FUJITSU MICROELECTRONICS
CONTROLLER MANUAL

CM26-10120-1E

F²MC-8FX

8-BIT MICROCONTROLLER

MB95200H/210H Series
HARDWARE MANUAL

FUJITSU

F²MC-8FX

8-BIT MICROCONTROLLER

MB95200H/210H Series

HARDWARE MANUAL

The information for microcontroller supports is shown in the following homepage.

Be sure to refer to the "Check Sheet" for the latest cautions on development.

“Check Sheet” is seen at the following support page.

“Check Sheet” lists the minimal requirement items to be checked to prevent problems beforehand in 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 MB95200H/210H 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 MB95200H/210H 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 MB95200H/210H Series of microcontrollers, this manual describes its functions, features, and operations. You should read through the manual.

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

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

■ Trademark

■ Sample Programs

Fujitsu Microelectronics 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://edevice.fujitsu.com/micom/en-support/

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 Microelectronics assumes no liability for any damages whatsoever arising out of the use of sample programs.

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

CONTENTS

FUJITSU

CHAPTER 1
OVERVIEW

This chapter describes the features and basic specifications of the MB95200H/210H Series.

1.1 Features of MB95200H/210H Series

In addition to a compact instruction set, MB95200H/210H is a series of general-purpose single-chip microcontrollers with a variety of peripheral functions.

■ Feature of MB95200H/210H Series

● F^2MC-8FX CPU core

Instruction set optimized for controllers

- Multiplication and division instructions
- 16-bit arithmetic operations
- Bit test branch instructions
- Bit manipulation instructions, etc.

● Clock

- Selectable main clock source
  Main OSC clock (Up to 16.25 MHz, maximum machine clock frequency is 8.125 MHz)
  External clock (Up to 32.5 MHz, maximum machine clock frequency is 16.25 MHz)
  Main internal CR clock (1/8/10/12.5 MHz ± 2%, maximum machine clock frequency is 12.5 MHz)
- Selectable subclock source
  Sub-OSC clock (32.768 kHz)
  External clock (32.768 kHz)
  Sub-internal CR clock (Typ: 100 kHz, Min: 50 kHz, Max: 200 kHz)

● Timer

- 8/16-bit composite timer
- Timebase timer
- Watch prescaler

● LIN-UART **(MB95F204H/F204K/F203H/F203KF202H/F202K/F202H/F202K)**

- Full duplex double buffer
- Capable of clock-synchronized serial data transfer and clock-asynchronized serial data transfer

● External interrupt

- Interrupt by the edge detection (rising edge, falling edge, and both edges can be selected)
- Can be used to wake up the device from different low-power consumption modes (also called standby modes)

● 8/10-bit A/D converter

- 8-bit or 10-bit resolution can be selected

- Low power consumption modes (standby modes)
  - Stop mode
  - Sleep mode
  - Watch mode
  - Timebase timer mode
- I/O port (max: 17) **(MB95F204K/F203K/F202K)**
  - General-purpose I/O ports (max):
    CMOS I/O: 15, N-ch. open drain: 2
- I/O port (max: 16) **(MB95F204H/F203H/F202H)**
  - General-purpose I/O ports (max):
    CMOS I/O: 15, N-ch. open drain: 1
- I/O port (max: 5) **(MB95F214K/F213K/F212K)**
  - General-purpose I/O ports (max):
    CMOS I/O: 3, N-ch. open drain: 2
- I/O port (max: 4) **(MB95F214H/F213H/F212H)**
  - General-purpose I/O ports (max):
    CMOS I/O: 3, N-ch. open drain: 1
- On-chip debug
  - 1-wire serial control
  - Serial writing supported (asynchronous mode)
- Hardware/software watchdog timer
  - Built-in hardware watchdog timer
- Low-voltage detection reset circuit
  - Built-in low-voltage detector
- Clock supervisor counter
  - Built-in clock supervisor counter function
- Programmable port input voltage level
  - CMOS input level / hysteresis input level
- Flash memory security function
  - Protects the contents of flash memory

1.2 Product Line-up of MB95200H/210H Series

Table 1.2-1 lists the product line-up of the MB95200H/210H Series.

■ Product Line-up of MB95200H/210H Series

Table 1.2-1 Product Line-up of MB95200H/210H Series

Parameter \ Part number	MB95F204H	MB95F203H	MB95F202H	MB95F204K	MB95F203K	MB95F202K	MB95F214H	MB95F213H	MB95F212H	MB95F214K	MB95F213K	MB95F212K
Type	Flash memory product											
Clock supervisor counter	It supervises the main clock oscillation.											
ROM capacity	16 KB	8 KB	4 KB	16 KB	8 KB	4 KB	16 KB	8 KB	4 KB	16 KB	8 KB	4 KB
RAM capacity	496 B	496 B	240 B	496 B	496 B	240 B	496 B	496 B	240 B	496 B	496 B	240 B
Low-voltage detection reset	No			Yes			No			Yes		
Reset input	Dedicated			Software select			Dedicated			Software Select		
CPU functions	Number of basic instructions : 136 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 (with machine clock = 16.25 MHz) Interrupt processing time : 0.6 µs (with machine clock = 16.25 MHz)											
General-purpose I/O	I/O ports (max): 16 CMOS: 15, N-ch.: 1			I/O ports (max): 17 CMOS: 15, N-ch.: 2			I/O ports (max): 4 CMOS: 3, N-ch.: 1			I/O ports (max): 5 CMOS: 3, N-ch.: 2		
Timebase timer	Interrupt cycle: 0.256 ms - 8.3 s (when external clock = 4 MHz)											
Hardware/ software watchdog timer	Reset generation cycle Main oscillation clock at 10 MHz : 105 ms (min) The sub-internal CR clock can be used as the source clock of hardware watchdog timer.											
Wild register	It can be used to replace 3 bytes of data.											
LIN-UART	A wide range of communication speed can be selected by a dedicated reload timer. It has a full duplex double buffer. Clock-synchronized serial data transfer and clock-asynchronized serial data transfer is enabled. The LIN function can be used as a LIN master or a LIN slave						No LIN-UART					
8/10-bit A/D converter	6 ch.						2 ch.					
	8-bit or 10-bit resolution can be selected.											
8/16-bit composite timer	2 ch.						1 ch.					
	The timer can be configured as an "8-bit timer x 2 channels" or a "16-bit timer x 1 channel". It has built-in timer function, PWC function, PWM function and capture function. Count clock: it can be selected from internal clocks (7 types) and external clocks. It can output square wave.											
External interrupt	6 ch.						2 ch.					
	Interrupt by edge detection (rising edge, falling edge, or both edges can be selected.) It can be used to wake up the device from standby mode.											
On-chip debug	1-wire serial control It supports serial writing. (asynchronous mode)											

Part number / Parameter	MB95 F204H	MB95 F203H	MB95 F202H	MB95 F204K	MB95 F203K	MB95 F202K	MB95 F214H	MB95 F213H	MB95 F212H	MB95 F214K	MB95 F213K	MB95 F212K
Watch prescaler	Eight different time intervals can be selected.											
Flash memory	It supports automatic programming, Embedded Algorithm, write/erase/erase-suspend/resume commands. It has a flag indicating the completion of the operation of Embedded Algorithm. Number of write/erase cycles (min): 100000 Data retention time: 20 years For write/erase, external Vpp (+10 V) input is required. Flash security feature for protecting the content of the flash.											
Standby mode	Sleep mode, stop mode, watch mode, timebase timer mode											
Package (Width, Length, Height, Pitch)	SDIP-24 SOP-20						DIP-8 SOP-8					

1.3 Differences among Products and Notes on Product Selection

The following describes differences among the products of the MB95200H/210H Sseries product and notes on product selection.

■ Differences among Prodcuts and Notes on Product Selection

- Current consumption

  When using the on-chip debug function, take account of the current consumption of flash erase/program.

  For details of current consumption, refer to "■ ELECTRICAL CHARACTERISTICS" in the data sheet of the MB95200H/210H Series.

- Package

  For details of information on each package, refer to "■ PACKAGES AND CORRESPONDING PRODUCTS" and "■ PACKAGE DIMENSION" in the data sheet of the MB95200H/210H Series.

- Operating voltage

  The operating voltage varies, depending on whether the on-chip debug function is used or not. For details of the operating voltage, refer to "■ ELECTRICAL CHARACTERISTICS" in the data sheet of the MB95200H/210H Series.

- On-chip debug function

  The on-chip debug function requires that V_{CC}, V_{SS} and 1 serial-wire be connected to an evaluation tool. In addition, if the flash memory data has to be updated, the RSTX/PF2 pin must also be connected to the same evaluation tool.

1.4 Block Diagrams of MB95200H/210H Series

Figure 1.4-1 and Figure 1.4-2 are block diagrams of the MB95200H/210H Series.

■ Block Diagrams of MB95200H/210H Series

Figure 1.4-1 Block Diagram of MB95200H/210H Series (20-pin MCU)

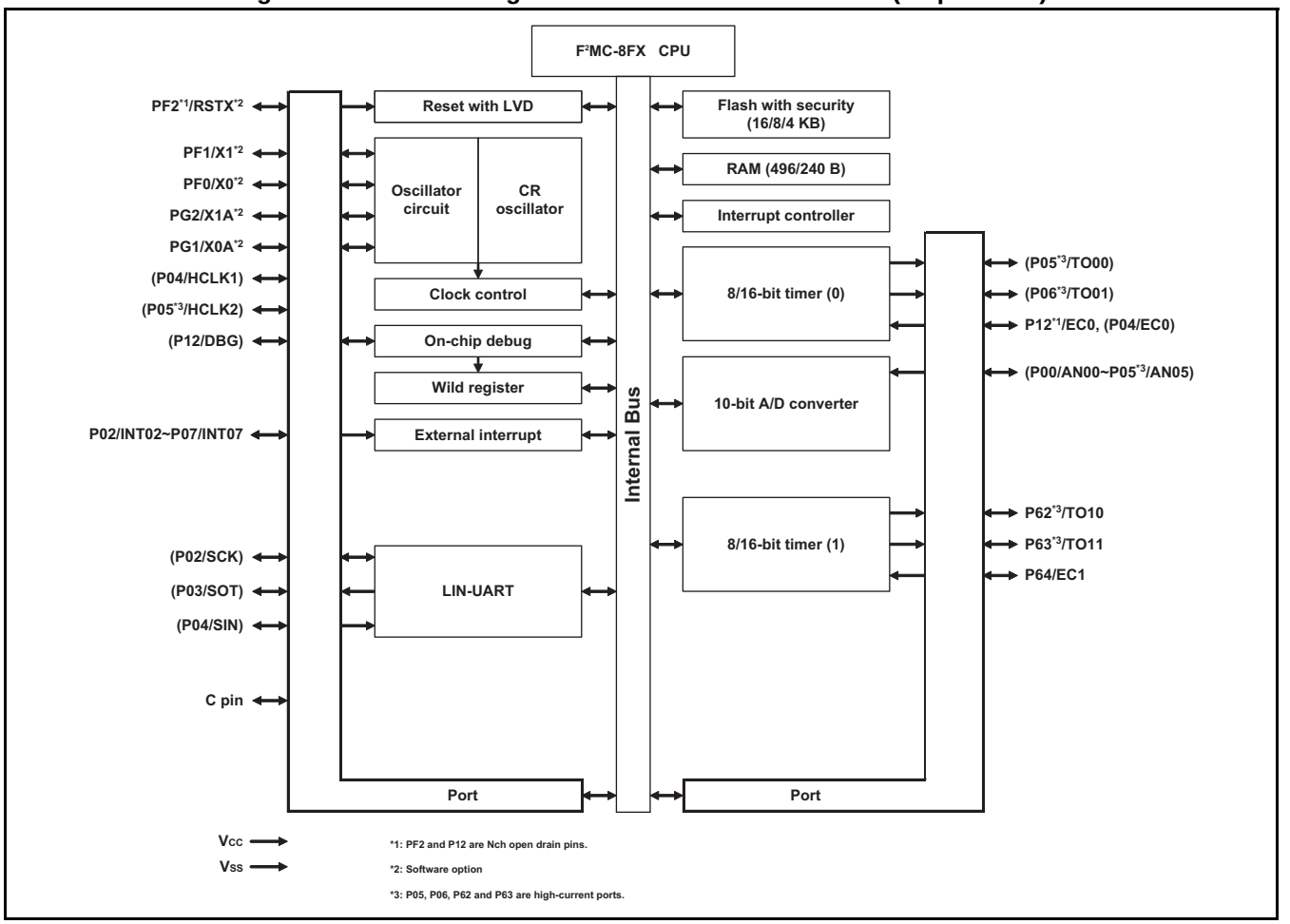
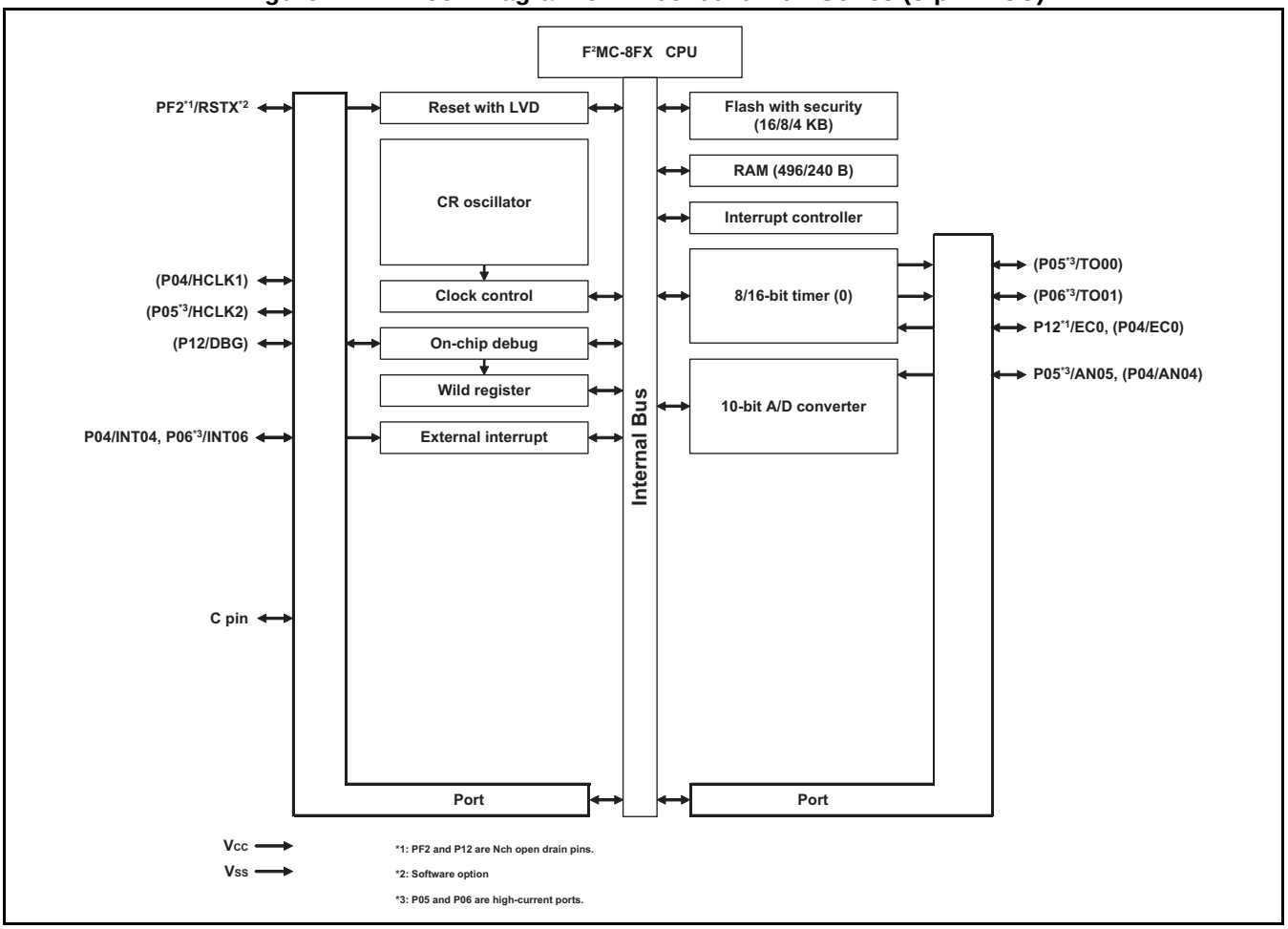


Figure 1.4-2 Block Diagram of MB95200H/210H Series (8-pin MCU)

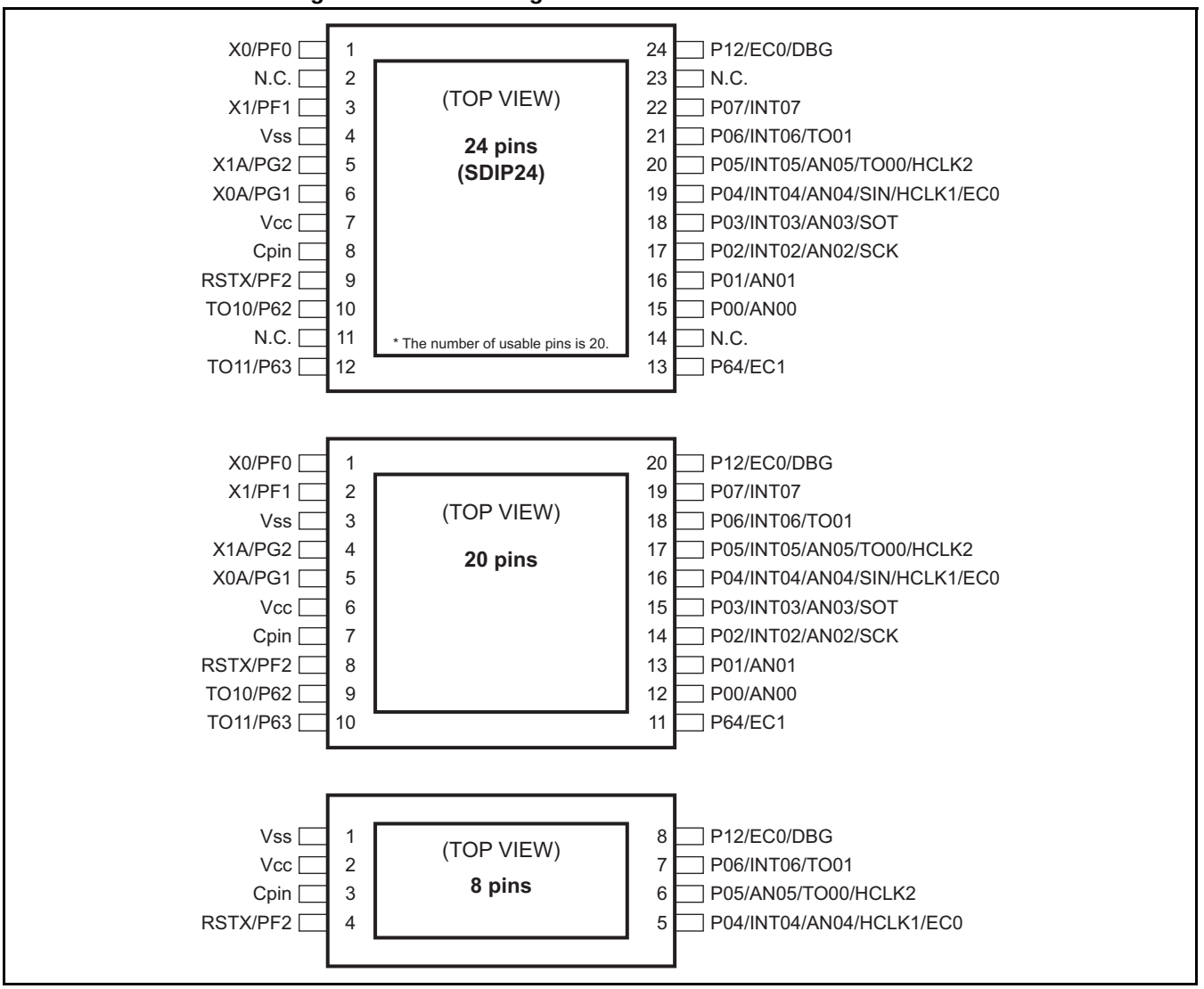


1.5 Pin Assignment

Figure 1.5-1 shows the pin assignment of the MB95200H/210H Series.

