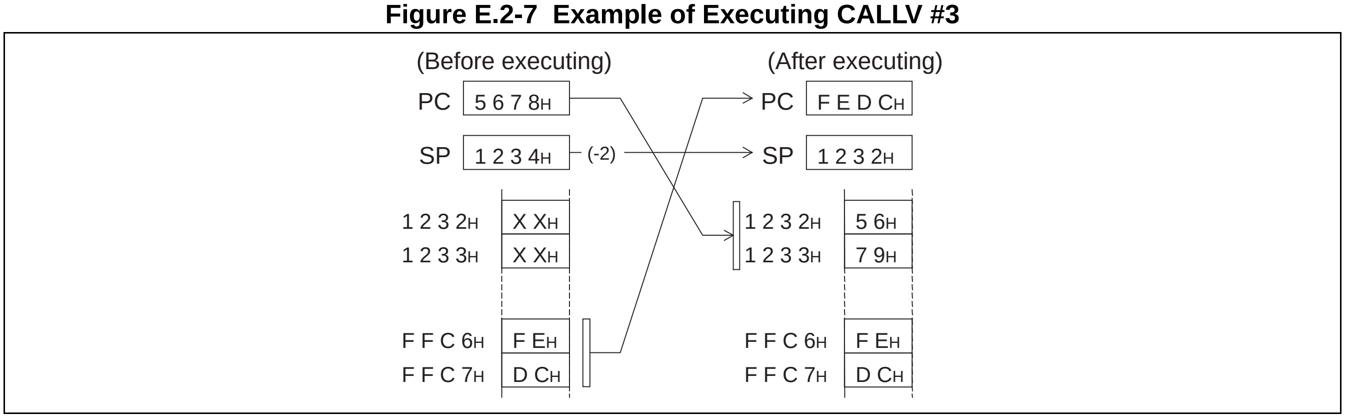


● CALLV #vct

This instruction is used to branch to a subroutine address stored in the vector table. The instruction saves the return address (contents of PC) in the location at the address contained in SP (stack pointer), and uses vector addressing to cause a branch to the address stored in the vector table. Because CALLV #vct is a 1-byte instruction, the use of this instruction for frequently used subroutines can reduce the entire program size.

Figure E.2-7 shows a summary of the instruction.

Figure E.2-7 Example of Executing CALLV #3



After the CALLV #vct instruction is executed, the contents of PC saved on the stack area are the address of the operation code of the next instruction, rather than the address of the operation code of CALLV #vct. Accordingly, Figure E.2-7 shows that the value saved in the stack (1232_H and 1233_H) is 5679_H, which is the address of the operation code of the instruction that follows "CALLV vct" (return address).

APPENDIX E Instruction Overview

Table E.2-1 Vector Table

Vector use (call instruction)	Vector table address	
	Upper	Lower
CALLV #7	$FFCE_H$	$FFCF_H$
CALLV #6	$FFCC_H$	$FFCD_H$
CALLV #5	$FFCA_H$	$FFCB_H$
CALLV #4	$FFC8_H$	$FFC9_H$
CALLV #3	$FFC6_H$	$FFC7_H$
CALLV #2	$FFC4_H$	$FFC5_H$
CALLV #1	$FFC2_H$	$FFC3_H$
CALLV #0	$FFC0_H$	$FFC1_H$

APPENDIX E Instruction Overview

E.3 Bit Manipulation Instructions (SETB, CLRB)

Some peripheral function registers include bits that are read differently than usual by a bit manipulation instruction.

■ Read-modify-write Operation

By using these bit manipulation instructions, you can set only the specified bit in a register or RAM location to "1" (SETB) or clear to "0" (CLRB). However, as the CPU operates data in 8-bit units, the actual operation (read-modify-write operation) involves a sequence of steps: 8-bit data is read, the specified bit is changed, and the data is written back to the location at the original address.

Table E.3-1 shows bus operation for bit manipulation instructions.