■ Pin Assignment of MB95200H/210H Series

Figure 1.5-1 Pin Assignment of MB95200H/210H Series

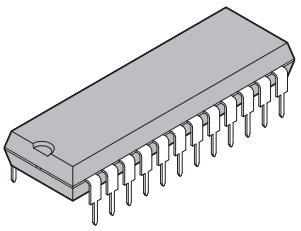


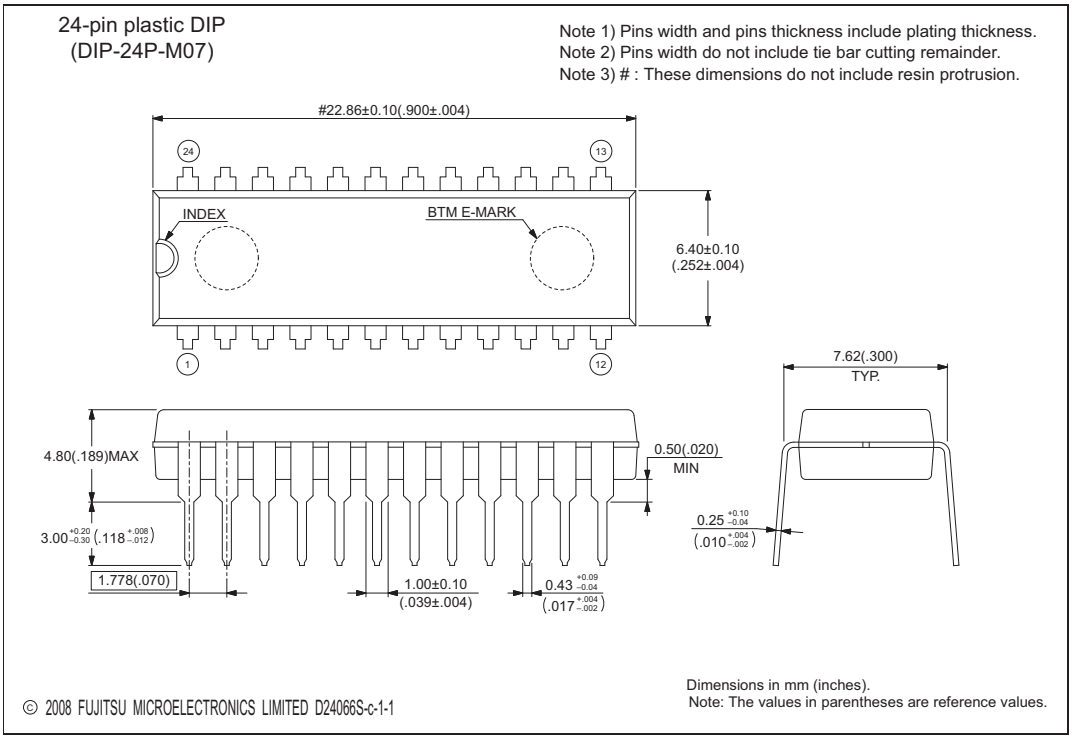
1.6 Package Dimensions

The MB95200H/210H Series is available in four types of package.

■ Package Dimensions of DIP-24P-M07

Figure 1.6-1 Package Dimensions of DIP-24P-M07

24-pin plastic DIP  (DIP-24P-M07)	Lead pitch	1.778 mm
	Package width × package length	6.40 mm × 22.86 mm
	Sealing method	Plastic mold
	Mounting height	4.80 mm Max

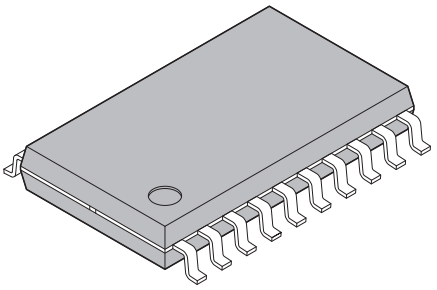


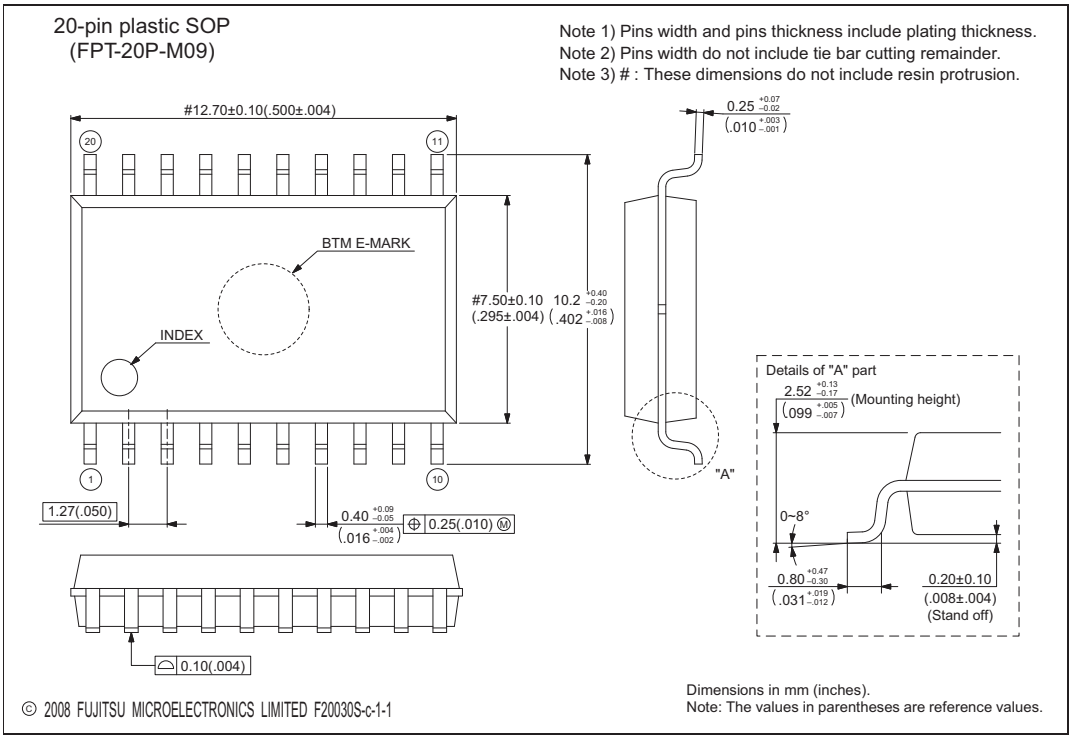
Please check the latest package dimensions at the following URL.

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

■ Package Dimensions of FPT-20P-M09

Figure 1.6-2 Package Dimensions of FPT-20P-M09

20-pin plastic SOP  (FPT-20P-M09)	Lead pitch	1.27 mm
	Package width × package length	7.50 mm × 12.70 mm
	Lead shape	Gullwing
	Lead bend direction	Normal bend
	Sealing method	Plastic mold
	Mounting height	2.65 mm Max

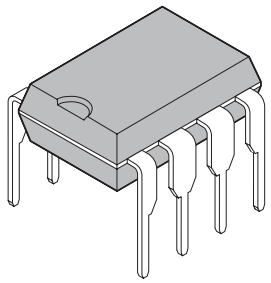


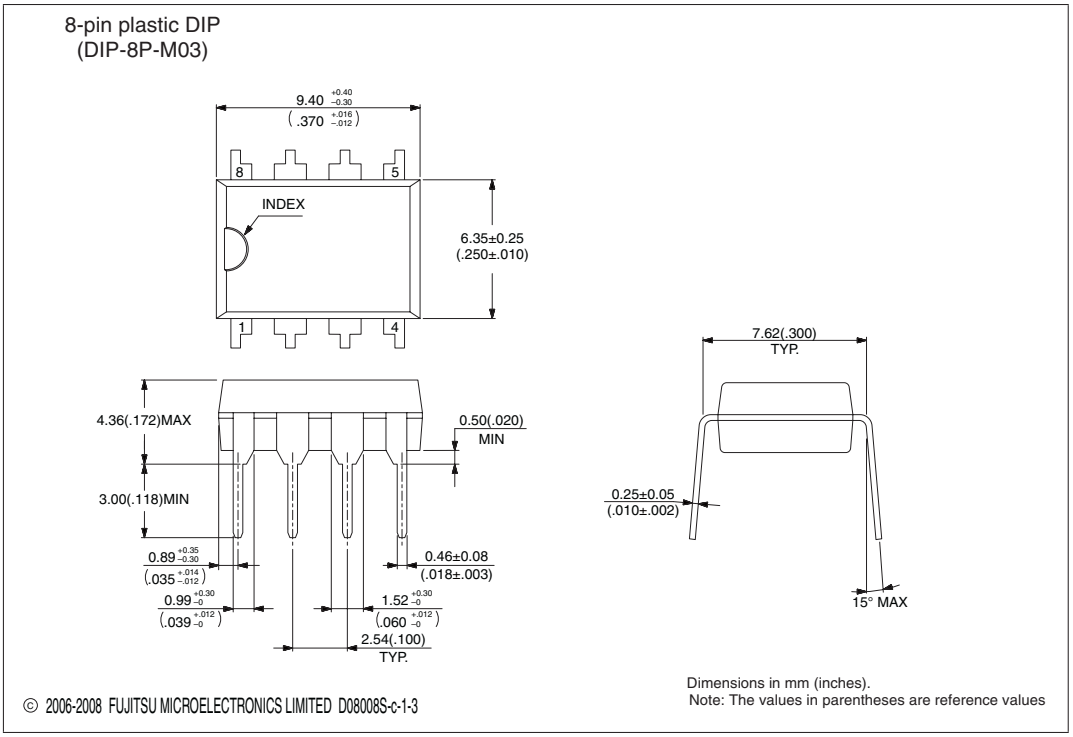
Please check the latest package dimensions at the following URL.

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

■ Package Dimensions of DIP-8P-M03

Figure 1.6-3 Package Dimensions of DIP-8P-M03

8-pin plastic DIP  (DIP-8P-M03)	Lead pitch	2.54 mm
	Sealing method	Plastic mold

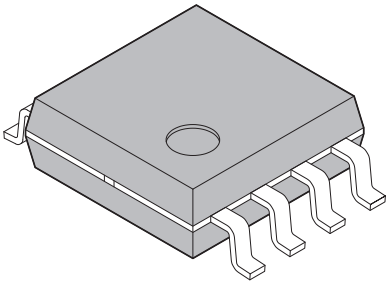


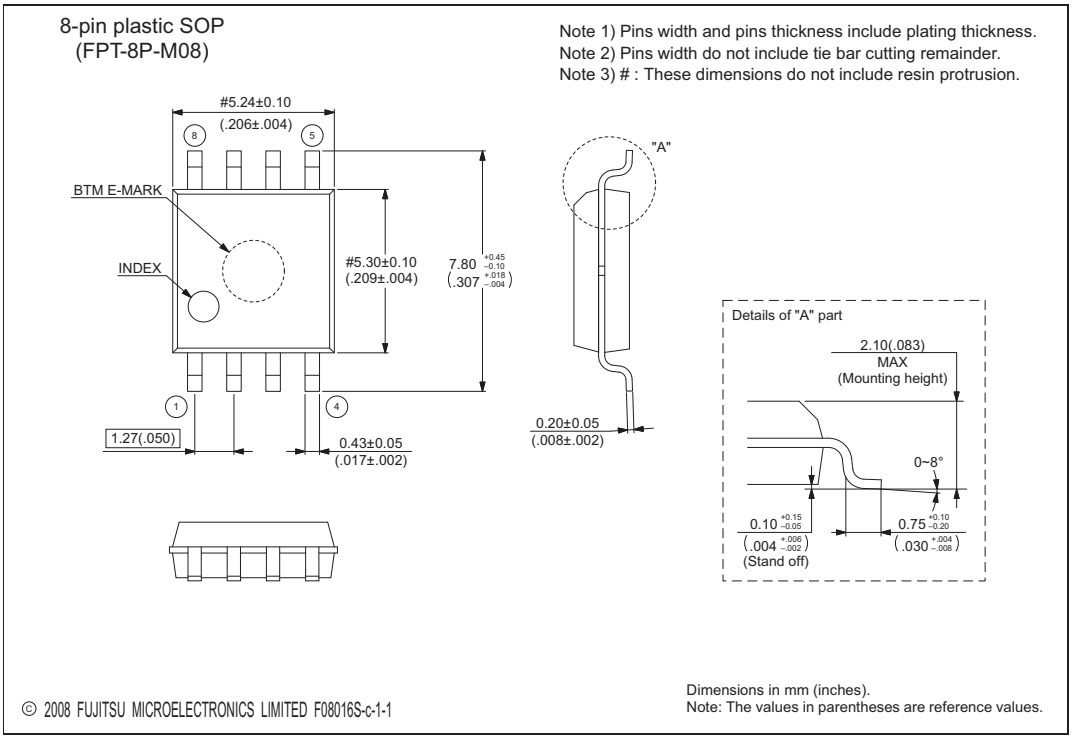
Please check the latest package dimensions at the following URL.

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

■ Package Dimensions of FPT-8P-M08

Figure 1.6-4 Package Dimensions of FPT-8P-M08

8-pin plastic SOP		
	Lead pitch	1.27 mm
	Package width × package length	5.30 mm × 5.24 mm
	Lead shape	Gullwing
	Lead bend direction	Normal bend
	Sealing method	Plastic mold
	Mounting height	2.10 mm Max
(FPT-8P-M08)		



Please check the latest package dimensions at the following URL.

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

1.7 Pin Description

Table 1.7-1 and Table 1.7-2 show pin description. The alphabet in "I/O circuit type" column of Table 1.7-1 and Table 1.7-2 corresponds to the one in "Type" column of Table 1.8-1.

■ Pin Description (20-pin MCU)

Table 1.7-1 Pin Description (20-pin MCU) (1 / 2)

Pin no.	Pin name	I/O circuit type*	Function
1	PF0/X0	B	General-purpose I/O port This pin is also used as the main clock input oscillation pin.
2	PF1/X1	B	General-purpose I/O port This pin is also used as the main clock input/output oscillation pin.
3	V_{SS}	—	Power supply pin (GND)
4	PG2/X1A	C	General-purpose I/O port This pin is also used as the subclock input/output oscillation pin.
5	PG1/X0A	C	General-purpose I/O port This pin is also used as the subclock input oscillation pin.
6	V_{CC}	—	Power supply pin
7	Cpin	—	Capacitor connection pin
8	PF2/RSTX	A	General-purpose I/O port This pin is also used as a reset pin. This pin is a dedicated reset pin in MB95F204H/F203H/F202H.
9	P62/TO10	D	General-purpose I/O port High-current port This pin is also used as the 8/16-bit composite timer ch. 1 output.
10	P63/TO11	D	General-purpose I/O port High-current port This pin is also used as the 8/16-bit composite timer ch. 1 output.
11	P64/EC1	D	General-purpose I/O port This pin is also used as the 8/16-bit composite timer ch. 1 clock input.
12	P00/AN00	E	General-purpose I/O port This pin is also used as the A/D converter analog input.
13	P01/AN01	E	General-purpose I/O port This pin is also used as the A/D converter analog input.

Table 1.7-1 Pin Description (20-pin MCU) (2 / 2)

Pin no.	Pin name	I/O circuit type*	Function
14	P02/INT02/AN02/ SCK	E	General-purpose I/O port This pin is also used as the external interrupt input. This pin is also used as the A/D converter analog input. This pin is also used as the LIN-UART clock I/O.
15	P03/INT03/AN03/ SOT	E	General-purpose I/O port This pin is also used as the external interrupt input. This pin is also used as the A/D converter analog input. This pin is also used as the LIN-UART data output.
16	P04/INT04/AN04/ SIN/HCLK1/EC0	F	General-purpose I/O port This pin is also used as the external interrupt input. This pin is also used as the A/D converter analog input. This pin is also used as the LIN-UART data input. This pin is also used as the external clock input. This pin is also used as the 8/16-bit composite timer ch. 0 clock input.
17	P05/INT05/AN05/ TO00/HCLK2	E	General-purpose I/O port High-current port This pin is also used as the external interrupt input. This pin is also used as the A/D converter analog input. This pin is also used as the 8/16-bit composite timer ch. 0 output. This pin is also used as the external clock input.
18	P06/INT06/TO01	G	General-purpose I/O port High-current port This pin is also used as the external interrupt input. This pin is also used as the 8/16-bit composite timer ch. 0 output.
19	P07/INT07	G	General-purpose I/O port This pin is also used as the external interrupt input.
20	P12/EC0/DBG	H	General-purpose I/O port This pin is also used as the DBG input pin. This pin is also used as the 8/16-bit composite timer ch. 0 clock input.

*: For the I/O circuit types, see "1.8 I/O Circuit Types".

■ Pin Description (8-pin MCU)

Table 1.7-2 Pin Description (8-pin MCU)

Pin no.	Pin name	I/O circuit type*	Function
1	V_{SS}	—	Power supply pin (GND)
2	V_{CC}	—	Power supply pin
3	Cpin	—	Capacitor connection pin
4	RSTX/PF2	A	General-purpose I/O port This pin is also used as a reset pin. This pin is a dedicated reset pin in MB95F214H/F213H/F212H.
5	P04/INT04/AN04/ HCLK1/EC0	E	General-purpose I/O port This pin is also used as the external interrupt input. This pin is also used as the A/D converter analog input. This pin is also used as the external clock input. This pin is also used as the 8/16-bit composite timer ch. 0 clock input.
6	P05/AN05/TO00/ HCLK2	E	General-purpose I/O port High-current port This pin is also used as the A/D converter analog input. This pin is also used as the 8/16-bit composite timer ch. 0 output. This pin is also used as the external clock input.
7	P06/INT06/TO01	G	General-purpose I/O port High-current port This pin is also used as the external interrupt input. This pin is also used as the 8/16-bit composite timer ch. 0 output.
8	P12/EC0/DBG	H	General-purpose I/O port This pin is also used as the DBG input pin. This pin is also used as the 8/16-bit composite timer ch. 0 clock input.

*: For the I/O circuit types, see " 1.8 I/O Circuit Types".

1.8 I/O Circuit Types

Table 1.8-1 lists the I/O circuit types. The alphabet in "Type" column of Table 1.8-1 corresponds to the one in "I/O circuit type" column of Table 1.7-1 and Table 1.7-2.

■ I/O Circuit Types

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

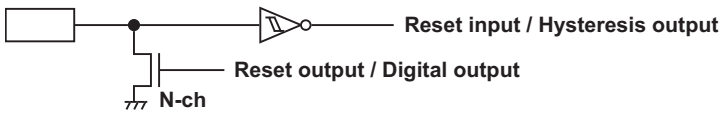
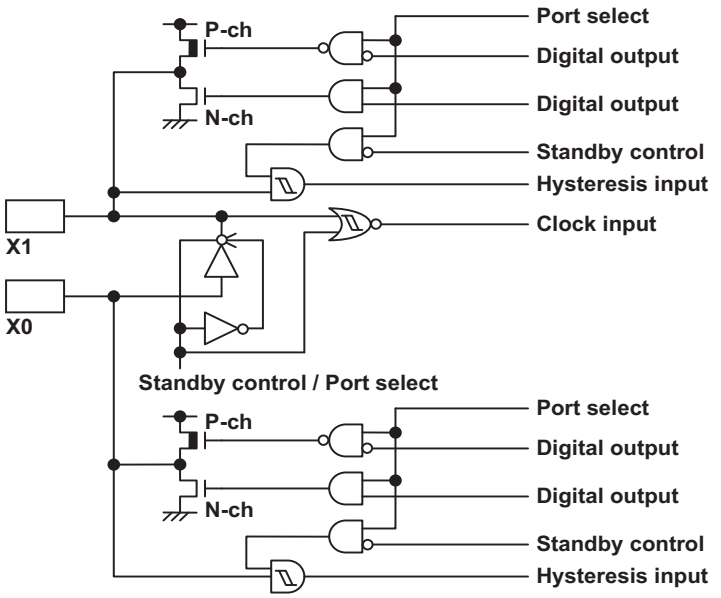
Type	Circuit	Remarks
A		• N-ch. open drain output • Hysteresis input • Reset output
B		• Oscillation circuit • High-speed side Feedback resistance: approx. 1 MΩ • CMOS output • Hysteresis input

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

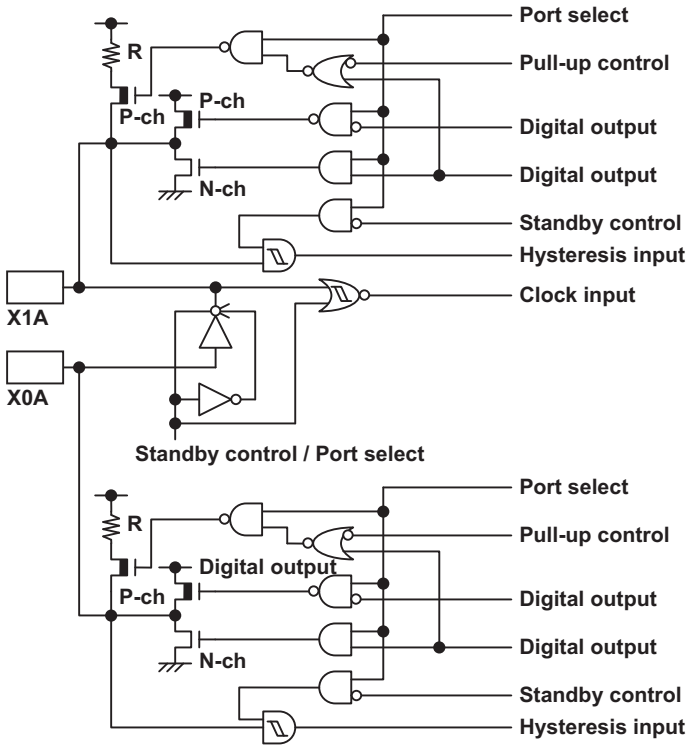
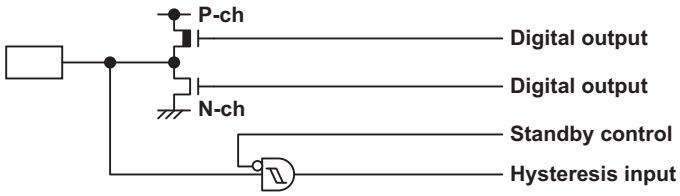
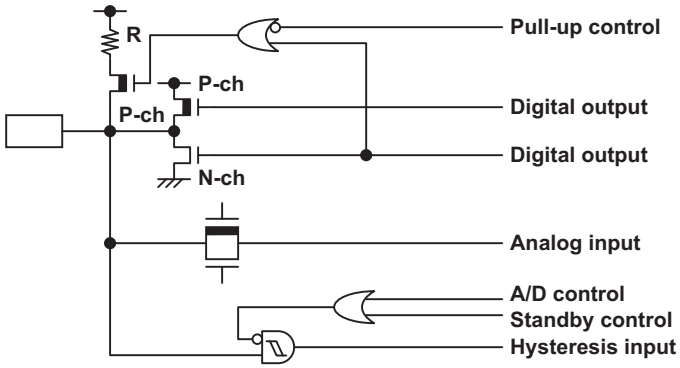
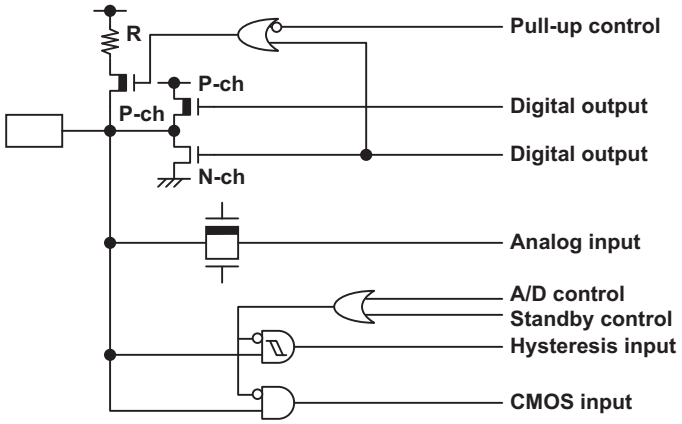
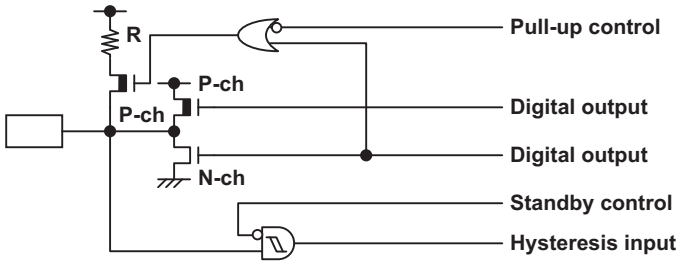
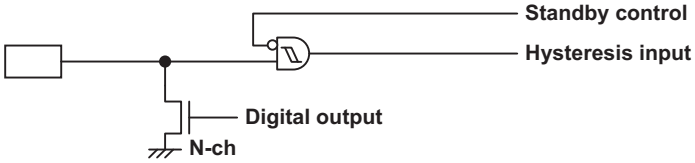
Type	Circuit	Remarks
C		• Oscillation circuit • Low-speed side Feedback resistance: approx. 10 MΩ • CMOS output • Hysteresis input • Pull-up control available
D		• CMOS output • Hysteresis input
E		• CMOS output • Hysteresis input • Pull-up control available

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

Type	Circuit	Remarks
F		• CMOS output • Hysteresis input • CMOS input • Pull-up control available
G		• Hysteresis input • CMOS output • Pull-up control available
H		• N-ch. open drain output • Hysteresis input

CHAPTER 2
NOTES ON DEVICE HANDLING

This chapter gives notes on using the MB95200H/210H Series.