Table E.3-1 Bus Operation for Bit Manipulation Instructions

CODE	MNEMONIC	~	Cycle	Address bus	Data bus	RD	WR	RMW
A0 to A7	CLRB dir:b	4	1	N+2	Next instruction	1	0	1
			2	dir address	Data	1	0	1
A8 to AF	SETB dir:b		3	dir address	Data	0	1	0
			4	N+3	Instruction after next	1	0	0

■ Read Destination on the Execution of Bit Manipulation Instructions

For some I/O ports and the interrupt request flag bits, the read destination differs between a normal read operation and a read-modify-write operation.

● I/O ports (during a bit manipulation)

From some I/O ports, an I/O pin value is read during a normal read operation, while a port data register value is read during a bit manipulation. This prevents the other port data register bits from being changed accidentally, regardless of the I/O directions and states of the pins.

● Interrupt request flag bits (during a bit manipulation)

An interrupt request flag bit functions as a flag bit indicating whether an interrupt request exists during a normal read operation, however, "1" is always read from this bit during a bit manipulation. This prevents the flag from being cleared accidentally by writing the value "0" to the interrupt request flag bit when manipulating another bit.

APPENDIX E Instruction Overview

E.4 F2MC-8FX Instructions

Table E.4-1 to Table E.4-4 show the instructions used by the F2MC-8FX.