2.1 Notes on Device Handling

This section provides notes on power supply voltage and pin treatment.

■ HANDLING DEVICES

- Preventing latch-ups

  When using the device, ensure that the voltage applied does not exceed the maximum voltage rating.

  In a CMOS IC, if a voltage higher than V_{CC} or a voltage lower than V_{SS} is applied to an input/output pin that is neither a medium-withstand voltage pin nor a high-withstand voltage pin, or if a voltage out of the rating range of power supply voltage mentioned in "1. Absolute Maximum Ratings" of "■ ELECTRICAL CHARACTERISTICS" in the data sheet of the MB95200H/210H Series is applied to the V_{CC} pin or the V_{SS} pin, a latch-up may occur.

  When a latch-up occurs, power supply current increases significantly, which may cause a component to be thermally destroyed.

- Stabilizing supply voltage

  Supply voltage must be stabilized.

  A malfunction may occur when power supply voltage fluctuates rapidly even though the fluctuation is within the guaranteed operating range of the V_{CC} power supply voltage.

  As a rule of voltage stabilization, suppress voltage fluctuation so that the fluctuation in V_{CC} ripple (p-p value) at the commercial frequency (50 Hz/60 Hz) does not exceed 10% of the standard V_{CC} value, and the transient fluctuation rate does not exceed 0.1 V/ms at a momentary fluctuation such as switching the power supply.

- Notes on using the external clock

  When an external clock is used, oscillation stabilization wait time is required for power-on reset, wakeup from subclock mode or stop mode.

■ PIN CONNECTION

- Treatment of unused pins

  If an unused input pin is left unconnected, a component may be permanently damaged due to malfunctions or latch-ups. Always pull up or pull down an unused input pin through a resistor of at least 2 kΩ. Set an unused input/output pin to the output state and leave it unconnected, or set it to the input state and treat it the same as an unused input pin. If there is an unused output pin, leave it unconnected.

- Power supply pins

  To reduce unnecessary electro-magnetic emission, prevent malfunctions of strobe signals due to an increase in the ground level, and conform to the total output current standard, always connect the V_{CC} pin and the V_{SS} pin to the power supply and ground outside the device. In addition, connect the current supply source to the V_{CC} pin and the V_{SS} pin with low impedance.

  It is also advisable to connect a ceramic bypass capacitor of approximately 0.1 µF between the V_{CC} pin and the V_{SS} pin at a location close to this device.

- DBG pin

  Connect the DBG pin directly to an external pull-up resistor.

  To prevent the device from unintentionally entering the debug mode due to noise, minimize the distance between the DBG pin and the V_{CC} or V_{SS} pin when designing the layout of the printed circuit board.

  The DBG pin should not stay at “L” level after power-on until the reset output is released.

- RSTX pin

  Connect the RSTX pin directly to an external pull-up resistor.

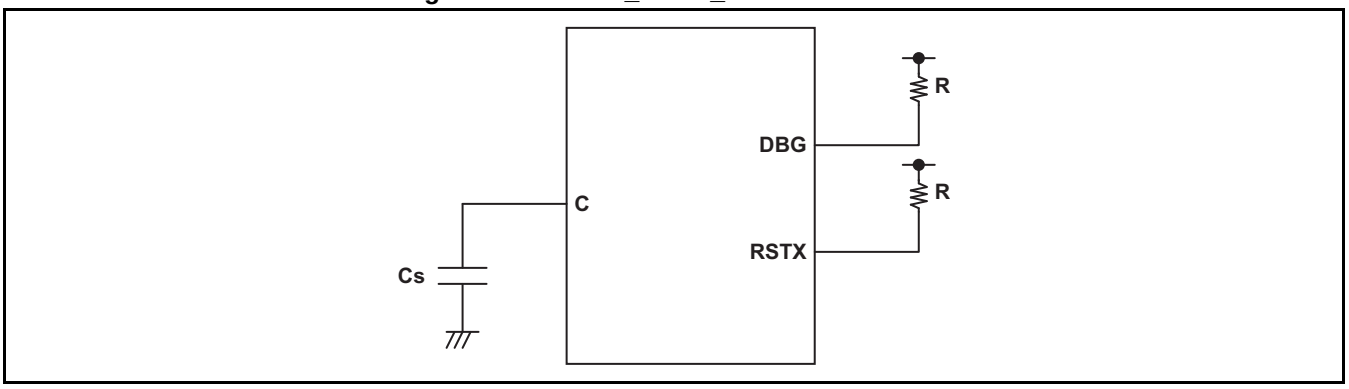
  To prevent the device from unintentionally entering the reset mode due to noise, minimize the distance between the RSTX pin and the V_{CC} or V_{SS} pin when designing the layout of the printed circuit board.

  The RSTX/PF2 pin functions as the reset input/output pin after power-on. In addition, the reset output can be enabled by the RSTOE bit of the SYSC register, and the reset input function or the general purpose I/O function can be selected by the RSTEN bit of the SYSC register.

- C pin

  Use a ceramic capacitor or a capacitor with equivalent frequency characteristics. The bypass capacitor for the V_{CC} pin must have a capacitance larger than C_S. For the connection to a smoothing capacitor C_S, see the diagram below. To prevent the device from unintentionally entering an unknown mode due to noise, minimize the distance between the C pin and C_S and the distance between C_S and the V_{SS} pin when designing the layout of a printed circuit board.

Figure 2.1-1 DBG_RSTX_C Pin Connection



CHAPTER 3
MEMORY SPACE

This chapter describes the memory space.

3.1 Memory Space

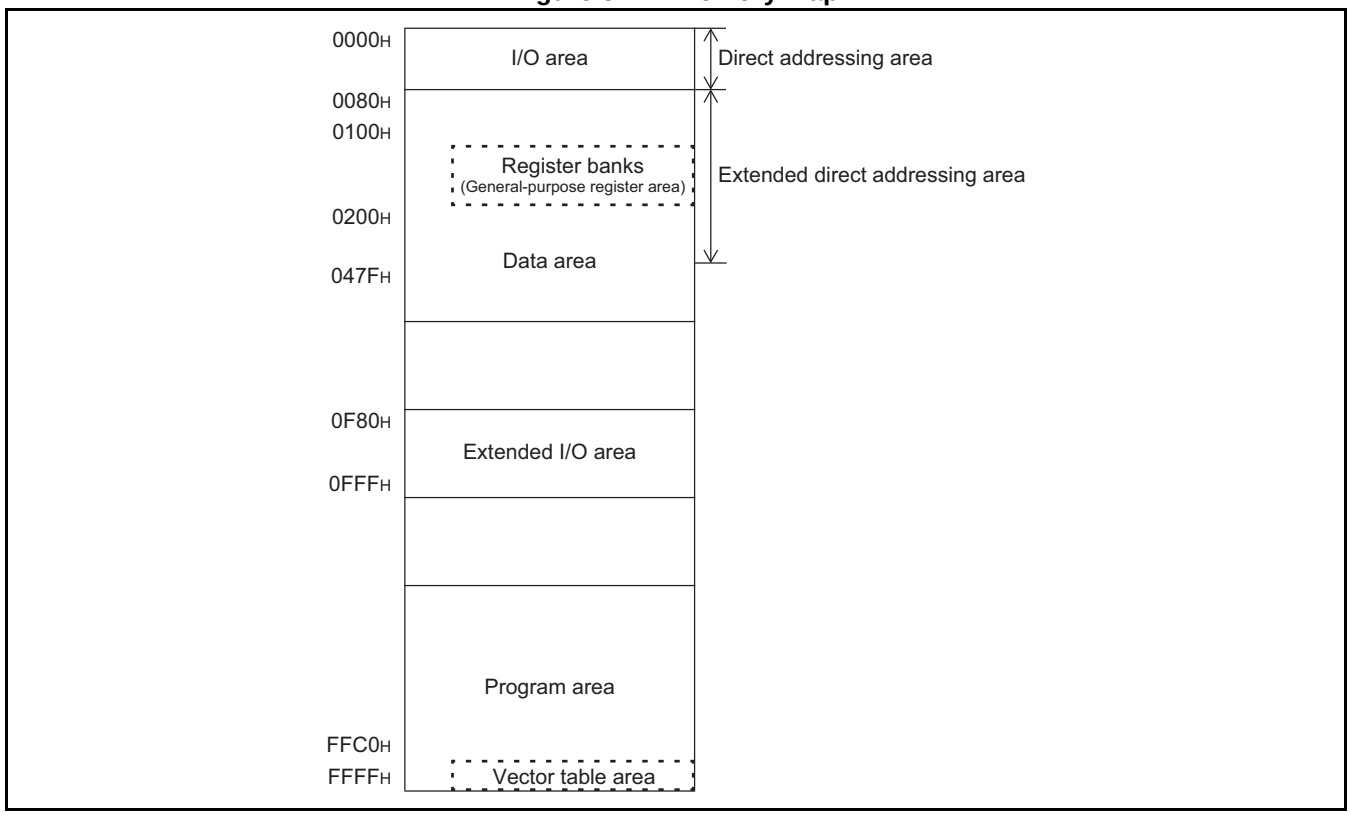
The memory space of the MB95200H/210H Series is 64 KB in size and consists of an I/O area, a data area, and a program area. The memory space includes areas for specific applications such as general-purpose registers and a vector table.

■ Configuration of Memory Space

- I/O area (addresses: 0000_H to $007F_H$)
  - This area contains the control registers and data registers for built-in peripheral functions.
  - As the I/O area forms part of the memory space, it can be accessed in the same way as the memory. It can also be accessed at high-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 built-in peripheral functions.
  - As the extended I/O area forms part of the memory space, it can be accessed in the same way as the memory.
- Data area
  - Static RAM is incorporated in the data area as the internal data area.
  - The internal RAM size varies according to product.
  - The RAM area from 0090_H to $00FF_H$ can be accessed at high-speed by using direct addressing instructions.
  - The area from 0100_H to $027F_H$ is an extended direct addressing area. It can be accessed at high-speed by direct addressing instructions with a direct bank pointer set. (MB95F204H/F204K/F214H/F214K/F203H/F203K/F213H/F213K)
  - The area from 0100_H to $017F_H$ is an extended direct addressing area. It can be accessed at high-speed by direct addressing instructions with a direct bank pointer set. (MB95F202H/F202K/F212H/F212K)
  - The area from 0100_H to $01FF_H$ can be used as a general-purpose register area. (MB95F204H/F204K/F214H/F214K/F203H/F203K/F213H/F213K)
  - The area from 0100_H to $017F_H$ can be used as a general-purpose register area. (MB95F202H/F202K/F212H/F212K)
- Program area
  - ROM is incorporated in the program area as the internal program area.
  - The internal ROM size varies according to product.
  - The area from $FFC0_H$ to $FFFF_H$ is used as the vector table.
  - The area from $FFBC_H$ to $FFBF_H$ is used to store data of the non-volatile register.

■ 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 $01FF_H$ in MB95F204H/F204K/F214H/F214K/F203H/F203K/F213H/F213K) (Addresses: 0100_H to $017F_H$ in MB95F202H/F202K/F212H/F212K)

- This area contains the auxiliary registers used for 8-bit arithmetic operations, transfer, etc.
- As this area forms part of the RAM area, it can also be used as conventional RAM.
- When the area is used as general-purpose registers, general-purpose register addressing enables high-speed access with short instructions.

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

■ Non-volatile Register Data Area (Addresses: $FFBC_H$ to $FFBF_H$)

- The area from $FFBC_H$ to $FFBF_H$ is used to store data of the non-volatile register. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

■ 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 top of the ROM area is allocated to the vector table area. The start address of a service routine is set to an address in the vector table in the form of data.

Table 8.1-1 in "CHAPTER 8 INTERRUPTS" lists the vector table addresses corresponding to vector call instructions, interrupts, and resets.

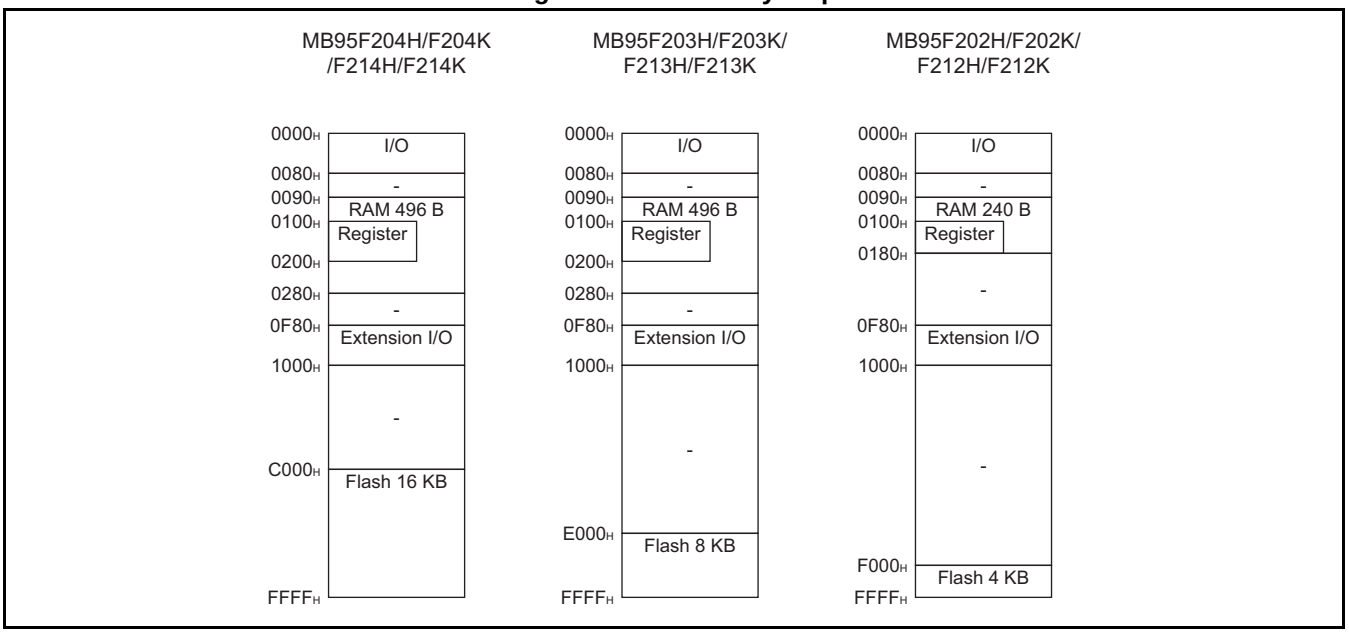
For details, see "CHAPTER 7 RESET", "CHAPTER 8 INTERRUPTS", and "■ Special Instruction ● CALLV #vct" in section "E.2 Special Instruction" in "APPENDIX".

3.2 Memory Map

This section shows a memory map of the MB95200H/210H Series.

■ Memory Map

Figure 3.2-1 Memory Map



Parameter / Part number	Flash memory	RAM
MB95F204H/F204K/F214H/F214K	16 KB	496 B
MB95F203H/F203K/F213H/F213K	8 KB	496 B
MB95F202H/F202K/F212H/F212K	4 KB	240 B

CHAPTER 4

MEMORY ACCESS MODE

This chapter describes the memory access mode.

4.1 Memory Access Mode

The MB95200H/210H Series supports only one memory access mode: single-chip mode.

■ Single-chip Mode

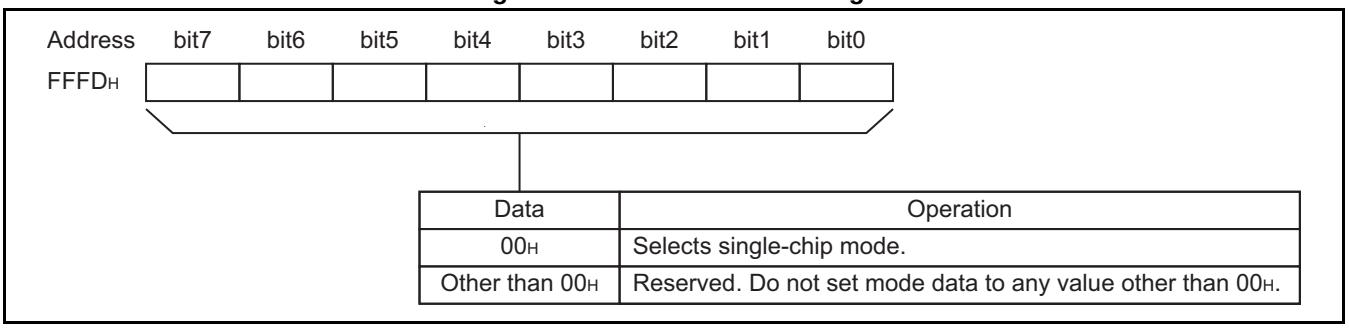
In single-chip mode, only the internal RAM and ROM are used, and no external bus access is executed.

● Mode data

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

The mode data address is fixed at "$FFFD_H$". Always set the mode data of the internal ROM to "00_H" to select the single-chip mode.

Figure 4.1-1 Mode Data Settings



Data	Operation
00_H	Selects single-chip mode.
Other than 00_H	Reserved. Do not set mode data to any value other than 00_H.

After a reset is released, the CPU fetches mode data first.

The CPU then fetches the reset vector after the mode data. It starts executing instructions from the address set in the reset vector.

CHAPTER 5

CPU

This chapter describes the functions and operations of the CPU.

5.1 Dedicated Registers

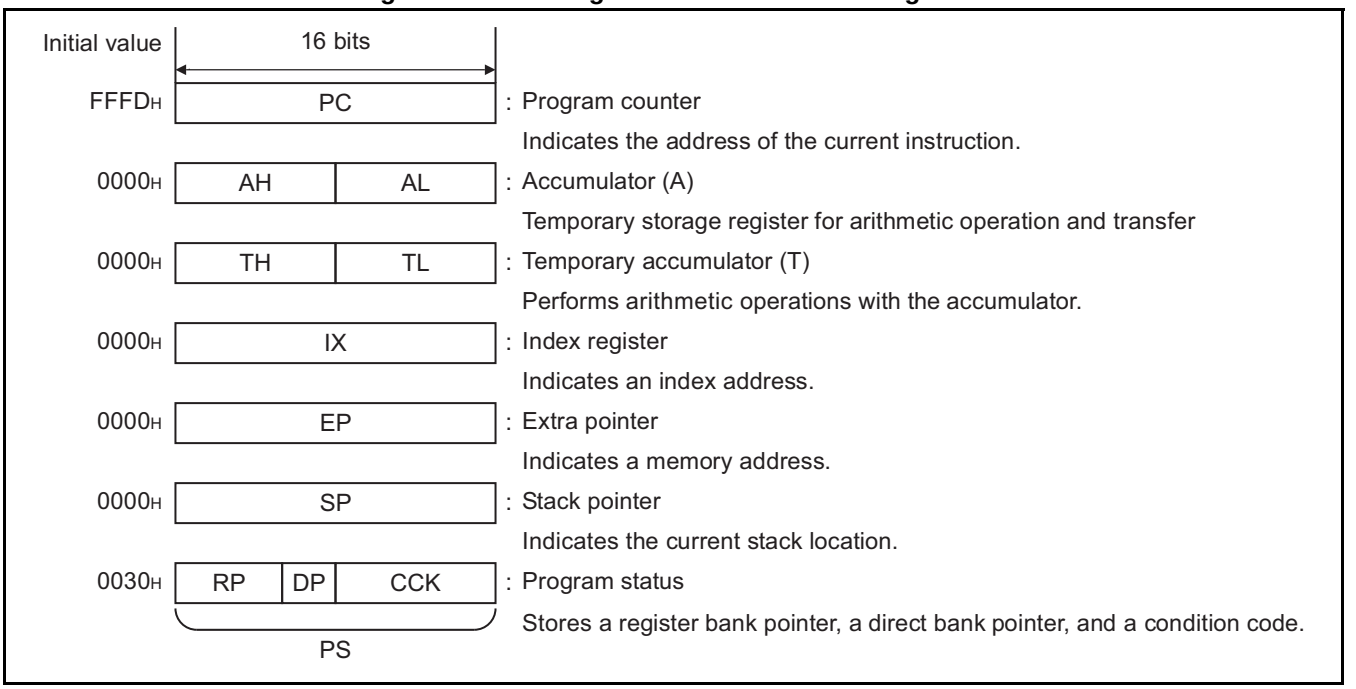
The CPU has dedicated registers: a program counter (PC), two registers for arithmetic operations (A and T), three address pointers (IX, EP, and SP), and the program status (PS) register. Each of the registers is 16 bits long. The PS register consists of the register bank pointer (RP), direct pointer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

■ Configuration of Dedicated Registers

The dedicated registers in the CPU consist of seven 16-bit registers. As for the accumulator (A) and the temporary accumulator (T), using only the lower eight bits of the respective registers is also supported.

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 a 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 set immediately 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 a sub-routine 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 consists of the register bank pointer (RP) and direct bank pointer (DP); the lower eight bits consists of the condition code register (CCR).

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

  The lower eight bits consists of 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 and MOVW PS,A. The register bank pointer (RP) and direct bank pointer (DP) in the program status register can also be read from and written to by accessing the mirror address (0078_H).

  Note that the condition code register (CCR) is a 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.

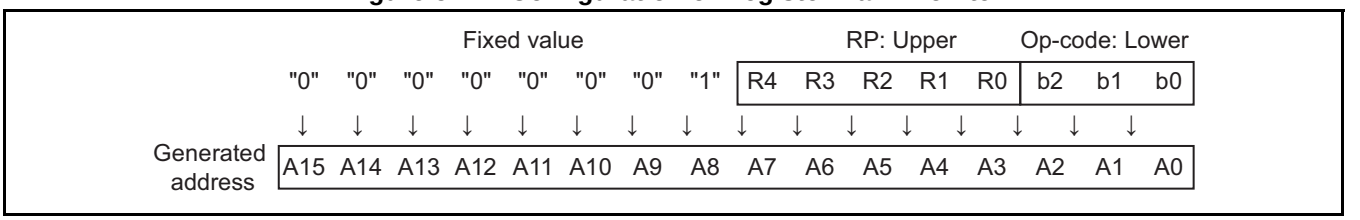
5.1.1 Register Bank Pointer (RP)

The register bank pointer (RP) in bit15 to bit11 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)

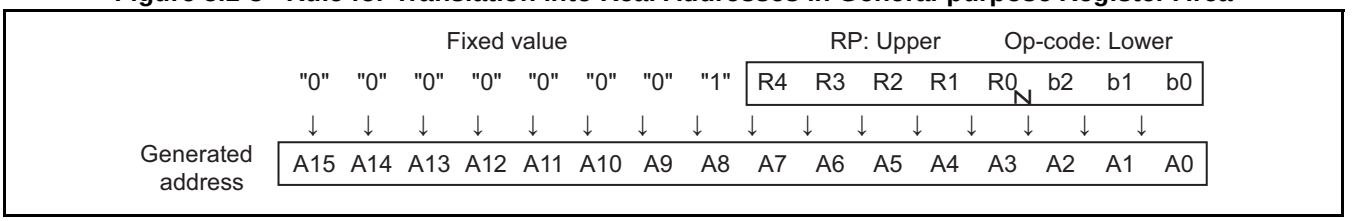
Figure 5.1-2 shows the configuration of the register bank pointer.

Figure 5.1-2 Configuration of Register Bank Pointer



The register bank pointer contains the address of the register bank currently in use. 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



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 "$01FF_H$"(max) to be used as a general-purpose register area. However, certain products have restrictions on the size of the area available for the general-purpose register area. The initial value of the register bank pointer after a reset is "0000_H".

■ Mirror Address for Register Bank and Direct Bank Pointer

Values can be written to the register bank pointer (RP) and the direct bank pointer (DP) by accessing the program status (PS) register with the "MOVW A,PS" instruction; the two pointers can be read by accessing PS with the "MOVW PS,A" instruction. Values can also be directly written to and read from the two pointers by accessing "0078_H", the mirror address of the register bank pointer.

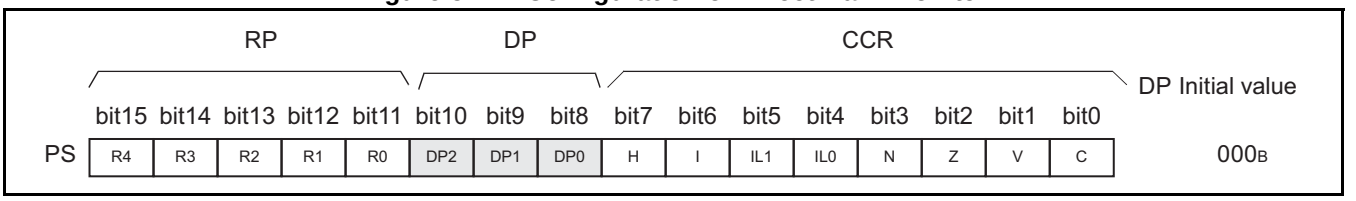
5.1.2 Direct Bank Pointer (DP)

The direct bank pointer (DP) in bit10 to bit8 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



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

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

Table 5.1-1 Direct Bank Pointer and Access Area

Direct bank pointer (DP) [2:0]	Operand-specified dir	Access area
XXX_B (It does not affect mapping.)	0000_H to $007F_H$	0000_H to $007F_H$
000_B (Initial value)	0080_H to $00FF_H$	0080_H to $00FF_H$*1
001_B		0100_H to $017F_H$
010_B		0180_H to $01FF_H$*2
011_B		0200_H to $027F_H$
100_B		0280_H to $02FF_H$*3
101_B		0300_H to $037F_H$
110_B		0380_H to $03FF_H$
111_B		0400_H to $047F_H$

*1: Due to the memory size limit, it is "0090_H to $00FF_H$" in the MB95200H/210H Series.

*2: The available access area is up to "0180_H" in MB95F204H/F204K/F214H/F214K/F203H/F203K/F213H/F213K.

*3: The available access area is up to "0280_H" in MB95F202H/F202K/F212H/F212K.

Table 5.1-2 Direct Address Instruction List

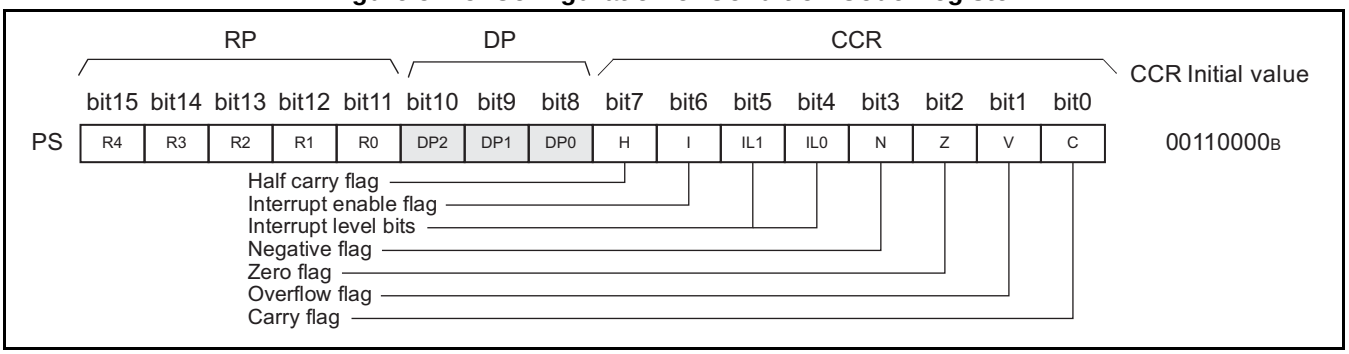
Applicable instructions
CLRB dir:bit
SETB dir:bit
BBC dir:bit,rel
BBS dir:bit,rel
MOV A,dir
CMP A,dir
ADDC A,dir
SUBC A,dir
MOV dir,A
XOR A,dir
AND A,dir
OR A,dir
MOV dir,#imm
CMP dir,#imm
MOVW A,dir
MOVW dir,A

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 Showing Operation Results

● 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 due to 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" due to the result of an operation, and is set to "0" when the value of the most significant bit is "0".

● Zero flag (Z)

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

● Overflow flag (V)

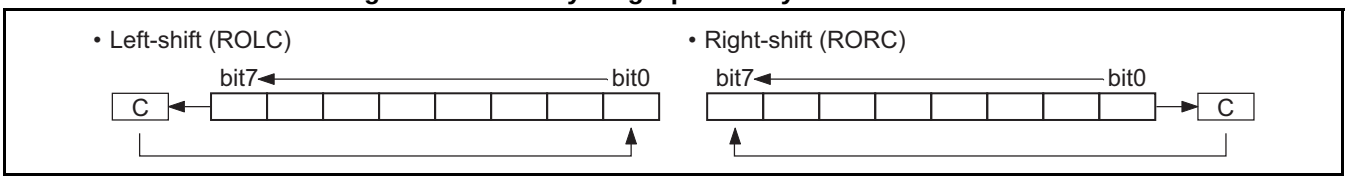
This flag indicates whether the result of an operation has caused an overflow, with the operand used in the operation being regarded as an integer expressed as a complement of two. If an overflow occurs, the overflow flag is set to "1"; otherwise, it is set to "0".

● Carry flag (C)

This flag is set to "1" when a carry from bit7 or a borrow to bit7 occurs due to 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 function.

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" when the CPU does not service an interrupt (with the main program running).

For details of interrupts, see "8.1 Interrupts".

5.2 General-purpose Register

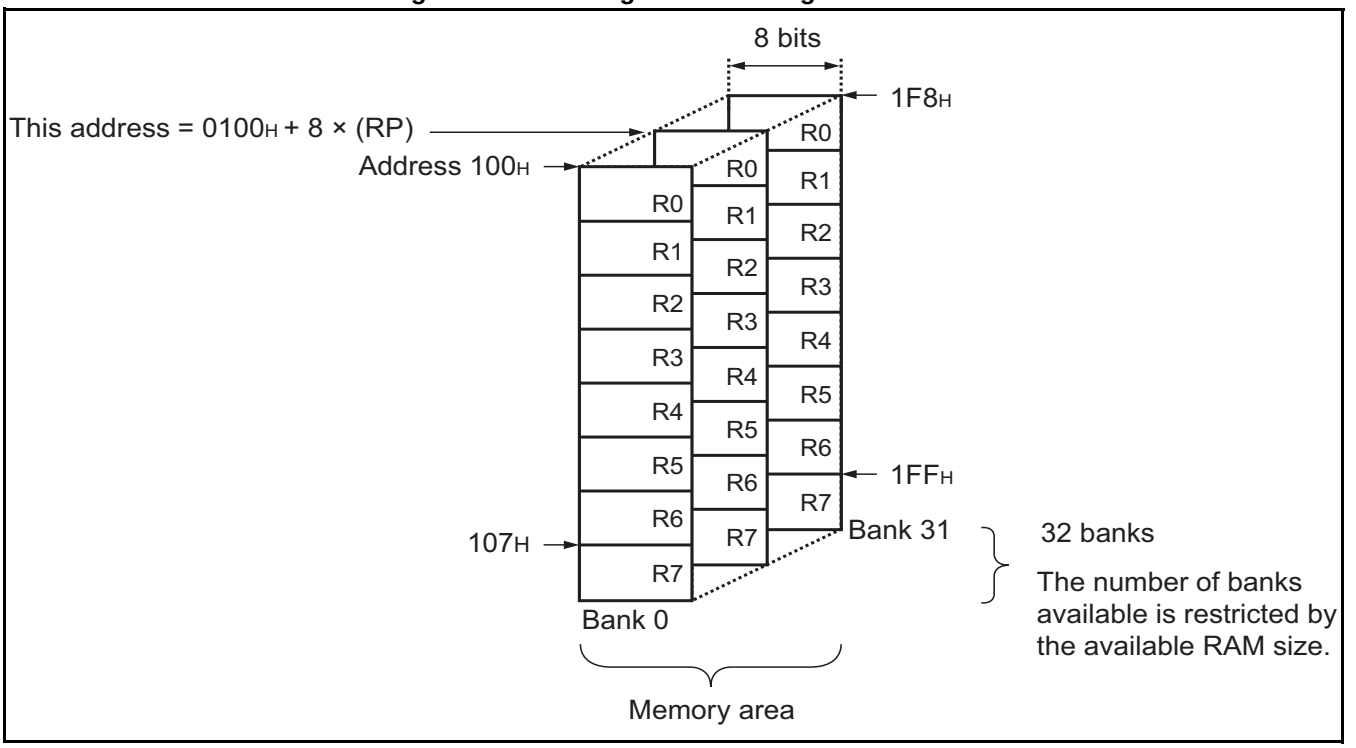
The general-purpose registers are a memory block in which each bank consists of eight 8-bit registers. Up to 32 register banks can be used in total. The register bank pointer (RP) is used to specify a register bank.
Register banks are useful for interrupt handling, vector call processing, and sub-routine calls.

■ Configuration of General-purpose Register

- The general-purpose register is an 8-bit register and is located in a register bank in the general-purpose register area (in RAM).
- Up to 32 banks can be used, each of which 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 the general-purpose register 0 (R0) to the general-purpose register 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 in each model, see "3.1.1 Areas for Specific Applications".

■ Features of General-purpose Registers

The general-purpose register has the following features.

- High-speed access to RAM with short instructions (general-purpose register addressing).
- Grouping registers into a block of register banks facilitates data protection and division of registers in terms of functions.

A general-purpose register bank can be allocated exclusively to an interrupt service routine or a vector call (CALLV #0 to #7) processing routine. For instance, the fourth register bank is always assigned to the second interrupt.

Data of a general-purpose register before an interrupt can be saved to a dedicated register bank by just specifying that register bank at the beginning of an interrupt service routine. This therefore eliminates the need to save data of a general-purpose register in a stack, thereby enabling the CPU to receive interrupts at high speed.

Notes:

In an interrupt service routine, include one of the following in a program to ensure that values of the interrupt level bits (CCR:IL1, IL0) of the condition code register are not modified when modifying a register bank pointer (RP) to specify a register bank.

- Read the interrupt level bits and save their values before writing a value to the RP.
- Directly write a new value to the RP mirror address "0078_H" to update the RP.
- As for the product whose RAM size is 240 B, the area available for general-registers is from "0100_H" to "$017F_H$", which is half of that of the product whose RAM size is 496 B. Therefore, when using a program development tool such as a C compiler to set a general-register area, ensure that the area used as a general-register area does not exceed the size of RAM installed.

5.3 Placement of 16-bit Data in Memory

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

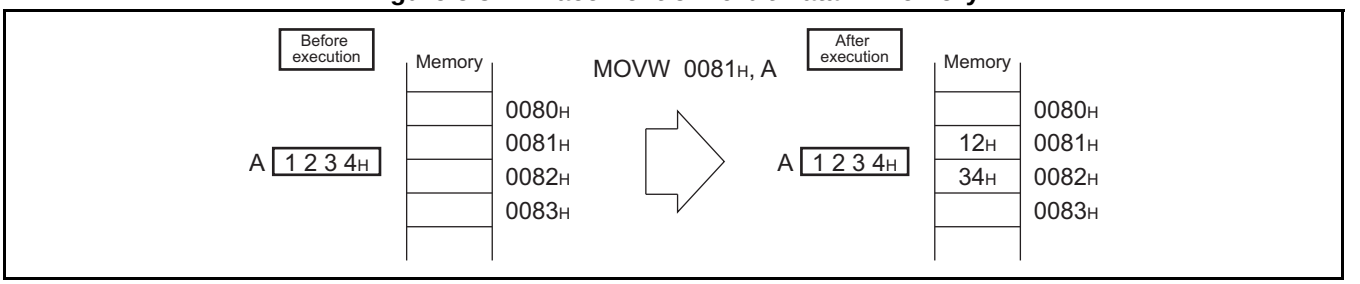
■ Placement of 16-bit Data in Memory

● State of 16-bit data stored in RAM

When 16-bit data is written 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 16-bit data is read, it is handled in the same way.

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

Figure 5.3-1 Placement of 16-bit Data in Memory



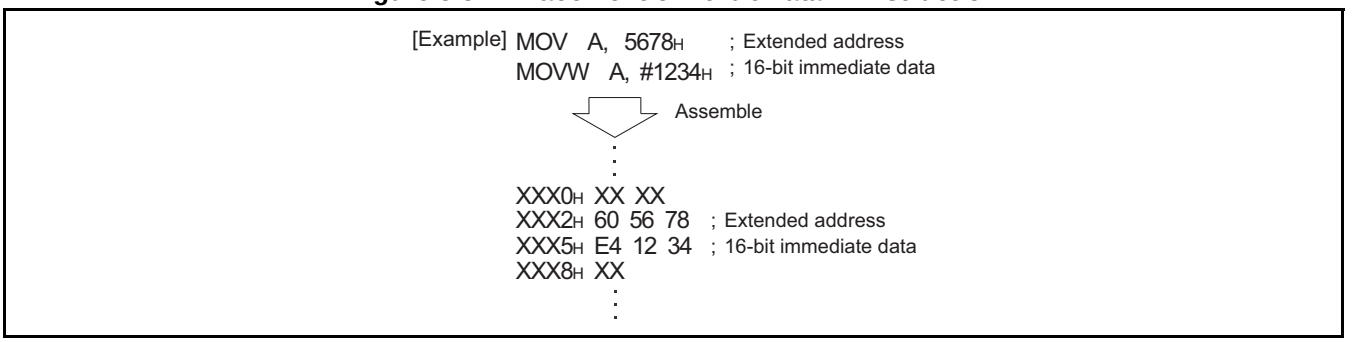
● Storage state of 16-bit data specified by an operand

Even when the operand 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 address next to the one at which the upper byte is stored.

That is true whether an operand is either a memory address or 16-bit immediate data.

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

Figure 5.3-2 Placement of 16-bit Data in Instruction



● Storage state of 16-bit data in the stack

When 16-bit register data is saved in a stack on an interrupt, the upper byte is stored at a lower address in the same way as 16-bit data specified by an operand.

CHAPTER 6

CLOCK CONTROLLER

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

6.1 Overview of Clock Controller

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

■ Overview of Clock Controller

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

The clock controller controls the internal clock according to the clock mode, standby mode settings and the reset operation. The clock mode is used to select an internal operating clock; the standby mode is used to enable and disable clock oscillation and signal supply.