■ Transfer Instructions

Table E.4-1 Transfer Instructions

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	MOV	dir, A	3	2	(dir) ← (A)	-	-	-	-	-	-	-	45
2	MOV	@IX + off, A	3	2	((IX) + off) ← (A)	-	-	-	-	-	-	-	46
3	MOV	ext, A	4	3	(ext) ← (A)	-	-	-	-	-	-	-	61
4	MOV	@EP, A	2	1	((EP)) ← (A)	-	-	-	-	-	-	-	47
5	MOV	Ri, A	2	1	(Ri) ← (A)	-	-	-	-	-	-	-	48 to 4F
6	MOV	A, #d8	2	2	(A) ← d8	AL	-	-	+	+	-	-	04
7	MOV	A, dir	3	2	(A) ← (dir)	AL	-	-	+	+	-	-	05
8	MOV	A, @IX + off	3	2	(A) ← ((IX) - off)	AL	-	-	+	+	-	-	06
9	MOV	A, ext	4	3	(A) ← (ext)	AL	-	-	+	+	-	-	60
10	MOV	A, @A	2	1	(A) ← ((A))	AL	-	-	+	+	-	-	92
11	MOV	A, @EP	2	1	(A) ← ((EP))	AL	-	-	+	+	-	-	07
12	MOV	A, Ri	2	1	(A) ← (Ri)	AL	-	-	-	+	-	-	08 to 0F
13	MOV	dir, #d8	4	3	(dir) ← d8	-	-	-	-	-	-	-	85
14	MOV	@IX + off, #d8	4	3	((IX) + off) ← d8	-	-	-	-	-	-	-	86
15	MOV	@EP, #d8	3	2	((EP)) ← d8	-	-	-	-	-	-	-	87
16	MOV	Ri, #d8	3	2	(Ri) ← d8	-	-	-	-	-	-	-	88 to 8F
17	MOVW	dir, A	4	2	(dir) ← (AH) , (dir + 1) ← (AL)	-	-	-	-	-	-	-	D5
18	MOVW	@IX + off, A	4	2	((IX) + off) ← (AH) , ((IX) + off + 1) ← (AL)	-	-	-	-	-	-	-	D6
19	MOVW	ext, A	5	3	(ext) ← (AH) , (ext + 1) ← (AL)	-	-	-	-	-	-	-	D4
20	MOVW	@EP, A	3	1	((EP)) ← (AH) , ((EP) + 1) ← (AL)	-	-	-	-	-	-	-	D7
21	MOVW	EP, A	1	1	(EP) ← (A)	-	-	-	-	-	-	-	E3
22	MOVW	A, #d16	3	3	(A) ← d16	AL	AH	dH	+	+	-	-	E4
23	MOVW	A, dir	4	2	(AH) ← (dir) , (AL) ← (dir + 1)	AL	AH	dH	+	+	-	-	C5
24	MOVW	A, @IX + off	4	2	(AH) ← ((IX) + off) , (AL) ← ((IX) + off + 1)	AL	AH	dH	-	+	-	-	C6
25	MOVW	A, ext	5	3	(AH) ← (ext) , (AL) ← (ext + 1)	AL	AH	dH	+	+	-	-	C4
26	MOVW	A, @A	3	1	(AH) ← ((A)) , (AL) ← ((A) + 1)	AL	AH	dH	+	+	-	-	93
27	MOVW	A, @EP	3	1	(AH) ← ((EP)) , (AL) ← ((EP) + 1)	AL	AH	dH	-	+	-	-	C7
28	MOVW	A, EP	1	1	(A) ← (EP)	-	-	dH	-	-	-	-	F3
29	MOVW	EP, #d16	3	3	(EP) ← d16	-	-	-	-	-	-	-	E7
30	MOVW	IX, A	1	1	(IX) ← (A)	-	-	-	-	-	-	-	E2
31	MOVW	A, IX	1	1	(A) ← (IX)	-	-	dH	-	-	-	-	F2
32	MOVW	SP, A	1	1	(SP) ← (A)	-	-	-	-	-	-	-	E1
33	MOVW	A, SP	1	1	(A) ← (SP)	-	-	dH	-	-	-	-	F1
34	MOV	@A, T	2	1	((A)) ← (T)	-	-	-	-	-	-	-	82
35	MOVW	@A, T	3	1	((A)) ← (TH) , ((A) + 1) ← (TL)	-	-	-	-	-	-	-	83
36	MOVW	IX, #d16	3	3	(IX) ← d16	-	-	-	-	-	-	-	E6
37	MOVW	A, PS	1	1	(A) ← (PS)	-	-	dH	-	-	-	-	70
38	MOVW	PS, A	1	1	(PS) ← (A)	-	-	-	+	+	-	+	71
39	MOVW	SP, #d16	3	3	(SP) ← d16	-	-	-	-	-	-	-	E5
40	SWAP		1	1	(AH) ←→ (AL)	-	-	AL	-	-	-	-	10
41	SETB	dir:b	4	2	(dir) : b← 1	-	-	-	-	-	-	-	A8 to AF
42	CLRB	dir:b	4	2	(dir) : b← 0	-	-	-	-	-	-	-	A0 to A7
43	XCH	A, T	1	1	(AL) ←→ (TL)	AL	-	-	-	-	-	-	42
44	XCHW	A, T	1	1	(A) ←→ (T)	AL	AH	dH	-	-	-	-	43
45	XCHW	A, EP	1	1	(A) ←→ (EP)	-	-	dH	-	-	-	-	F7
46	XCHW	A, IX	1	1	(A) ←→ (IX)	-	-	dH	-	-	-	-	F6
47	XCHW	A, SP	1	1	(A) ←→ (SP)	-	-	dH	-	-	-	-	F5
48	MOVW	A, PC	2	1	(A) ← (PC)	-	-	dH	-	-	-	-	F0

APPENDIX E Instruction Overview

Note:

In automatic transfer to T during byte transfer to A, AL is transferred to TL.
If an instruction has plural operands, they are saved in the order indicated by MNEMONIC.