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

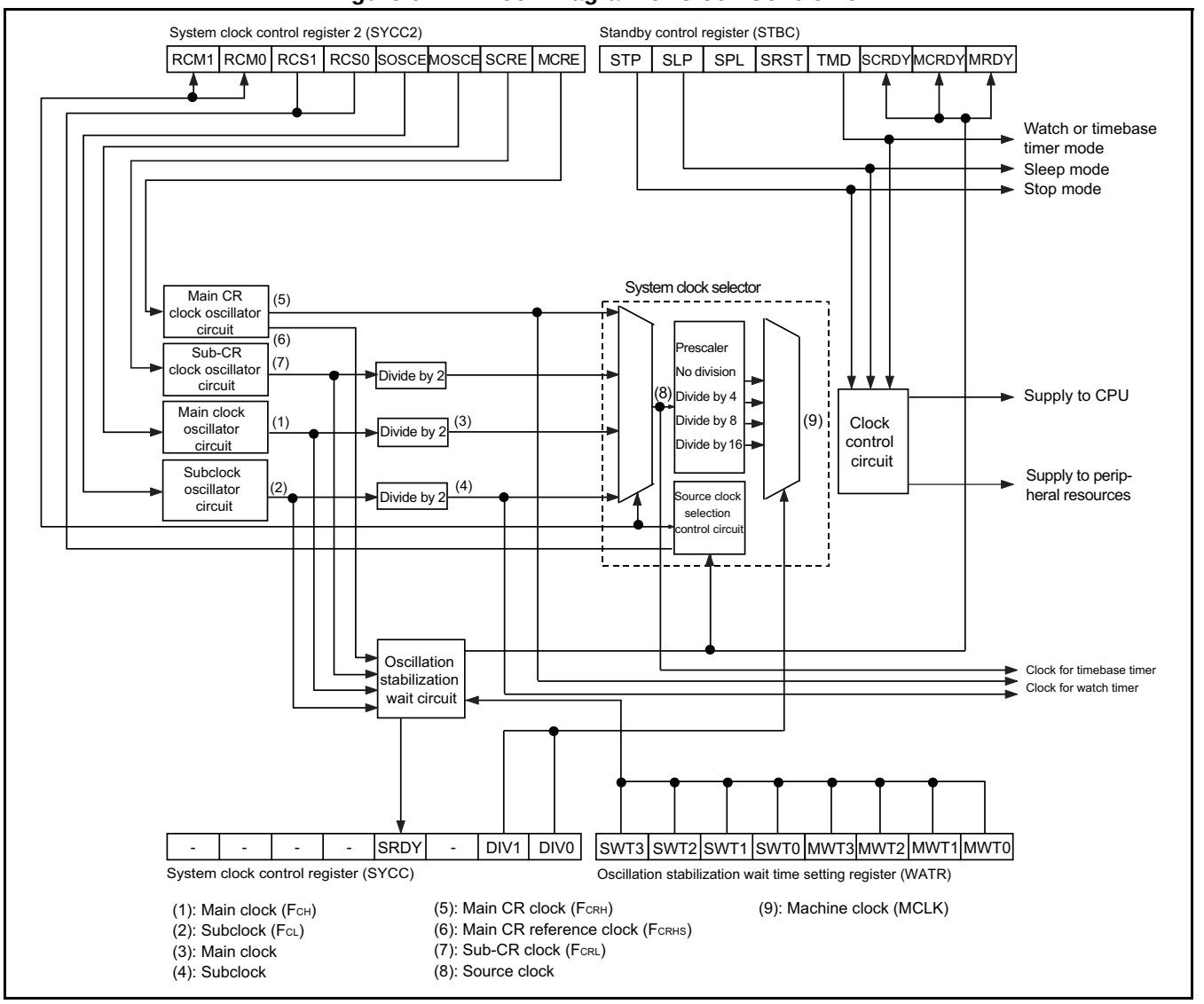
The dual external clock product has four source clocks: a main clock that is the main oscillation clock divided by two; a subclock that is the sub-oscillation clock divided by two; a main CR clock that is the trimmed accurate clock, and a sub-CR clock that is not trimmed by the CR clock divided by two.

A single external clock product has three source clocks: a main clock that is the main oscillation clock divided by two, a main CR clock and a sub-CR clock.

■ Block Diagram of Clock Controller

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

Figure 6.1-1 Block Diagram of Clock Controlller



The clock controller consists of the following blocks:

- Main clock oscillator circuit

  This block is the oscillator circuit for the main clock.

- Subclock oscillator circuit (Dual external clock product)

  This block is the oscillator circuit for the subclock.

- Main CR oscillator circuit

  This block is the oscillator circuit for the main CR clock.

- Sub-CR oscillator circuit

  This block is the oscillator circuit for the sub-CR clock.

- System clock selector

  This block selects a clock according to the clock mode used from the following four types of source clock: main clock, subclock, main CR clock and sub-CR clock. The source clock selected is divided by the prescaler. The divided clock is called "machine clock", which is to be 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 used or oscillation stabilization wait time.

- Oscillation stabilization wait circuit

  This block outputs one of the 14 types of oscillation stabilization signals created by the timebase timer as the oscillation stabilization signal for the main clock, or one of the 15 types of oscillation stabilization signals created by the watch prescaler as the oscillation stabilization wait time signal for the subclock.

- System clock control register (SYCC)

  This register is used to select the machine clock divide ratio.

- 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, timebase timer mode, or watch mode, and the generation of software resets.

- System clock control register 2 (SYCC2)

  This register is used to enable/disable the oscillations of the main clock, main CR clock, subclock, and sub-CR clock, and current clock mode display, clock mode selection.

- Oscillation stabilization wait time setting register (WATR)

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

■ Clock Modes

There are four clock modes: main clock mode, main CR clock mode, subclock mode, and sub-CR clock mode.

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

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 CR clock mode	The machine clock is generated from the main CR clock.
Subclock mode (Dual external clock product only)	The machine clock is generated from the subclock (subclock divided by 2).
Sub-CR clock mode	The machine clock is generated from the sub-CR clock.

In any clock mode, the frequency of a selected clock can be divided. In addition, in a mode in which the main CR clock is used, the clock frequency can also be selected.

■ Peripheral Function not Affected by Clock Mode

The peripheral function listed in the table below is not affected by the clock mode, division, or CR multiplier settings. Table 6.1-2 lists the peripheral function not affected by the clock mode.

Table 6.1-2 Peripheral Function Not Affected by Clock Mode

Peripheral function	Operating clock
Watchdog timer	Main clock (with timebase timer output selected) Subclock (with watch prescaler output selected) (Dual external clock product only)

For some peripheral functions other than the one listed above, the timebase timer or the watch prescaler can be selected as the count clock. Check the description of each peripheral resource for details.

■ Standby Mode

The clock controller selects whether to enable or disable clock oscillation and clock supply to the internal circuitry according to the standby mode selected. With the exception of timebase 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 Mode and Clock Supply States

Standby mode	Clock supply state
Sleep mode	Clock supply to the CPU is stopped. As a result, the CPU stops operating, but other peripheral functions continue operating.
Timebase timer mode	Clock signals are only supplied to the timebase timer and the watch prescaler, while the clock supply to other circuits is stopped. As a result, all the functions other than the timebase timer, watch prescaler, external interrupt, and low-voltage detection reset (option) are stopped. The timebase timer mode is the only standby mode that can be used in main clock mode and main CR clock mode.
Watch mode (Dual external clock product only)	Main clock oscillation is stopped. Clock signals are supplied only to the watch prescaler, while clock supply to other circuits is stopped. As a result, all the functions other than the watch prescaler, external interrupt, and low-voltage detection reset (option) are stopped. The watch mode is the only standby mode that can be used in subclock mode and sub-CR clock mode.
Stop mode	Main clock oscillation and subclock oscillation are stopped, and clock supply to all circuits is stopped. 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 the respective operating states of different internal circuits with different combinations of clock mode and standby mode.

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

Function	RUN				Sleep				Timebase timer		Watch prescaler		Stop	
	Main clock mode	Main CR clock mode	Sub-clock mode (Dual external clock product)	Sub-CR clock mode	Main clock mode	Main CR clock mode	Sub-clock mode (Dual external clock product)	Sub-CR clock mode	Main clock mode	Main CR clock mode	Sub-clock mode (Dual external clock product)	Sub-CR clock mode	Main CR clock mode	Sub-CR clock mode
Main clock	Operating	Stopped*1	Stopped		Operating	Stopped*1	Stopped		Operating	Stopped*1	Stopped		Stopped	
Main CR clock	Stopped*2	Operating	Stopped		Stopped*2	Operating	Stopped		Stopped*2	Operating	Stopped		Stopped	
Subclock	Operating*3		Operating		Operating*3		Operating		Operating*3		Operating		Operating*3	Stopped
Sub-CR clock	Stopped*4		Stopped*4	Operating	Stopped*4		Stopped*4	Operating	Stopped*4		Stopped*4	Operating	Stopped*4	Stopped
CPU	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	
ROM	Operating		Operating		Value held		Value held		Value held		Value held		Value held	
RAM														
I/O ports	Operating		Operating		Output held		Output held		Output held		Output held		Output held/Hi-Z	
Timebase timer	Operating		Stopped		Operating		Stopped		Operating		Stopped		Stopped	
Watch prescaler	Operating*3		Operating		Operating*3		Operating		Operating*3		Operating		Operating*3	Stopped
External interrupt	Operating		Operating		Operating		Operating		Operating		Operating		Operating	
Hardware watchdog timer	Operating		Operating		Operating*5		Operating*5		Operating*5		Operating*5		Operating*5	
Software watchdog timer	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	
Low-voltage detection reset	Operating		Operating		Operating		Operating		Operating		Operating		Operating	
Other peripheral functions	Operating		Operating		Operating		Operating		Stopped		Stopped		Stopped	

*1: The main clock operates when the main clock oscillation enable bit in the system clock control register 2 (SYCC2:MOSCE) is set to "1".

*2: The main CR clock operates when main CR clock oscillation enable bit in the system clock control register 2 (SYCC2:MCRE) is set to "1".

*3: The module stops when the subclock oscillation enable bit in the system clock control register 2 (SYCC2:SOSCE) is set to "0".

*4: The module operates when the sub-CR clock oscillation enable bit in the system clock control register 2 (SYCC2:SCRE) is set to "1".

*5: The hardware watchdog timer stops when the hardware watchdog timer is disabled by the non-volatile register in standby mode.

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 after the start of oscillation by counting a specific number of oscillation clock cycles. During the oscillation stabilization wait time, the clock controller stops clock supply to internal circuits.

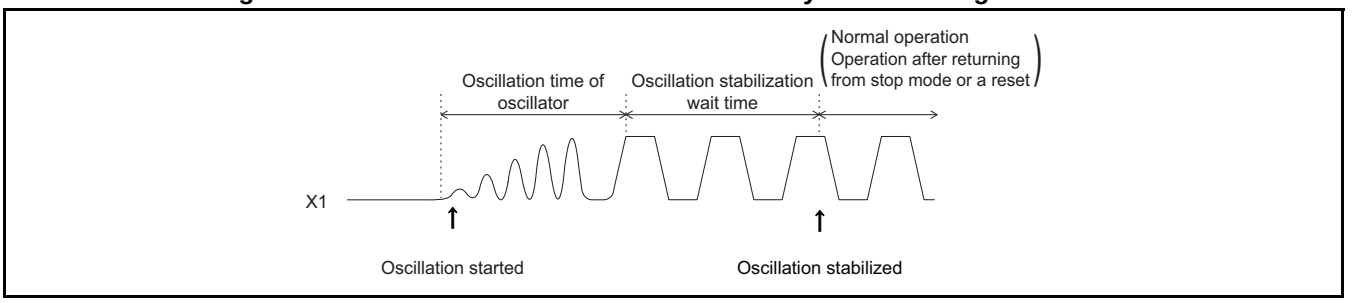
■ Oscillation Stabilization Wait Time

The clock controller obtains the oscillation stabilization wait time after the start of oscillation by counting a specific number of oscillation clock cycles. During the oscillation stabilization wait time, the clock controller stops clock supply to internal circuits.

When the power is switched on, or when a state transition request making the oscillator start from the oscillation stop state is generated due to a change of clock mode caused by a reset, by an interrupt in standby mode or by the software operation, the clock controller automatically waits for the oscillation stabilization wait time of the main clock or of the subclock to elapse before making the clock mode transit to another mode.

Figure 6.2-1 shows how the oscillator operates immediately after starting oscillating.

Figure 6.2-1 Behavior of Oscillator Immediately after Starting Oscillation



Oscillation stabilization wait time of main clock, subclock, main CR clock, sub-CR clock is counted by using a dedicated counter. The count value 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 at the initial value.

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

Table 6.2-1 Oscillation Stabilization Wait Time

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

After the oscillation stabilization wait time of the main clock ends, the measurement of the oscillation

stabilization wait time of the subclock is started.

■ CR Clock Oscillation Stabilization Wait Time

As with the oscillation stabilization wait time of the oscillator, when a state transition request making CR oscillation start from the CR oscillation stop state is generated due to a change of clock mode caused by an interrupt in standby mode or by the software operation, the clock controller automatically waits for the CP oscillation stabilization wait time to elapse.

The CR clock oscillation stabilization wait time changes according to the CR start time.

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

Table 6.2-2 CR Oscillation Stabilization Wait Time

	CR oscillation stabilization wait time
Main CR clock	$2^8/F_{CRHS}$
Sub-CR clock	$2^4/F_{CRL}$

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

If state transition occurs, the clock controller automatically waits for the oscillation stabilization wait time to elapse whenever necessary. Depending on the circumstances under which state transition occurs, the clock controller does not wait for the oscillation stabilization wait time to elapse even if state transition occurs.

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

6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) is used to select the machine clock divide ratio, and indicates the condition of subclock oscillation stabilization.

■ 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
0007H	-	-	-	-	SRDY	-	DIV1	DIV0	0000X011B
	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R0/WX	R/W	R/W	

RCS1	RCS0	Machine clock divide ratio select bits
0	0	Source clock
0	1	Source clock/4
1	0	Source clock/8
1	1	Source clock/16

SRDY	Subclock oscillation stabilization bit
0	Indicates the subclock oscillation stabilization wait state or subclock oscillation has been stopped.
1	Indicates subclock oscillation has become stable.

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R0WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R/W : Readable/writable (The read value is the same as the write value.)
X : Indeterminate
: 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 to bit4, bit2</td><td>Undefined bits</td><td>These bits are undefined.
• When these bits are read, they always return "0".
• These bits are read-only. Writing values to them has no effect on operation.</td></tr>
<tr><td>bit3</td><td>SRDY:
Subclock oscillation stabilization bit
(Dual external clock product only)</td><td>This bit indicates whether subclock oscillation has become stable.
• When the SDRY bit is set to "1", that indicates the oscillation stabilization wait time for the subclock has elapsed.
• When the SRDY bit is set to "0", that indicates that the clock controller is in the subclock oscillation stabilization wait state or that subclock oscillation has been stopped.
This bit is read-only. Writing data to it has no effect on operation.
On a single external clock product, the value of the bit is meaningless</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 for the source clock.
• The machine clock is generated from the source clock according to the divide ratio set by these bits.
<table>
<tr><th>DIV1</th><th>DIV0</th><th>Machine clock divide ratio select 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 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 6.4-1 Configuration of Oscillation Stabilization Wait Time Setting Register (WATR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0005H	SWT3	SW72	SWT1	SWT0	MWT3	MWT2	MWT1	MWT0	11111111B
	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 2.05 ms
1	1	0	1	$2^{12} - 2$	$(2^{12} - 2)/F_{CH}$	About 1.02 ms
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 1.00 s
1	1	1	0	$2^{14} - 2$	$(2^{14} - 2)/F_{CL}$	About 0.5 s
1	1	0	1	$2^{13} - 2$	$(2^{13} - 2)/F_{CL}$	About 0.25 s
1	1	0	0	$2^{12} - 2$	$(2^{12} - 2)/F_{CL}$	About 0.125 s
1	0	1	1	$2^{11} - 2$	$(2^{11} - 2)/F_{CL}$	About 62.44 ms
1	0	1	0	$2^{10} - 2$	$(2^{10} - 2)/F_{CL}$	About 31.19 ms
1	0	0	1	$2^{9} - 2$	$(2^{9} - 2)/F_{CL}$	About 15.56 ms
1	0	0	0	$2^{8} - 2$	$(2^{8} - 2)/F_{CL}$	About 7.75 ms
0	1	1	1	$2^{7} - 2$	$(2^{7} - 2)/F_{CL}$	About 3.85 ms
0	1	1	0	$2^{6} - 2$	$(2^{6} - 2)/F_{CL}$	About 1.89 ms
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 (The read value is the same as the write value.)
: Initial value (For mask ROM products, the customer can specify the initial value when ordering ROM.)

Table 6.4-1 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: Subclock oscillation stabilization wait time select bits</td><td>These bits set the subclock oscillation stabilization wait time.
<table>
<tr><th>SWT3, SWT2, SWT1, SWT0</th><th>Number of cycles</th><th colspan="2">Subclock F_{CL} = 32.768 kHz</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 915.5 µs</td></tr>
<tr><td>0100_B</td><td>2^{4}-2</td><td>$(2^{4}-2)/F_{CL}$</td><td>About 427.2 µs</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 a single external clock product, the value of these bits is meaningless.
The number of cycles in the above table is the minimum subclock oscillation stabilization wait time.
The maximum value is the number of cycles in the above table plus $1/F_{CL}$.
Note: Do not modify these bits during subclock oscillation stabilization wait time. Modify them either when the subclock oscillation stabilization bit in the system clock control register (SYCC:SRDY) has been set to "1", or in main clock mode, main CR clock mode or sub-CR clock mode. These bits can also be modified when the subclock is stopped with the subclock oscillation stop bit in the system clock control register 2 (SYCC2:SOSCE) set to "0" in main clock mode, main CR clock mode or sub-CR clock mode.</td></tr>
</table>

Table 6.4-1 Functions of Bits in Oscillation Stabilization Wait Time Setting Register (WATR) (2 / 2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit3
to
bit0</td><td>MWT3, MWT2, MWT1, MWT0:
Main clock oscillation stabilization wait time select bits</td><td>These bits set the main clock oscillation stabilization wait time.
<table>
<tr><th>MWT3, MWT2, MWT1, MWT0</th><th>Number of cycles</th><th colspan="2">Main clock F_{CH} = 4 MHz</th></tr>
<tr><td>1111_B</td><td>2^{14}-2</td><td>$(2^{14}-2)/F_{CH}$</td><td>About 4.10 ms</td></tr>
<tr><td>1110_B</td><td>2^{13}-2</td><td>$(2^{13}-2)/F_{CH}$</td><td>About 2.05 ms</td></tr>
<tr><td>1101_B</td><td>2^{12}-2</td><td>$(2^{12}-2)/F_{CH}$</td><td>About 1.02 ms</td></tr>
<tr><td>1100_B</td><td>2^{11}-2</td><td>$(2^{11}-2)/F_{CH}$</td><td>511.5 µs</td></tr>
<tr><td>1011_B</td><td>2^{10}-2</td><td>$(2^{10}-2)/F_{CH}$</td><td>255.5 µs</td></tr>
<tr><td>1010_B</td><td>2^{9}-2</td><td>$(2^{9}-2)/F_{CH}$</td><td>127.5 µs</td></tr>
<tr><td>1001_B</td><td>2^{8}-2</td><td>$(2^{8}-2)/F_{CH}$</td><td>63.5 µs</td></tr>
<tr><td>1000_B</td><td>2^{7}-2</td><td>$(2^{7}-2)/F_{CH}$</td><td>31.5 µs</td></tr>
<tr><td>0111_B</td><td>2^{6}-2</td><td>$(2^{6}-2)/F_{CH}$</td><td>15.5 µs</td></tr>
<tr><td>0110_B</td><td>2^{5}-2</td><td>$(2^{5}-2)/F_{CH}$</td><td>7.5 µs</td></tr>
<tr><td>0101_B</td><td>2^{4}-2</td><td>$(2^{4}-2)/F_{CH}$</td><td>3.5 µs</td></tr>
<tr><td>0100_B</td><td>2^{3}-2</td><td>$(2^{3}-2)/F_{CH}$</td><td>1.5 µs</td></tr>
<tr><td>0011_B</td><td>2^{2}-2</td><td>$(2^{2}-2)/F_{CH}$</td><td>0.5 µs</td></tr>
<tr><td>0010_B</td><td>2^{1}-2</td><td>$(2^{1}-2)/F_{CH}$</td><td>0.0 µs</td></tr>
<tr><td>0001_B</td><td>2^{1}-2</td><td>$(2^{1}-2)/F_{CH}$</td><td>0.0 µs</td></tr>
<tr><td>0000_B</td><td>2^{1}-2</td><td>$(2^{1}-2)/F_{CH}$</td><td>0.0 µs</td></tr>
</table>
The number of cycles in the above table is the minimum main clock oscillation stabilization wait time. The maximum value is the number of cycles in the above table plus $1/F_{CH}$.
Note: Do not modify these bits during main clock oscillation stabilization wait time. Modify them either when the main clock oscillation stabilization bit in the system clock control register (SYCC:MRDY) has been set to "1", or in main CR clock mode, subclock mode or sub-CR clock mode. These bits can also be modified when the main clock is stopped with the main clock oscillation stop bit in the system clock control register 2 (SYCC2:MOSCE) set to "0" in main CR clock mode, subclock mode or sub-CR clock mode.</td></tr>
</table>

6.5 Standby Control Register (STBC)

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

■ Standby Control Register (STBC)

Figure 6.5-1 Standby Control Register (STBC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0008H	STP	SLP	SPL	SRST	TMD	SCRDY	MCRDY	MRDY	00000XXXB
	R0,W	R0,W	R/W	R0,W	R0,W	R/WX	R/WX	R/WX	

MRDY	Main clock oscillation stabilization bit
0	Indicates main clock oscillation stabilization wait state or main clock oscillation has been stopped.
1	Indicates main clock oscillation has become stable.

MCRDY	Main CR clock oscillation stabilization bit
0	Indicates main CR clock oscillation stabilization wait state or main CR clock oscillation has been stopped.
1	Indicates main CR clock oscillation has become stable.

SCRDY	Sub-CR clock oscillation stabilization bit
0	Indicates sub-CR clock oscillation stabilization wait state or sub-CR clock oscillation has been stopped.
1	Indicates sub-CR clock oscillation has become stable.

TMD	Watch bit		
	Read	Write	
0	"0" is always read.	Has no effect on operation.	
1	-	Main clock mode/ Main CR clock mode	Subclock mode/Sub-CR clock mode
		Causes transition to timebase timer mode	Causes transition to watch mode

SRST	Software reset bit	
	Read	Write
0	"0" is always read.	Has no effect on 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, timebase timer mode, or watch mode.
1	Places external pins in a high impedance state in stop mode, timebase timer mode, or watch mode.

SLP	Sleep bit	
	Read	Write
0	"0" is always read.	Has no effect on operation
1	-	Causes transition to sleep mode

STP	Stop bit	
	Read	Write
0	"0" is always read.	Has no effect on operation
1	-	Causes transition to stop mode

R0,W : Write only (Writable. "0" is read.)
R/W : Readable/writable (The read value is the same as the write value.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
- : Undefined
X : Indeterminate
: Initial value

Table 6.5-1 Functions of Bits in Standby Control Register (STBC)

Bit name		Function
bit7	STP: Stop bit	This bit sets the transition to stop mode. **Writing "0":** this bit is meaningless. **Writing "1":** causes the device to transit to stop mode. When this bit is read, it always returns "0". Note: After an interrupt request is issued, writing "1" to this bit is ignored. For details, see "6.8.1 Notes on Using Standby Mode".
bit6	SLP: Sleep bit	This bit sets the transition to sleep mode. **Writing "0":** this bit is meaningless. **Writing "1":** causes the device to transit to sleep mode. When this bit is read, it always returns "0". Note: After an interrupt request is issued, writing "1" to this bit is ignored. For details, see "6.8.1 Notes on Using Standby Mode".
bit5	SPL: Pin state setting bit	This bit sets the states of external pins in stop mode, timebase timer mode, and watch mode. **Writing "0":** the state (level) of an external pin is kept in stop mode, timebase timer mode and watch mode. **Writing "1":** an external pin becomes high impedance in stop mode, timebase timer mode and watch mode. (A pin for which connection to a pull-up resistor has been selected in the pull-up setting register is pulled up.)
bit4	SRST: Software reset bit	This bit sets a software reset. **Writing "0":** has no effect on operation. **Writing "1":** generates a 3-machine clock reset signal. When this bit is read, it always returns "0".
bit3	TMD: Watch bit	On a dual external clock product, this bit sets transition to timebase timer mode or watch mode. On a single external clock product, the bit sets transition to timebase timer mode. • Writing "1" to this bit in main clock mode or main CR clock mode causes the device to transit to timebase timer mode. • Writing "1" to this bit in subclock mode or sub-CR clock mode causes the device to transit to watch mode. • Writing "0" to this bit has no effect on operation. • When this bit is read, it always returns "0". Note: After an interrupt request is issued, writing "1" to this bit is ignored. For details, see "6.8.1 Notes on Using Standby Mode".
bit2	SCRDY: Sub-CR clock oscillation stabilization bit	This bit indicates whether sub-CR clock oscillation has become stable. • When the SCDRY bit is set to "1", that indicates the oscillation stabilization wait time for the sub-CR clock has elapsed • When the SCRDY bit is set to "0", that indicates that the clock controller is in the sub-CR clock oscillation stabilization wait state or that sub-CR clock oscillation has been stopped. This bit is read-only. Writing a value to it has no effect on operation. On a single external clock product, the value of this bit is meaningless.
bit1	MCRDY: Main CR clock oscillation stabilization bit	This bit indicates whether main CR clock oscillation has become stable. • When the MCDRY bit is set to "1", that indicates the oscillation stabilization wait time for the main CR clock has elapsed. • When the MCDRY bit is set to "0", that indicates that the clock controller in the main CR clock oscillation stabilization wait state or that main CR clock stabilization has been stopped. This bit is read-only. Writing a value to it has no effect on operation.
bit0	MRDY: Main clock oscillation stabilization bit	This bit indicates whether main clock oscillation has become stable. • When the MRDY bit is set to "1", that indicates that the oscillation stabilization wait time for the main clock has elapsed. • When the MRDY bit is set to "0", that indicates that the clock controller is in the main clock oscillation stabilization wait state or that main clock oscillation has been stopped. This bit is read-only. Writing a value to it 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 (SYCC2:RCM1,RCM0) and clock mode setting bits (SYCC2:RCS1,RCS0) in the system clock control register 2.
- If two or more of the following bits, stop bit (STP), sleep bit (SLP), software reset bit (SRST) and watch bit (TMD), are set to "1" together, the order of priority for such bits is as follows:

  (1) Software reset bit (SRST)

  (2) Stop bit (STP)

  (3) Watch bit (TMD)

  (4) Sleep bit (SLP)

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

6.6 System Clock Control Register 2 (SYCC2)

The system clock control register 2 (SYCC2) is used to indicate the current clock mode and switch the clock mode, and control subclock, sub-CR clock, main clock, main CR clock oscillations.

■ Configuration of System Clock Control Register 2 (SYCC2)

Figure 6.6-1 Configuration of System Clock Control Register 2 (SYCC2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000DH	RCMI	RCM0	RCS1	RCS0	SOSCE	MOSCE	SCRE	MCRE	10100011B
	R/WX	R/WX	R/W	R/W	R/W	R/W	R/W	R/W	

MCRE	Main CR clock oscillation enable bit
0	Disables main CR clock oscillation
1	Enables main CR clock oscillation

SCRE	Sub-CR clock oscillation enable bit
0	Disables sub-CR clock oscillation
1	Enables sub-CR clock oscillation

MOSCE	Main clock oscillation enable bit
0	Disables main clock oscillation
1	Enables main clock oscillation

SOSCE	Subclock oscillation enable bit
0	Disables subclock oscillation
1	Enables subclock oscillation

RCS1	RCS0	Clock mode select bits
0	0	Sub-CR clock mode
0	1	Subclock mode
1	0	Main CR clock mode
1	1	Main clock mode

RCM1	RCM0	Clock mode monitor bits
0	0	Sub-CR clock mode
0	1	Subclock mode
1	0	Main CR clock mode
1	1	Main clock mode

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R/W : Readable/writable (The read value is the same as the write value.)
X : Indeterminate
: Initial value

Table 6.6-1 Functions of Bits in System Clock Control Register 2 (SYCC2)

Bit name		Function
bit7,bit6	RCM1, RCM0: Clock mode monitor bits	These bits indicate the current clock mode. "**00_B**": indicate sub-CR clock mode. "**01_B**": indicate subclock mode. "**10_B**": indicate main CR clock mode. "**11_B**": indicate main clock mode. These bits are read-only. Writing values to them has no effect on operation.
bit5,bit4	RCS1, RCS0: Clock mode select bits	These bits specify the current clock mode. **Writing "00_B"**: transition to sub-CR clock mode **Writing "01_B"**: transition to subclock mode **Writing "10_B"**: transition to main CR clock mode **Writing "11_B"**: transition to main clock mode • If main clock oscillation has been disabled by the system clock control register 2, writing "11_B" to these bits is ignored, and their values remain unchanged. • If subclock oscillation has been disabled by the system clock control register 2, writing "01_B" to these bits is ignored, and their values remain unchanged.
bit3	SOSCE: Subclock oscillation enable bit	This bit enables/disables the subclock. **Writing "0"**: disables subclock oscillation. **Writing "1"**: enables subclock oscillation. • If the RCS bits are set to "01_B", this bit is set to "1". • If the RCS or RCM bits are "01_B", writing "0" to this bit is ignored, and its value remains unchanged. • If subclock oscillation has been disabled by the system clock control register 2, writing "1" to this bit is ignored, and its value remains unchanged.
bit2	MOSCE: Main clock oscillation enable bit	This bit enables/disables the main clock. **Writing "0"**: disables main clock oscillation. **Writing "1"**: enables main clock oscillation. • If the RCS bits are set to "11_B", this bit is set to "1". • If the RCS or RCM bits are "11_B", writing "0" to this bit is ignored, and its value remains unchanged. • When the RCM bits are modified to other values from "11_B", this bit is set to "0". • If the RCM1 bit is "0", writing "1" to this bit is ignored. • If main clock oscillation has been disabled by the system clock control register 2, writing "11_B" to these bits is ignored, and their values remain unchanged.
bit1	SCRE: Sub-CR clock oscillation enable bit	This bit enables/disables the sub-CR clock. **Writing "0"**: disables sub-CR clock oscillation. **Writing "1"**: enables sub-CR clock oscillation. • If the RCS bits are set to "01_B", this bit is set to "1". • If the RCS or RCM bits are "01_B", writing "0" to this bit is ignored, and its value remains unchanged. • If the hardware watchdog timer is used, this bit is set to "1".
bit0	MCRE: Main CR clock oscillation enable bit	This bit enables/disables the main CR clock. **Writing "0"**: disables main CR clock oscillation. **Writing "1"**: enables main CR clock oscillation. • If the RCS bits are set to "10_B", the bit is set to "1". • If the RCS or RCM bits are "10_B", writing "0" to this bit is ignored, and its value remains unchanged. • When the RCM bits are modified to other values from "10_B", the bit is set to "0". • If the RCM1 bit is "0", writing "1" to this bit is ignored.

6.7 Clock Modes

There are four clock modes: main clock mode, subclock mode, main CR clock mode, and sub-CR clock mode. Mode switching occurs according to the settings in the system clock control register 2 (SYCC2).

■ Operations in Main Clock Mode

In main clock mode, main clock is used as the machine clock for the CPU and peripheral functions.

The timebase timer operates with the main clock.

The watch prescaler operates with the subclock (on the dual external clock product).

While the device is operating in main clock mode, it can be set to transit to one of the following standby mode: sleep mode, stop mode, or timebase timer mode.

After a reset, the device always enters main CR clock mode regardless of the clock mode used before that reset.

■ Operations in Subclock Mode (on Dual External Clock Product)

In subclock mode, main clock oscillation[*1] is stopped and the subclock is used as the machine clock for the CPU and peripheral functions. In this mode, the timebase timer stops as it requires the main clock for operation.

While the device is operating in subclock mode, it can be set to transit to one of the following standby mode: sleep mode, stop mode, or watch mode.

■ Operations in Main CR Clock Mode

In main CR clock mode, the main CR clock is used as the machine clock for the CPU and peripheral functions. The timebase timer and the watchdog timer operate with the main clock.

The watch prescaler operates with the subclock (on the dual external clock product).

While the device is operating in main CR clock mode, it can be set to transit to one of the following standby mode: sleep mode, stop mode, or timebase timer mode.

■ Operations in Sub-CR Clock Mode (on Dual External Clock Product)

In sub-CR clock mode, main clock oscillation[1] is stopped and the sub-CR clock is used as the machine clock for the CPU and peripheral functions. In this mode, the timebase timer stops as it requires the main clock for operation. The watch prescaler and the watch counter operate with the subclock.

While the device is operating in sub-CR clock mode, it can be set to transit to one of the following standby mode: sleep mode, stop mode, or watch mode.

Note:

1. The main clock and the main CR clock are automatically disabled (SYCC2:MOSCE is set to "0" or SYCC2:MCRE is set to "0") when the clock mode transits from main clock mode or main CR clock mode to another clock mode. If the new clock mode is subclock mode or sub-CR clock mode, the main CR clock cannot be enabled.

■ Clock Mode State Transition Diagram

There are four clock mode: main clock mode, subclock mode, main CR clock mode and sub-CR clock mode. The device can switch between these modes according to the settings in the system clock control register 2 (SYCC2).

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

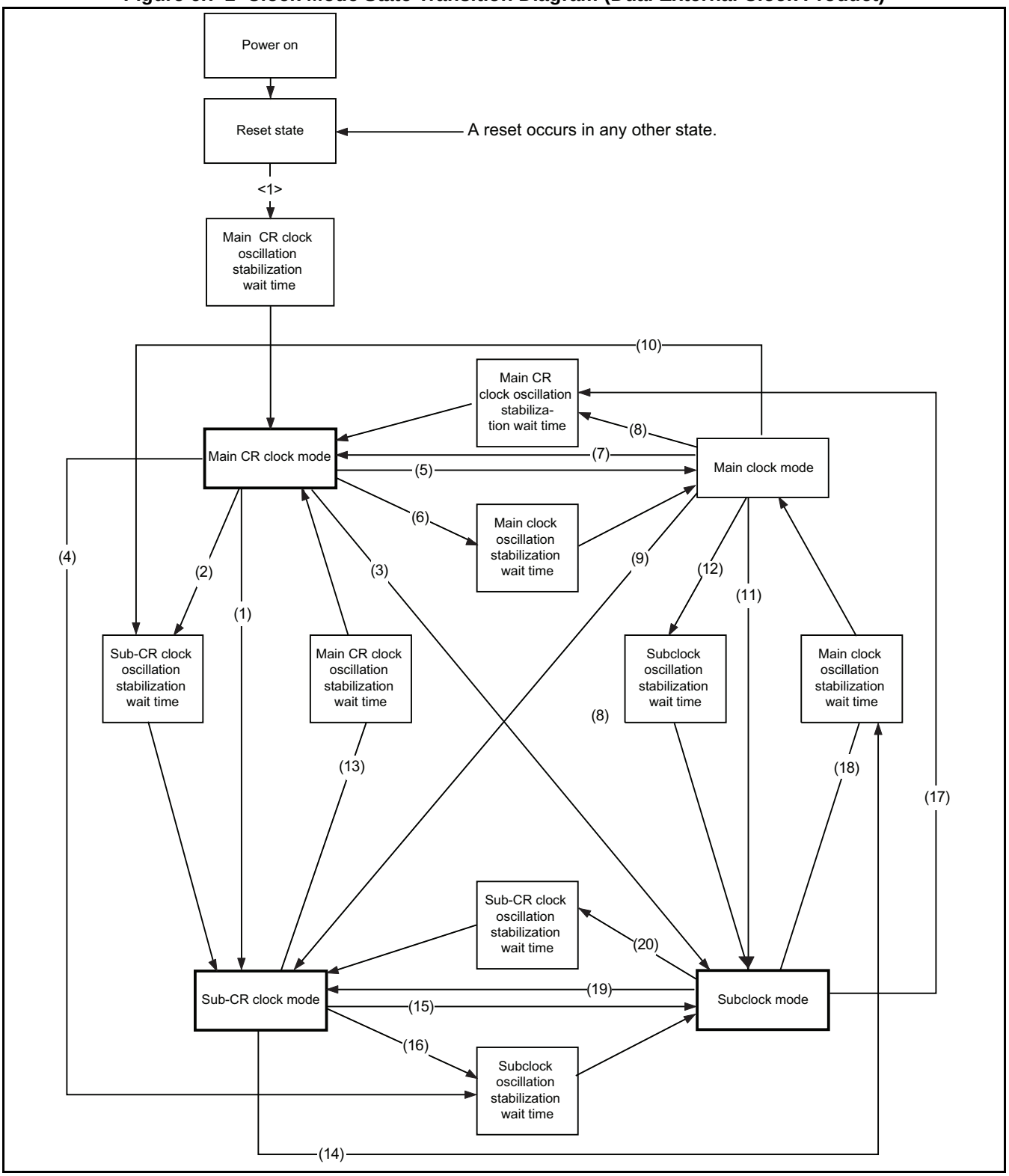


Figure 6.7-2 Clock Mode State Transition Diagram (Single External Clock Product)

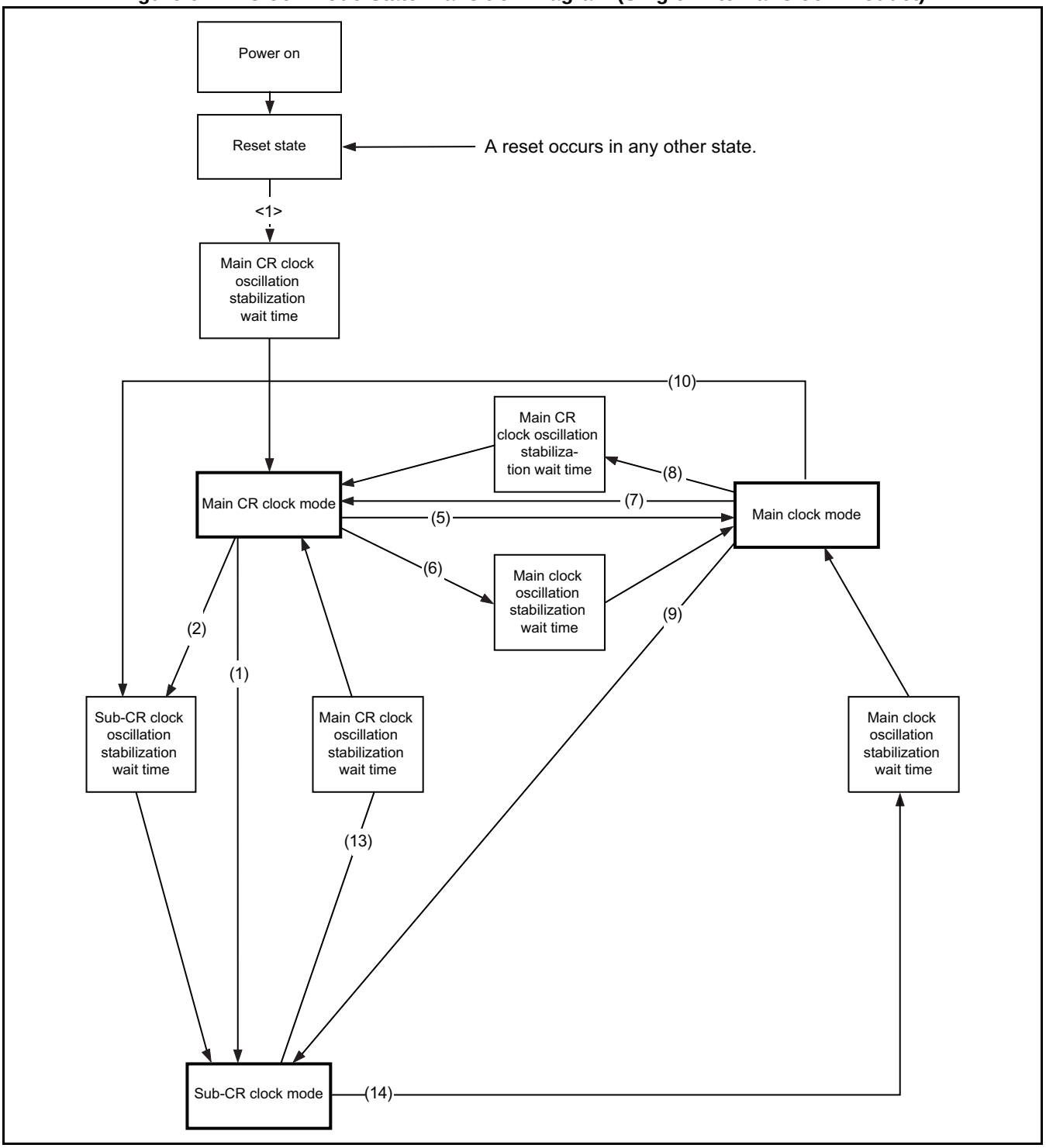


Table 6.7-1 Clock Mode State Transition Table (1 / 2)

	Current State	Next State	Description
<1>	Reset state	Main CR clock	After a reset, the device waits for the main CR clock oscillation stabilization wait time to elapse and transits to main CR clock mode. Even if that reset is a watchdog reset, software reset or external reset caused in any clock mode, the device waits for the main CR clock oscillation stabilization wait time to elapse.
(1)	Main CR clock	Sub-CR clock	The device transits to sub-CR clock mode when the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "00_B". However, if the sub-CR has been stopped according to the setting of the sub-CR clock oscillation enable bit in the system clock control register 2 (SYCC2:SCRE), the device waits for the sub-CR clock oscillation stabilization wait time to elapse before transiting to subclock mode. In other words, if the sub-CR clock oscillation is enabled in advance and the sub-CR clock oscillation stabilization bit in the standby control register (STBC:SCRDY) is "1_B", the device transits to sub-CR clock mode immediately after the system clock selection bits (SYCC2:RCS1, RCS0) are set to "00_B".
(2)			
(3)		Subclock	When the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "01_B", the device transits to subclock mode after waiting for the subclock oscillation stabilization wait time. The device does not wait for the subclock oscillation stabilization wait time to elapse if the subclock has been oscillating according to the setting of the subclock oscillation enable bit in the system clock control register 2 (SYCC2:SOSCE). In other words, if subclock oscillation is enabled in advance and the subclock oscillation stabilization bit in the standby control register (SYCC:SRDY) is "1_B", the device transits to subclock mode immediately after the system clock selection bits (SYCC2:RCS1, RCS0) are set to "01_B".
(4)			
(5)		Main clock	When the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "11_B", the device transits to main clock mode after waiting for the main clock oscillation stabilization wait time. The device does not wait for the main clock oscillation stabilization wait time to elapse if the main clock has been oscillating according to the setting of the main clock oscillation enable bit in the system clock control register 2 (SYCC2:MOSCE). In other words, if main clock oscillation is enabled in advance and the main clock oscillation stabilization bit in the standby control register (SYCC:MRDY) is "1_B", the device transits to main clock mode immediately after the system clock selection bits (SYCC2:RCS1, RCS0) are set to "11_B".
(6)			
(7)	Main clock	Main CR clock	When the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "10_B", the device transits to main CR clock mode after waiting for the main CR clock oscillation stabilization wait time. The device does not wait for the main CR clock oscillation stabilization wait time to elapse if the main CR clock has been oscillating according to the setting of the main clock oscillation enable bit in the system clock control register 2 (SYCC2:MCRE). In other words, if main CR clock oscillation is enabled in advance and the main CR clock oscillation stabilization bit in the standby control register (SYCC:MCRDY) is "1_B", the device transits to main CR clock mode immediately after the system clock selection bits (SYCC2:RCS1, RCS0) are set to "10_B".
(8)			
(9)		Sub-CR clock	Same as (1) and (2)
(10)			
(11)		Subclock	Same as (3) and (4)
(12)			

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

	Current State	Next State	Description
(13)	Sub-CR clock	Main CR clock	When the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "10_B", the device transits to main CR clock mode after waiting for the main CR clock oscillation stabilization wait time.
(14)		Main clock	When the system clock selection bits in the system clock control register 2 (SYCC2:RCS1, RCS0) are set to "11_B", the device transits to main clock mode after waiting for the main clock oscillation stabilization wait time. Same as (3) and (4)
(15)		Subclock	Same as (3) and (4)
(16)			
(17)	Subclock	Main CR clock	Same as (13)
(18)		Main clock	Same as (15)
(19)		Sub-CR clock	Same as (1) and (2)
(20)			

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

There are four standby modes: sleep mode, stop mode, timebase timer mode and watch mode.