■ Arithmetic Operation Instructions

Table E.4-2 Arithmetic Operation Instruction (1 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	ADDC A, Ri	2	1	(A) ← (A) + (Ri) + C	-	-	-	+	+	+	+	28 to 2F
2	ADDC A, #d8	2	2	(A) ← (A) + d8 + C	-	-	-	+	+	+	+	24
3	ADDC A, dir	3	2	(A) ← (A) + (dir) + C	-	-	-	+	+	+	+	25
4	ADDC A, @IX + off	3	2	(A) ← (A) + ((IX) + off) + C	-	-	-	+	+	+	+	26
5	ADDC A, @EP	2	1	(A) ← (A) + ((EP)) + C	-	-	-	+	+	+	+	27
6	ADDCW A	1	1	(A) ← (A) + (T) + C	-	-	dH	+	+	+	+	23
7	ADDC A	1	1	(AL) ← (AL) + (TL) + C	-	-	-	+	+	+	+	22
8	SUBC A, Ri	2	1	(A) ← (A) - (Ri) - C	-	-	-	+	+	+	+	38 to 3F
9	SUBC A, #d8	2	2	(A) ← (A) - d8 - C	-	-	-	+	+	+	+	34
10	SUBC A, dir	3	2	(A) ← (A) - (dir) - C	-	-	-	+	+	+	+	35
11	SUBC A, @IX + off	3	2	(A) ← (A) - ((IX) + off) - C	-	-	-	+	+	+	+	36
12	SUBC A, @EP	2	1	(A) ← (A) - ((EP)) - C	-	-	-	+	+	+	+	37
13	SUBCW A	1	1	(A) ← (T) - (A) - C	-	-	dH	+	+	+	+	33
14	SUBC A	1	1	(AL) ← (TL) - (AL) - C	-	-	-	+	+	+	+	32
15	INC Ri	3	1	(Ri) ← (Ri) + 1	-	-	-	+	+	+	-	C8 to CF
16	INCW EP	1	1	(EP) ← (EP) + 1	-	-	-	-	-	-	-	C3
17	INCW IX	1	1	(IX) ← (IX) + 1	-	-	-	-	-	-	-	C2
18	INCW A	1	1	(A) ← (A) + 1	-	-	dH	+	+	-	-	C0
19	DEC Ri	3	1	(Ri) ← (Ri) - 1	-	-	-	+	+	+	-	D8 to DF
20	DECW EP	1	1	(EP) ← (EP) - 1	-	-	-	-	-	-	-	D3
21	DECW IX	1	1	(IX) ← (IX) - 1	-	-	-	-	-	-	-	D2
22	DECW A	1	1	(A) ← (A) - 1	-	-	dH	+	+	-	-	D0
23	MULU A	8	1	(A) ← (AL) × (TL)	-	-	dH	-	-	-	-	01
24	DIVU A	17	1	(A) ← (T) / (A) , MOD→ (T)	dL	dH	dH	-	+	-	-	11
25	ANDW A	1	1	(A) ← (A) ∧ (T)	-	-	dH	+	+	R	-	63
26	ORW A	1	1	(A) ← (A) ∨ (T)	-	-	dH	+	+	R	-	73
27	XORW A	1	1	(A) ← (A) ∀ (T)	-	-	dH	+	+	R	-	53
28	CMP A	1	1	(TL) - (AL)	-	-	-	+	+	+	+	12
29	CMPW A	1	1	(T) - (A)	-	-	-	+	+	+	+	13
30	RORC A	1	1	→ C→ A ┐	-	-	-	+	+	-	+	0302
31	ROLCA	1	1	┌ C← A ←	-	-	-	+	+	-	+	
32	CMP A, #d8	2	2	(A) - d8	-	-	-	+	+	+	+	14
33	CMP A, dir	3	2	(A) - (dir)	-	-	-	+	+	+	+	15
34	CMP A, @EP	2	1	(A) - ((EP))	-	-	-	+	+	+	+	17
35	CMP A, @IX + off	3	2	(A) - ((IX) + off)	-	-	-	+	+	+	+	16
36	CMP A, Ri	2	1	(A) - (Ri)	-	-	-	+	+	+	+	18 to 1F
37	DAA	1	1	decimal adjust for addition	-	-	-	+	+	+	+	84
38	DAS	1	1	decimal adjust for subtraction	-	-	-	+	+	+	+	94
39	XOR A	1	1	(A) ← (AL) ∀ (TL)	-	-	-	+	+	R	-	52
40	XOR A, #d8	2	2	(A) ← (AL) ∀ d8	-	-	-	+	+	R	-	54
41	XOR A, dir	3	2	(A) ← (AL) ∀ (dir)	-	-	-	+	+	R	-	55
42	XOR A, @EP	2	1	(A) ← (AL) ∀ ((EP))	-	-	-	+	+	R	-	57
43	XOR A, @IX + off	3	2	(A) ← (AL) ∀ ((IX) + off)	-	-	-	+	+	R	-	56
44	XOR A, Ri	2	1	(A) ← (AL) ∀ (Ri)	-	-	-	+	+	R	-	58 to 5F
45	AND A	1	1	(A) ← (AL) ∧ (TL)	-	-	-	+	+	R	-	62