■ Overview of Transiting to and Returning from Standby Mode

There are four standby modes: sleep mode, stop mode, timebase timer mode, and watch mode. The device transits to standby mode according to the settings in the standby control register (STBC).

The device is released from standby mode by an interrupt or a reset. Before transiting to normal operation, the device may wait for the oscillation stabilization wait time to elapse if necessary.

If the clock mode returns from standby mode due to a reset, the device returns to main clock mode. If the clock mode returns from standby mode due to an interrupt, before transiting to standby mode, the device returns to the clock mode in which the device was.

■ Pin States in Standby Mode

The pin state setting bit (STBC:SPL) of the standby control register can be used to keep the preceding state of an I/O port or a peripheral resource pin before its transition to stop mode, timebase timer mode or watch mode, and to set an I/O port or a peripheral resource pin to high impedance in stop mode, timebase timer mode or watch mode.

See "APPENDIX D Pin States of MB95200H/210H Series" for the states of all pins in standby mode.

6.8.1 Notes on Using Standby Mode

Even if the standby control register (STBC) sets standby mode, transition to standby mode does not occur when an interrupt request has been generated 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.

■ Insert at least three NOP instructions immediately after 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, insert at least three NOP instructions.

The device still operates normally even if instructions other than NOP instructions are inserted after the instruction that sets the device to transit to standby mode. On this occasion, the following two events may occur. Firstly, an instruction that should be executed after the standby mode is released may be executed before the device transits to standby mode. Secondly, the device may transit to standby mode while an instruction is being executed, and the execution of that same instruction is resumed after the device is released from standby mode (increasing the number of instruction execution cycles).

■ Check that clock mode transition has been completed before setting the standby mode.

Before setting the standby mode, ensure that clock-mode transition has been completed by comparing the values of the clock mode monitor bits (SYCC2:RCM1, RCM0) and clock mode setting bits (SYCC2:RCS1, RCS0) in the system clock control register.

■ An interrupt request may suppress the transition to standby mode.

When the standby mode is set with an interrupt request whose interrupt level is higher than "11_B" having been issued, the device ignores the value written to the standby control register and continues executing instructions without transiting to the standby mode set. Even after the interrupt of that interrupt request is processed, the device does not transit to the standby mode set.

The same operations are executed when interrupts are disabled by the interrupt enable flag (CCR:I) and the interrupt level bits (CCR:IL1, IL0) of the condition code register of the CPU.

■ The standby mode is also released when the CPU rejects interrupts.

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

After exiting standby mode

After being released from standby mode, the device processes interrupts if interrupts are to be accepted according to the settings of the condition code register (CCR) of the CPU. If interrupts are not to be accepted according to the settings of CCR, the device resumes instruction execution from the instruction following the one executed before the device transits to standby mode.

■ Standby Mode State Transition Diagrams

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 External Clock Product)

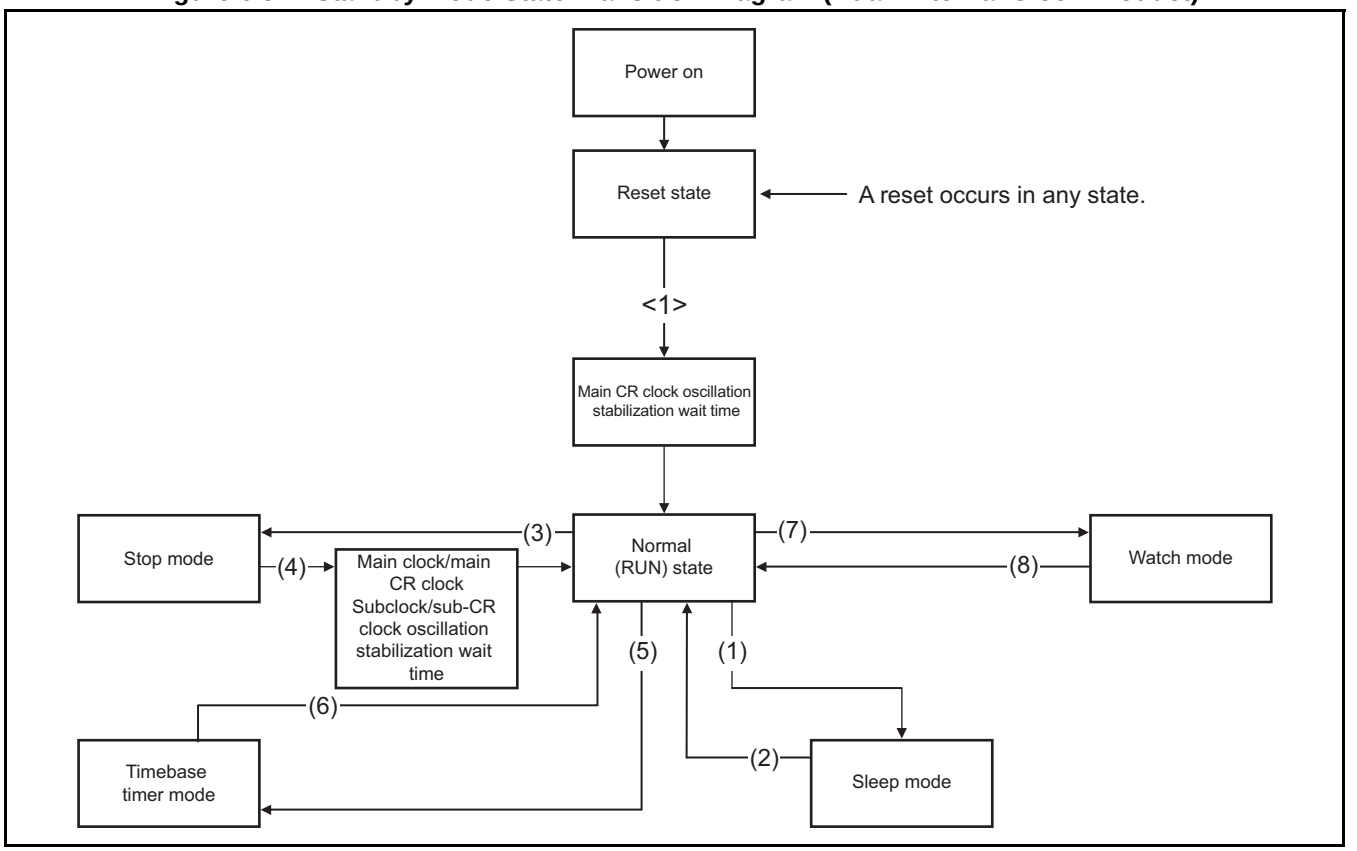


Figure 6.8-2 Standby Mode State Transition Diagram (Single External Clock Product)

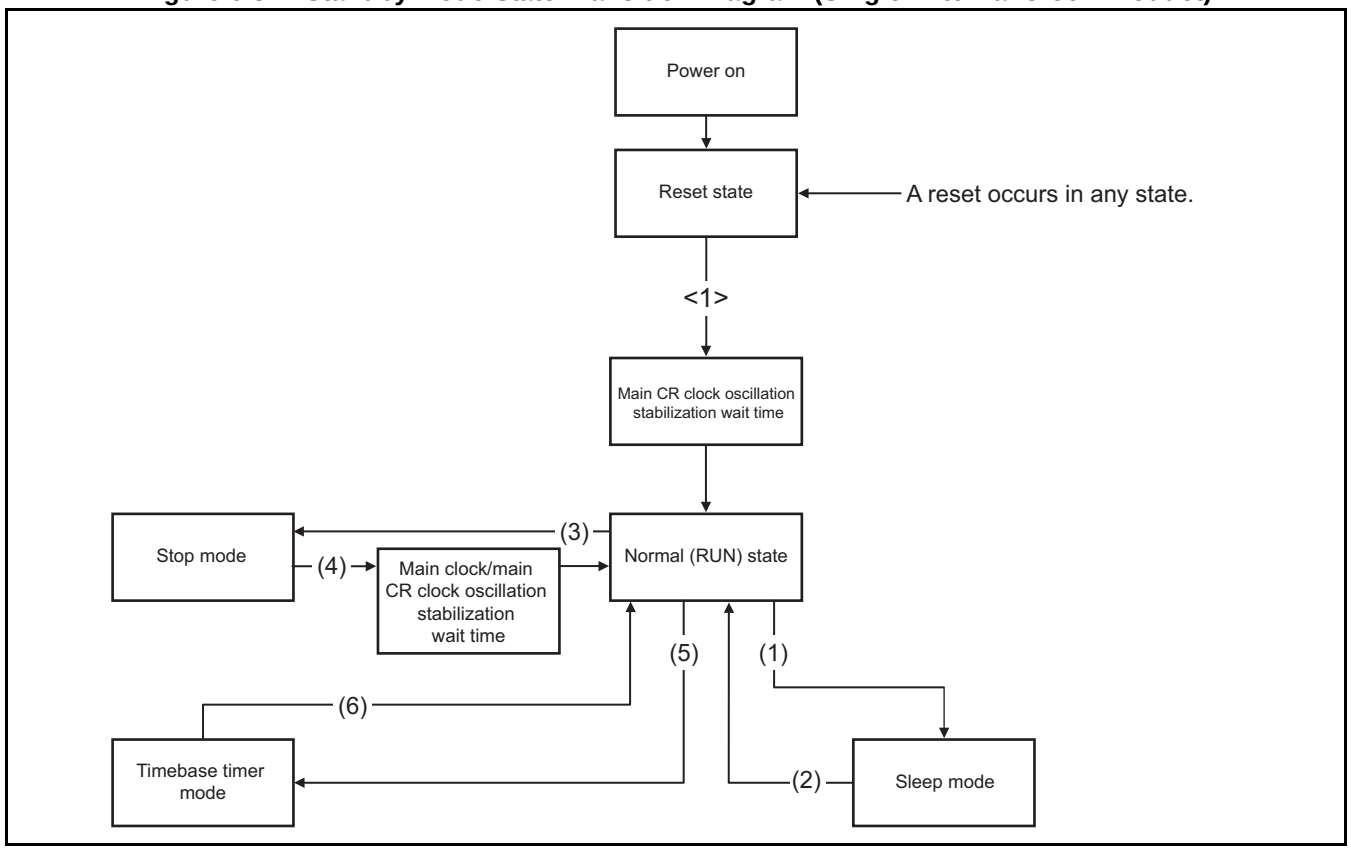


Table 6.8-1 State Transition Table (Transitions to and from Standby Modes)

	State Transition	Description
<1>	Normal operation after reset state	After a reset, the device transits to main CR clock mode. If the reset that has occurred is a power-on reset, a watchdog reset, a software reset, or an external reset, the device always wait for the main CR clock oscillation stabilization wait time to elapse.
(1)	Sleep mode	The device transits to 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 transits to stop mode when "1" is written to the stop bit in the standby control register (STBC:STP).
(4)		In response to an external interrupt, after waiting for the elapse of the oscillation stabilization wait time required according to the current clock mode, the device returns to the RUN state. Regarding the elapse of which oscillation stabilization wait time for which the device waits, if the oscillation stabilization wait time required according to the current clock mode is longer than the main CR/sub-CR oscillation stabilization wait time, the device waits for the oscillation stabilization wait time required; otherwise, the device waits for the main CR/sub-CR oscillation stabilization wait time.
(5)	Timebase timer mode	The device transits to timebase timer mode when "1" is written to the watch bit in the standby control register (STBC:TMD) in main clock mode or main CR clock mode.
(6)		
(7)	Watch mode	The device transits to watch mode when "1" is written to the watch bit in the standby control register (STBC:TMD) in subclock mode or sub-CR clock mode.
(8)		

6.8.2 Sleep Mode

In sleep mode, the operations of the CPU and watchdog timer are stopped.

■ Operations in Sleep Mode

In sleep mode, the CPU and the operating clock for the watchdog timer are stopped. The CPU retains the contents of registers and RAM existing at the point immediately before the device transits to sleep mode and stops; however, all peripheral functions except the watchdog timer continue operating.

In the case of hardware watchdog timer, if it is enabled in standby mode by the non-volatile register function, in sleep mode, the sub-CR clock does not stop and the hardware watchdog timer operates. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

- Transition to sleep mode

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

- Release from sleep mode

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

6.8.3 Stop Mode

In stop mode, the main clock, the main CR clock and the subclock are stopped.

■ Operations in Stop Mode

In stop mode, the main clock, the main CR clock, and the subclock are stopped. In this mode, while retaining the contents of registers and RAM existing at the point immediately before the device transits to stop mode, the device stops all functions except external interrupt and low-voltage detection reset.

In the case of hardware watchdog timer, if it is enabled in standby mode by the non-volatile register function, in stop mode, the sub-CR clock does not stop and the hardware watchdog timer operates. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

● Transition to stop mode

Writing "1" to the stop bit in the standby control register (STBC:STP) causes the device to transit to stop mode. At that point, if the pin state setting bit in the standby control register (STBC:SPL) is "0", the states of the external pins are kept; if the SPL bit is "1", the states of the external pins become high impedance (a pin is pulled up if the pull-up resistor connection for that pin is selected in the pull-up setting register).

In main clock mode or main CR clock mode, while the device is waiting for main clock oscillation to stabilize after being released from stop mode by an interrupt, a timebase timer interrupt request may be generated. If the interrupt interval time of the timebase timer is shorter than the main clock oscillation stabilization wait time, it is advisable to prevent any unexpected interrupt from occurring by disabling interrupt requests output from the timebase timer before making the device transit to stop mode

It is also advisable to disable interrupt requests output from the watch prescaler before making the device transit to stop mode from subclock mode or sub-CR clock mode.

● Release from stop mode

The device is released from stop mode by a reset or an external interrupt. In any clock mode, if the hardware watchdog timer is enabled in standby mode by the non-volatile register function, the sub-CR clock does not stop, and the watchdog timer and the watch prescaler operate in stop mode. The device can also be released from stop mode by an interrupt from the watch prescaler. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

Note:

If the device is released from stop mode by an interrupt, a peripheral function having transited to stop mode during operation resumes operating from the point at which it transited to stop mode. Therefore, some settings of that peripheral function, such as the initial interval time of the interval timer, become undefined. Initialize that peripheral function if necessary after releasing the device from stop mode.

6.8.4 Timebase Timer Mode

In timebase timer mode, only the main clock oscillator, the subclock oscillator, the timebase timer, and the watch prescaler operate. The CPU and the operating clock for peripheral functions are stopped in this mode.

■ Operations in Timebase Timer Mode

The timebase timer mode is a mode in which main clock supply is stopped except the clock supply to the timebase timer. In this mode, while retaining the contents of registers and RAM existing at the point immediately before the device transits to timebase timer mode, the device stops all functions except the timebase timer, external interrupt and low-voltage detection reset.

Subclock oscillation and sub-CR clock oscillation can be enabled or disabled by the subclock oscillation enable bit and the sub-CR clock oscillation enable bit in the system clock control register 2 (SYCC2:SOSCE, SCRE) respectively. If the subclock oscillates, the watch prescaler operates.

In the case of hardware watchdog timer, if it is enabled in standby mode by the non-volatile register function, in timebase timer mode, the sub-CR clock does not stop and the hardware watchdog timer operates. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

- Transition to timebase timer mode

If the system clock monitor bits in the system clock control register 2 (SYCC2:RCM1, RCM0) are "10_B" or "11_B", writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to transit to timebase timer mode.

The device can transit to timebase timer mode only when the clock mode is main clock mode or main CR clock mode.

After the device transits to timebase time mode, if the pin state setting bit in the standby control register (STBC:SPL) is "0", the states of the external pins are kept; if the SPL bit is "1", the states of the external pins become high impedance (a pin is pulled up if the pull-up resistor connection for that pin is selected in the pull-up setting register)

- Release from timebase timer mode

The device is released from timebase timer mode by a reset, a timebase timer interrupt, or an external interrupt.

Subclock oscillation and sub-CR clock oscillation can be enabled or disabled by setting the subclock oscillation enable bit (SOSCE) and the sub-CR clock oscillation enable bit (SCRE) in the system clock control register 2 (SYCC2). When the subclock oscillates, the device can be released from timebase timer mode by an interrupt from the watch prescaler.

Note:

If the device is released from timebase timer mode by an interrupt, a peripheral function having transited to timebase timer mode during operation resumes operating from the point at which it transited to timebase timer mode. Therefore, some settings of that peripheral function, such as the initial interval time of the interval timer, become undefined. Initialize that peripheral function if necessary after releasing the device from timebase timer mode.

6.8.5 Watch Mode

In watch mode, only the subclock, the sub-CR clock and the watch prescaler operate. The CPU and the operating clock for peripheral functions are stopped in this mode.

■ Operations in Watch Mode

In watch mode, while retaining the contents of registers and RAM existing at the point immediately before the device transits to watch mode, the device stops all functions except the watch prescaler, external interrupt and low-voltage detection reset.

In the case of hardware watchdog timer, if it is enabled in standby mode by the non-volatile register function, in watch mode, the sub-CR clock does not stop and the hardware watchdog timer operates. For details, see "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)".

- Transition to watch mode

If the system clock monitor bits in the system clock control register 2 (SYCC2:RCM1, RCM0) are "00_B" or "01_B", writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to transit to watch mode.

The device can transit to watch mode only when the clock mode is subclock mode or sub-CR clock mode.

After the device transits to watch mode, if the pin state setting bit in the standby control register (STBC:SPL) is "0", the states of the external pins are kept; if the SPL bit is "1", the states of the external pins become high impedance (a pin is pulled up if the pull-up resistor connection for that pin is selected in the pull-up setting register)

- Release from watch mode

The device is released from watch mode by a reset, a watch interrupt, or an external interrupt.

Note:

If the device is released from watch mode by an interrupt, a peripheral function having transited to timebase timer mode during operation resumes operating from the point at which it transited to timebase timer mode. Therefore, some settings of that peripheral function, such as the initial interval time of the interval timer, become undefined. Initialize that peripheral function if necessary after releasing the device from timebase timer mode.

6.9 Clock Oscillator Circuit

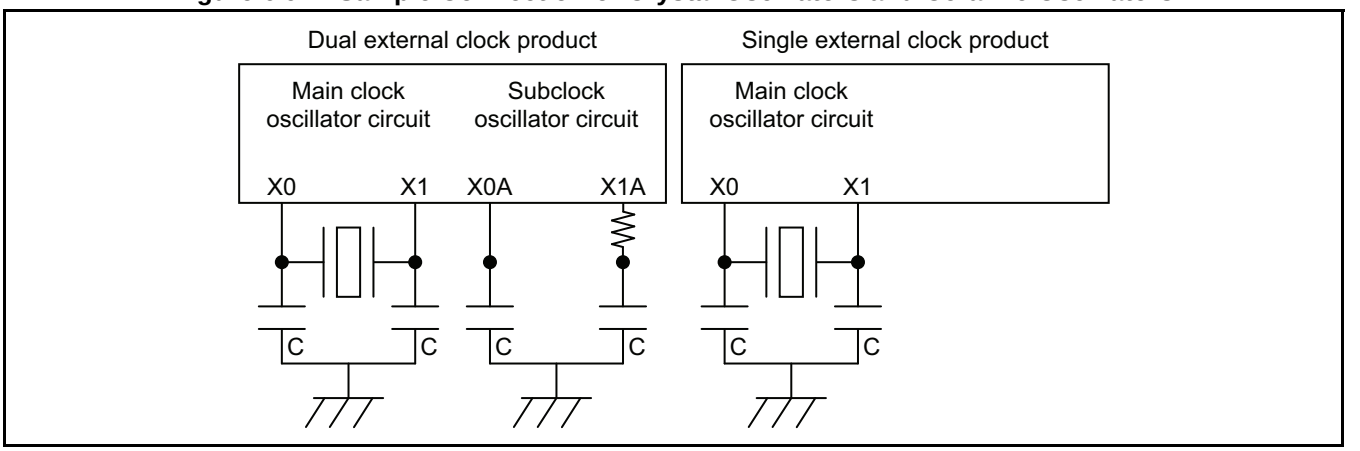
The clock oscillator circuit generates an internal clock with an oscillator connected to the clock oscillation pin or by inputting a clock signal to the clock oscillation pin.