APPENDIX E Instruction Overview

Table E.4-2 Arithmetic Operation Instruction (2 / 2)

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
46	AND	A, #d8	2	2	(A) ← (AL) ∧ d8	-	-	-	+	+	R	-	64
47	AND	A, dir	3	2	(A) ← (AL) ∧ (dir)	-	-	-	+	+	R	-	65
48	AND	A, @EP	2	1	(A) ← (AL) ∧ ((EP))	-	-	-	+	+	R	-	67
49	AND	A, @IX + off	3	2	(A) ← (AL) ∧ ((IX) + off)	-	-	-	+	+	R	-	66
50	AND	A, Ri	2	1	(A) ← (AL) ∧ (Ri)	-	-	-	+	+	R	-	68 to 6F
51	OR	A	1	1	(A) ← (AL) ∨ (TL)	-	-	-	+	+	R	-	72
52	OR	A, #d8	2	2	(A) ← (AL) ∨ d8	-	-	-	+	+	R	-	74
53	OR	A, dir	3	2	(A) ← (AL) ∨ (dir)	-	-	-	+	+	R	-	75
54	OR	A, @EP	2	1	(A) ← (AL) ∨ ((EP))	-	-	-	+	+	R	-	77
55	OR	A, @IX + off	3	2	(A) ← (AL) ∨ ((IX) + off)	-	-	-	+	+	R	-	76
56	OR	A, Ri	2	1	(A) ← (AL) ∨ (Ri)	-	-	-	+	+	R	-	78 to 7F
57	CMP	dir, #d8	4	3	(dir) - d8	-	-	-	+	+	+	+	95
58	CMP	@EP, #d8	3	2	((EP)) - d8	-	-	-	+	+	+	+	97
59	CMP	@IX + off, #d8	4	3	((IX) + off) - d8	-	-	-	+	+	+	+	96
60	CMP	Ri, #d8	3	2	(Ri) - d8	-	-	-	+	+	+	+	98 to 9F
61	INCW	SP	1	1	(SP) ← (SP) + 1	-	-	-	-	-	-	-	C1
62	DECW	SP	1	1	(SP) ← (SP) - 1	-	-	-	-	-	-	-	D1

■ Branch Instructions

Table E.4-3 Branch Instructions

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	BZ/BEQ	rel(at branch)	4	2	if Z = 1 then PC← PC + rel	-	-	-	-	-	-	-	FD
	BZ/BEQ	rel(at no branch)	2										
2	BNZ/BNE	rel(at branch)	4	2	if Z = 0 then PC← PC + rel	-	-	-	-	-	-	-	FC
	BNZ/BNE	rel(at no branch)	2										
3	BC/BLO	rel(at branch)	4	2	if C = 1 then PC← PC + rel	-	-	-	-	-	-	-	F9
	BC/BLO	rel(at no branch)	2										
4	BNC/BHS	rel(at branch)	4	2	if C = 0 then PC← PC + rel	-	-	-	-	-	-	-	F8
	BNC/BHS	rel(at no branch)	2										
5	BN	rel(at branch)	4	2	if N = 1 then PC← PC + rel	-	-	-	-	-	-	-	FB
	BN	rel(at no branch)	2										
6	BP	rel(at branch)	4	2	if N = 0 then PC← PC + rel	-	-	-	-	-	-	-	FA
	BP	rel(at no branch)	2										
7	BLT	rel(at branch)	4	2	if V ∀ N = 1 then PC← PC + rel	-	-	-	-	-	-	-	FF
	BLT	rel(at no branch)	2										
8	BGE	rel(at branch)	4	2	if V ∀ N = 0 then PC← PC + rel	-	-	-	-	-	-	-	FE
	BGE	rel(at no branch)	2										
9	BBC	dir : b, rel	5	3	if (dir : b) = 0 then PC← PC + rel	-	-	-	-	+	-	-	B0 to B7
10	BBS	dir : b, rel	5	3	if (dir : b) = 1 then PC← PC + rel	-	-	-	-	+	-	-	B8 to BF
11	JMP	@A	3	1	(PC) ← (A)	-	-	-	-	-	-	-	E0
12	JMP	ext	4	3	(PC) ← ext	-	-	-	-	-	-	-	21
13	CALLV	#vct	7	1	vector call	-	-	-	-	-	-	-	E8 to EF
14	CALL	ext	6	3	subroutine call	-	-	-	-	-	-	-	31
15	XCHW	A, PC	3	1	(PC) ← (A) , (A) ← (PC) + 1	-	-	dH	-	-	-	-	F4
16	RET		6	1	return from subroutine	-	-	-	-	-	-	-	20
17	RETI		8	1	return from interrupt	-	-	-		restore			30

■ Other Instructions

Table E.4-4 Other Instructions

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	PUSHW	A	4	1	((SP))← (A), (SP)← (SP) - 2	-	-	-	-	-	-	-	40
2	POPW	A	3	1	(A)← ((SP)), (SP)← (SP) + 2	-	-	dH	-	-	-	-	50
3	PUSHW	IX	4	1	((SP))← (IX), (SP)← (SP) - 2	-	-	-	-	-	-	-	41
4	POPW	IX	3	1	(IX)← ((SP)), (SP)← (SP) + 2	-	-	-	-	-	-	-	51
5	NOP		1	1	No operation	-	-	-	-	-	-	-	00
6	CLRC		1	1	(C)← 0	-	-	-	-	-	-	R	81
7	SETC		1	1	(C)← 1	-	-	-	-	-	-	S	91
8	CLRI		1	1	(I)← 0	-	-	-	-	-	-	-	80
9	SETI		1	1	(I)← 1	-	-	-	-	-	-	-	90

APPENDIX E Instruction Overview

E.5 Instruction Map

Table E.5-1 shows the instruction map of F2MC-8FX.

■ Instruction Map

Table E.5-1 Instruction Map of F2MC-8FX

L \ H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A, ext	MOVW A, PS	CLRI	SETI	CLRB dir : 0	BBC dir : 0, rel	INCW A	DECW A	JMP @A	MOVW A, PC
1	MULU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV ext, A	MOVW PS, A	CLRC	SETC	CLRB dir : 1	BBC dir : 1, rel	INCW SP	DECW SP	MOVW SP, A	MOVW A, SP
2	ROLC A	CMP A	ADDC A	SUBC A	XCH A, T	XOR A	AND A	OR A	MOV @A, T	MOV A, @A	CLRB dir : 2	BBC dir : 2, rel	INCW IX	DECW IX	MOVW IX, A	MOVW A, IX
3	RORC A	CMPW A	ADDCW A	SUBCW A	XCHW A, T	XORW A	ANDW A	ORW A	MOVW @A, T	MOVW A, @A	CLRB dir : 3	BBC dir : 3, rel	INCW EP	DECW EP	MOVW EP, A	MOVW A, EP
4	MOV A, #d8	CMP A, #d8	ADDC A, #d8	SUBC A, #d8		XOR A, #d8	AND A, #d8	OR A, #d8	DAA	DAS	CLRB dir : 4	BBC dir : 4, rel	MOVW A, ext	MOVW ext, A	MOVW A, #d16	XCHW A, PC
5	MOV A, dir	CMP A, dir	ADDC A, dir	SUBC A, dir	MOV dir, A	XOR A, dir	AND A, dir	OR A, dir	MOV dir, #d8	CMP dir, #d8	CLRB dir : 5	BBC dir : 5, rel	MOVW A, dir	MOVW dir, A	MOVW SP, #d16	XCHW A, SP
6	MOV A, @IX+d	CMP A, @IX+d	ADDC A, @IX+d	SUBC A, @IX+d	MOV @IX+d, A	XOR A, @IX+d	AND A, @IX+d	OR A, @IX+d	MOV @IX+d,#d8	CMP @IX+d,#d8	CLRB dir : 6	BBC dir : 6, rel	MOVW A, @IX+d	MOVW @IX+d, A	MOVW IX, #d16	XCHW A, IX
7	MOV A, @EP	CMP A, @EP	ADDC A, @EP	SUBC A, @EP	MOV @EP, A	XOR A, @EP	AND A, @EP	OR A, @EP	MOV @EP, #d8	CMP @EP, #d8	CLRB dir : 7	BBC dir : 7, rel	MOVW A, @EP	MOVW @EP, A	MOVW EP, #d16	XCHW A, EP
8	MOV A, R0	CMP A, R0	ADDC A, R0	SUBC A, R0	MOV R0, A	XOR A, R0	AND A, R0	OR A, R0	MOV R0, #d8	CMP R0, #d8	SETB dir : 0	BBS dir : 0, rel	INC R0	DEC R0	CALLV #0	BNC rel
9	MOV A, R1	CMP A, R1	ADDC A, R1	SUBC A, R1	MOV R1, A	XOR A, R1	AND A, R1	OR A, R1	MOV R1, #d8	CMP R1, #d8	SETB dir : 1	BBS dir : 1, rel	INC R1	DEC R1	CALLV #1	BC rel
A	MOV A, R2	CMP A, R2	ADDC A, R2	SUBC A, R2	MOV R2, A	XOR A, R2	AND A, R2	OR A, R2	MOV R2, #d8	CMP R2, #d8	SETB dir : 2	BBS dir : 2, rel	INC R2	DEC R2	CALLV #2	BP rel
B	MOV A, R3	CMP A, R3	ADDC A, R3	SUBC A, R3	MOV R3, A	XOR A, R3	AND A, R3	OR A, R3	MOV R3, #d8	CMP R3, #d8	SETB dir : 3	BBS dir : 3, rel	INC R3	DEC R3	CALLV #3	BN rel
C	MOV A, R4	CMP A, R4	ADDC A, R4	SUBC A, R4	MOV R4, A	XOR A, R4	AND A, R4	OR A, R4	MOV R4, #d8	CMP R4, #d8	SETB dir : 4	BBS dir : 4, rel	INC R4	DEC R4	CALLV #4	BNZ rel
D	MOV A, R5	CMP A, R5	ADDC A, R5	SUBC A, R5	MOV R5, A	XOR A, R5	AND A, R5	OR A, R5	MOV R5, #d8	CMP R5, #d8	SETB dir : 5	BBS dir : 5, rel	INC R5	DEC R5	CALLV #5	BZ rel
E	MOV A, R6	CMP A, R6	ADDC A, R6	SUBC A, R6	MOV R6, A	XOR A, R6	AND A, R6	OR A, R6	MOV R6, #d8	CMP R6, #d8	SETB dir : 6	BBS dir : 6, rel	INC R6	DEC R6	CALLV #6	BGE rel
F	MOV A, R7	CMP A, R7	ADDC A, R7	SUBC A, R7	MOV R7, A	XOR A, R7	AND A, R7	OR A, R7	MOV R7, #d8	CMP R7, #d8	SETB dir : 7	BBS dir : 7, rel	INC R7	DEC R7	CALLV #7	BLT rel

APPENDIX F Mask Option

The mask option list of the MB95150/M series is shown in Table F-1.

■ Mask Option List

Table F-1 Mask Option List

No.	Part number	MB95156M	MB95F156M MB95F156N MB95F156J	MB95FV100D-103
	Specifying procedure	Specify when ordering MASK	Setting disabled	Setting disabled
1	Clock mode select * • Single clock mode • Dual-system clock mode	Dual-system clock mode	Dual-system clock mode	Changing by the switch on MCU board
2	Low voltage detection reset * • With low voltage detection reset • Without low voltage detection reset	Specify when ordering MASK	Specified by part number	Changing by the switch on MCU board
3	Clock supervisor * • With clock supervisor • Without clock supervisor	Specify when ordering MASK	Specified by part number	Changing by the switch on MCU board
4	Reset output * • With reset output • Without reset output	Specify when ordering MASK	Specified by part number	When "with clock supervisor" is selected, it becomes "without reset output", and when "without clock supervisor" is selected, it becomes it is "without reset output" by the switch of the MCU board.
5	Oscillation stabilization wait time	Fixed to oscillation stabilization wait time of $(2^{14} – 2) /F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} – 2) /F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} – 2) /F_{CH}$

F_{CH}: Main clock

*: As for clock mode selection, low-voltage detection reset, selection of clock supervisor and reset output, see the following table.

Part number	Clock mode selection	Low-voltage detection reset	Clock supervisor	Reset output
MB95156M	Dual-system clock product	None	None	Yes
		Yes	None	Yes
		Yes	Yes	None
MB95F156M	Dual-system clock product	None	None	Yes
MB95F156N		Yes	None	Yes
MB95F156J		Yes	Yes	None
MB95FV100D-103	Single-system clock product	None	None	Yes
		Yes	None	Yes
		Yes	Yes	None
	Dual-system clock product	None	None	Yes
		Yes	None	Yes
		Yes	Yes	None

APPENDIX G Writing to Flash Microcontroller Using Parallel Writer

APPENDIX G Writing to Flash Microcontroller Using Parallel Writer

This section describes writing to flash microcontroller using parallel writer.

■ Writing to Flash Microcontroller Using Parallel Writer

Table G-1 Parallel Writer and Adaptor

Package	Compatible adaptor model	Parallel writer
	Flash Support Group, Inc.	
FPT-48P-M26	TEF110-95F156HPFV	AF9708 (Ver 02.35G higher) AF9709/B (Ver 02.35G higher) AF9723+AF9834 (Ver 02.08E higher)
FPT-52P-M01	TEF110-95F156HPMC	

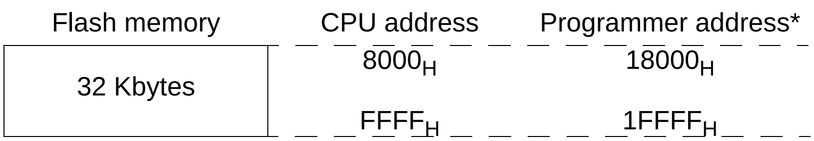
Contact

Flash Support Group, Inc. Tel: +81-53-428-8380

APPENDIX G Writing to Flash Microcontroller Using Parallel Writer

■ Sector Configuration

The following diagram shows the addresses corresponding to each sector on accessing using CPU and parallel writer.



*: Programmer addresses are corresponding to CPU addresses, used when the parallel programmer programs data into Flash memory.
These programmer addresses are used for the parallel programmer to program or erase data in Flash memory.

■ How to Write

1) Set the type code of the parallel programmer to 17222.

2) Load program data to programmer addresses 18000_H to $1FFFF_H$.

3) Programmed by parallel programmer

Index

M

P

R

Register Index

Pin Function Index

Interrupt Vector Index

CM26-10110-5E

FUJITSU MICROELECTRONICS • CONTROLLER MANUAL

F²MC-8FX

8-BIT MICROCONTROLLER

MB95150/M Series

HARDWARE MANUAL

February 2010 the fifrth edition

Published **FUJITSU MICROELECTRONICS LIMITED**

Edited Sales Promotion Dept.