■ Clock Oscillator Circuit

● Using crystal oscillators and ceramic oscillators

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

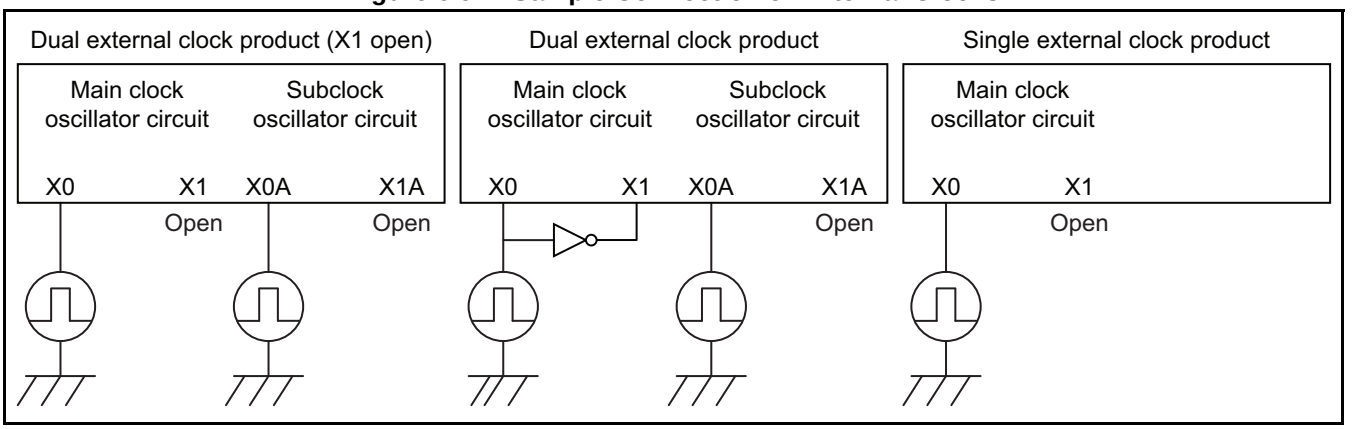
Figure 6.9-1 Sample Connection of Crystal Oscillators and Ceramic Oscillators



● Using external clock

As shown in Figure 6.9-2, connect the external clock to the X0 pin while leaving the X1 pin unconnected or supplying inverted clock of the X0 pin to the X1 pin. (Refer to the data sheet of this series.) To supply clock signals to the subclock from an external clock, connect that external clock to the X0A pin while leaving the X1A pin unconnected.

Figure 6.9-2 Sample Connection of External Clocks



6.10 Overview of Prescaler

The prescaler generates the count clock source to be supplied to various peripheral functions from the machine clock (MCLK) and the count clock output from the timebase timer.

■ Prescaler

The prescaler generates the count clock source to be supplied to various peripheral functions from the machine clock (MCLK) with which the CPU operates and from the count clock ($2^7/F_{CH}$, $2^8/F_{CH}$, $2^6/F_{CRH}$ or $2^7/F_{CRH}$) output from the timebase timer. The count clock source is a clock whose frequency is divided by the prescaler or a buffered clock. The peripheral functions listed below use the clock whose frequency is divided by the prescaler as the count clock source.

The prescaler has no control register and always operates with the machine clock (MCLK) and the count clock ($2^7/F_{CH}$, $2^8/F_{CH}$, $2^6/F_{CRH}$ or $2^7/F_{CRH}$) of the timebase timer.

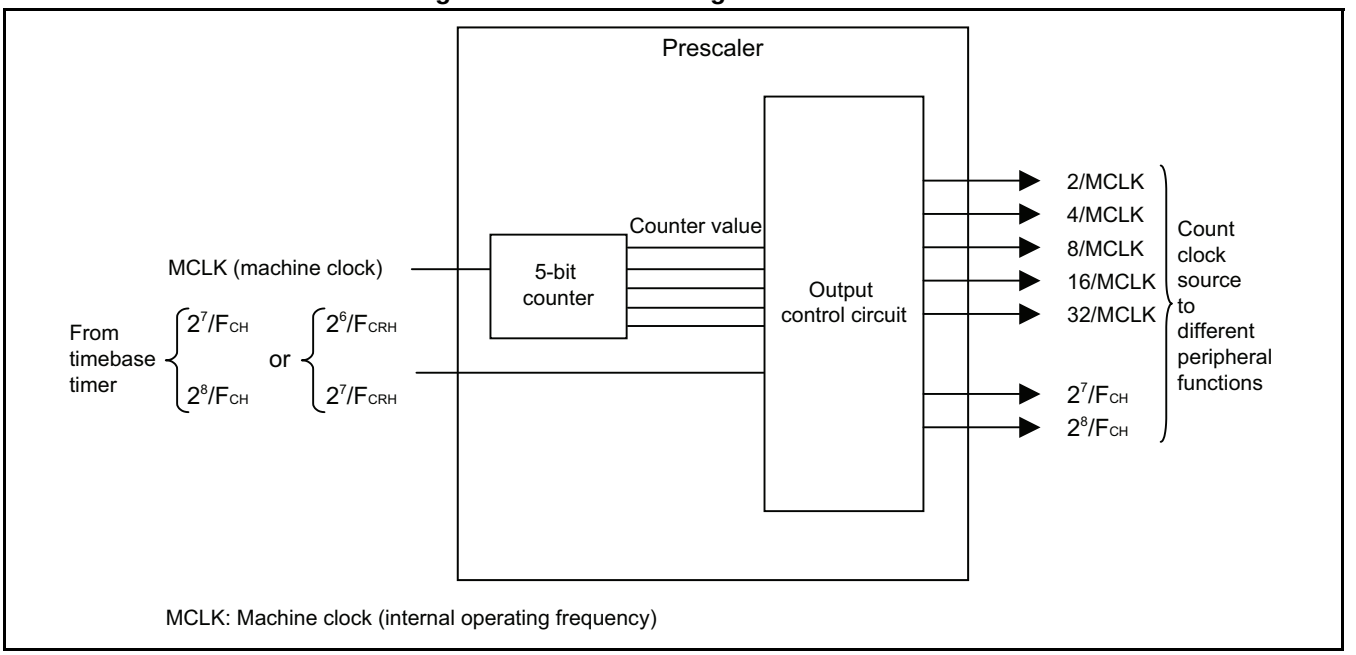
- 8/16-bit composite timer
- 8/10-bit A/D converter

6.11 Configuration of Prescaler

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

■ Block Diagram of Prescaler

Figure 6.11-1 Block Diagram of Prescaler



- 5-bit counter

  This counter counts the machine clock (MCLK) and outputs the count value to the output control circuit.

- Output control circuit

  Based on the 5-bit counter value, this circuit supplies clocks generated by dividing the machine clock (MCLK) by 2, 4, 8, 16, or 32 to individual peripheral functions. The circuit also buffers the clock from the timebase timer ($2^7/F_{CH}$, $2^8/F_{CH}$, $2^6/F_{CRH}$ or $2^7/F_{CRH}$) and supplies it to peripheral functions.

■ Input Clock

The prescaler uses the machine clock, or the output clock of the timebase timer as the input clock.

■ Output Clock

The prescaler supplies clocks to the 8/10-bit composite timer and the 8/10-bit A/D converter.

6.12 Operation of Prescaler

The prescaler generates count clock sources to different peripheral functions.

■ Operation of Prescaler

The prescaler generates count clock sources from a clock whose frequency is generated by dividing the machine clock (MCLK) and from buffered signals from the timebase timer ($2^7/F_{CH}$, $2^8/F_{CH}$, $2^6/F_{CRH}$ or $2^7/F_{CRH}$), and supplies them to different peripheral functions. The prescaler keeps operating while the machine clock and the clocks from the timebase timer are being supplied.

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

Table 6.12-1 Count Clock Sources Generated by Prescaler

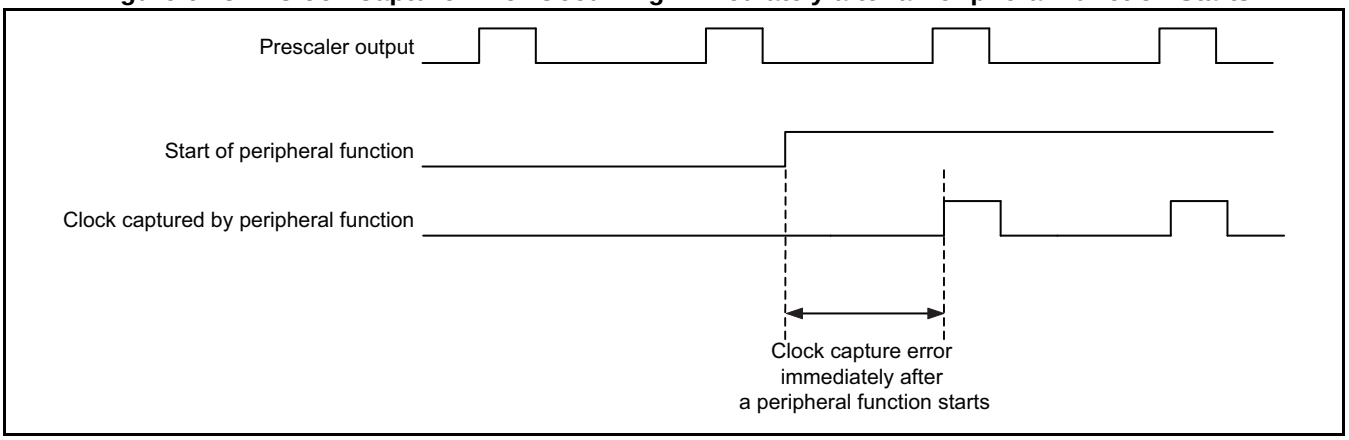
Count clock source cycle	Cycle (F_{CH} =10 MHz, MCLK=10 MHz)		Cycle (F_{CH} =16 MHz, MCLK=16 MHz)		Cycle (F_{CH} =16.25 MHz, MCLK=16.25 MHz)	
2/MCLK	MCLK/2	(5 MHz)	MCLK/2	(8 MHz)	MCLK/2	(8.125 MHz)
4/MCLK	MCLK/4	(2.5 MHz)	MCLK/4	(4 MHz)	MCLK/4	(4.0625 MHz)
8/MCLK	MCLK/8	(1.25 MHz)	MCLK/8	(2 MHz)	MCLK/8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

6.13 Notes on Using Prescaler

This section provides notes on using the prescaler.

The prescaler operates with the machine clock and the clock generated from the timebase timer, and keeps operating while those clocks are being supplied. Therefore, in the operation immediately after a peripheral resource is started, an error of up to one cycle of the clock source captured by that peripheral resource, depending on the output value of the prescaler.

Figure 6.13-1 Clock Capture Error Occurring Immediately after a Peripheral Function Starts



The prescaler count value affects the following peripheral functions:

- 8/16-bit composite timer
- 8/10-bit A/D converter

CHAPTER 7
RESET

This section describes the reset operation.

7.1 Reset Operation

When a reset source occurs, the CPU immediately stops the process being executed and enters the reset release wait state. When the reset is released, the CPU reads mode data and the reset vector from the internal ROM (mode fetch). When the power is switched on or when the device is released from a reset in subclock mode, sub-CR clock mode, or stop mode, the CPU performs mode fetch after the oscillation stabilization wait time has elapsed.

■ Reset Sources

There are four reset sources for the reset.

Table 7.1-1 Reset Sources

Reset source	Reset condition
External reset	"L" level is 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 overflows.
Power-on reset/ Low-voltage detection reset	The power is switched on or the supply voltage falls below the detection voltage. (Option)

● External reset

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

An external input reset signal is received asynchronously with the operating clock of the microcontroller via the internal noise filter and then generates an internal reset signal that is synchronized with the machine clock to initialize the internal circuit. Therefore, the operating clock of the microcontroller is necessary for initializing the internal circuit. In order to operate with the external clock, external clock signals must be input. However, the external pins (including I/O ports and peripheral functions) are reset asynchronously. In addition, there is a standard value of the pulse width for external reset input. If the value is below the standard value, a reset signal may not be accepted.

The standard value is shown in the data sheet of this series. Design an external reset circuit that satisfies the standard value.

● 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 predetermined period of time.

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

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

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

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 equivalent to that of the power-on reset. All information relating to the power-on reset of this hardware manual also applies to the low-voltage detection reset.

For details of the low-voltage detection reset, see "CHAPTER 18 LOW-VOLTAGE DETECTION RESET CIRCUIT".

■ Reset Time

In the case of a software reset or a watchdog reset, the reset time consists 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 initial machine clock frequency after the reset (1/32 of the main clock frequency). However, the reset time may be extended by the RAM access protection function, which suppresses resets during RAM access, by the machine clock cycle of the frequency selected before the reset. In addition, when in main clock oscillation stabilization standby mode, the reset time is further extended for the oscillation stabilization wait time. Both the external reset and the reset are affected by the RAM access protection function and the main clock oscillation stabilization wait time.

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

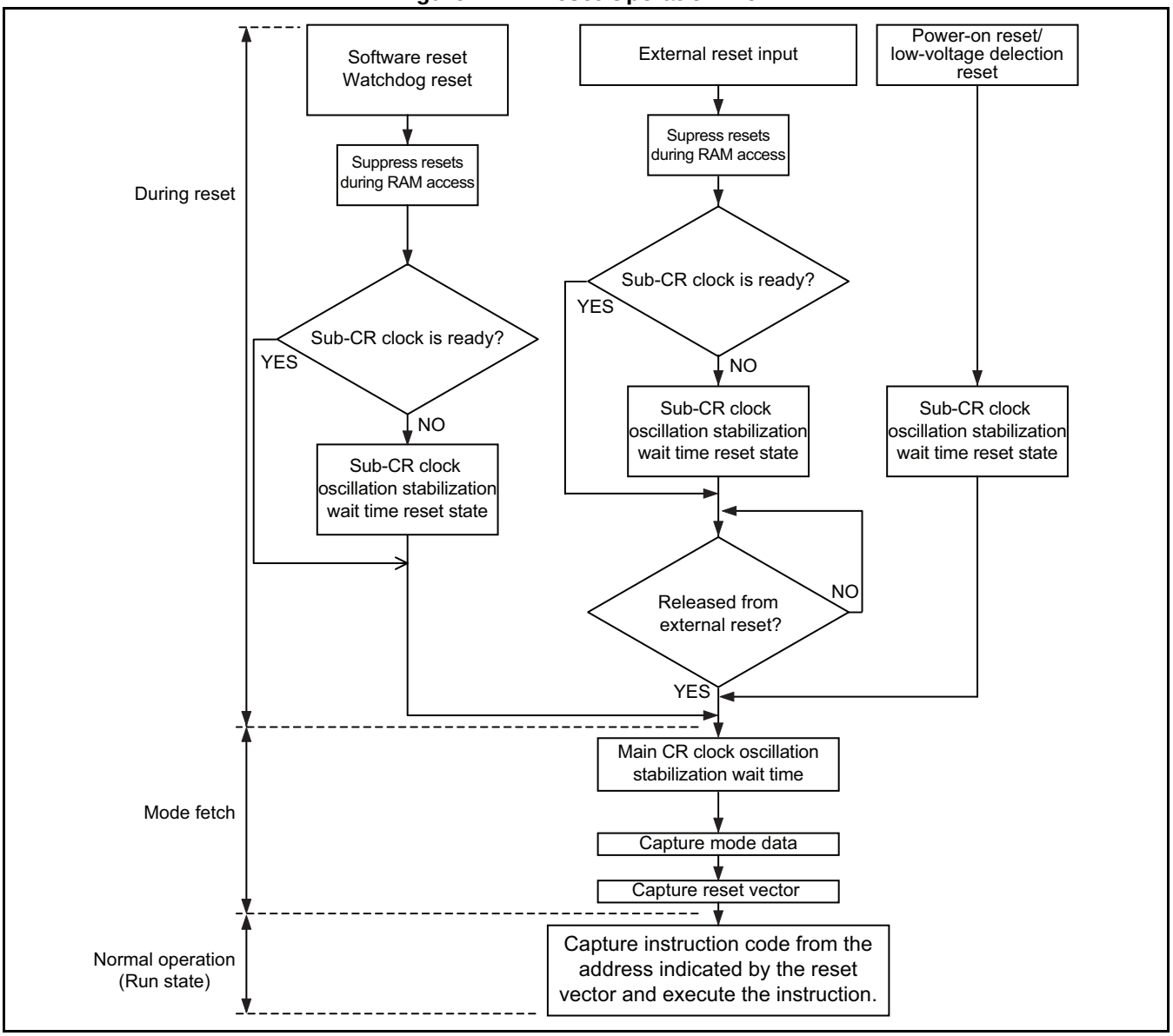
■ Reset Output

The reset pin of a 5-V product with reset output (For details, see Table 1.2-1.) outputs "L" level during a reset. However, in the case of an external reset, a reset pin does not output "L" level.

The reset pin of a 3-V/5-V product with no reset output does not output "L" level.

■ Overview of Reset Operation

Figure 7.1-1 Reset Operation Flow



In any reset, the CPU performs mode fetch after the main CR clock oscillation stabilization wait time elapses.

■ 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 state. However, during RAM access execution, in order to protect the RAM access, an internal reset signal synchronized with the machine clock is generated after an RAM access ends. This function prevents a word-data write operation from being interrupted by a reset while data of two bytes is being written.

■ Pin State During a Reset

When a reset occurs, an I/O port or a peripheral resource pin remains high impedance until the setting of that I/O port or that peripheral resource pin by software is executed after the reset is released.

Note:

Connect a pull-up resistor to a pin that becomes high impedance during a reset to prevent the device from malfunctioning.

For details of the states of all pins during a reset, see "APPENDIX D Pin States of MB95200H/210H Series".

7.2 Reset Source Register (RSRR)

The reset source register indicates the source of a reset generated.

■ Configuration of Reset Source Register (RSRR)

Figure 7.2-1 Configuration of Reset Source Register (RSRR)

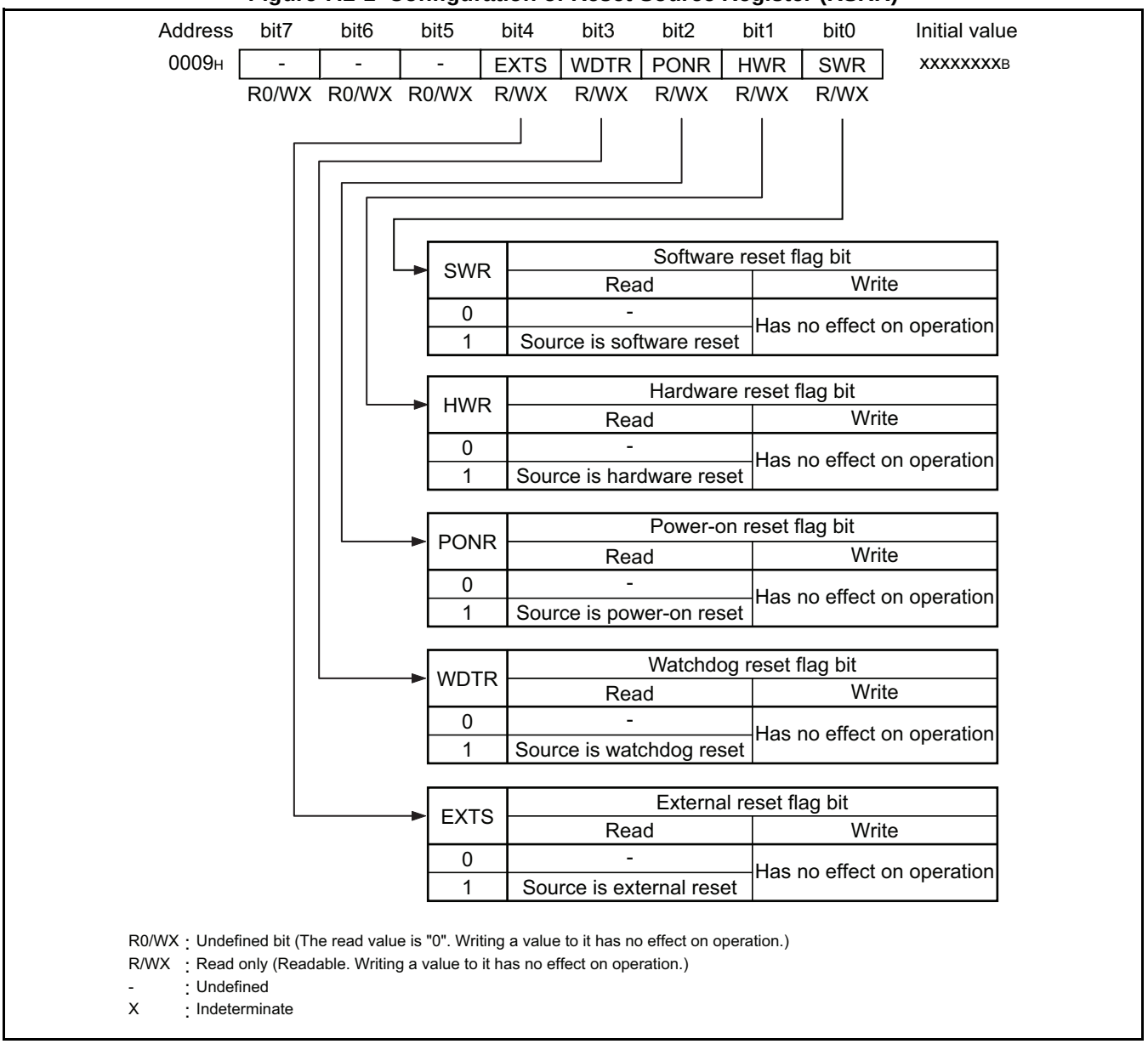


Table 7.2-1 Functions of Bits in Reset Source Register (RSRR)

Bit name		Function
bit7 to bit5	Undefined bits	When these bits are read, they always return "0". These bits are read-only. Writing values to them has no effect on operation.
bit4	EXTS: External reset flag bit	When this bit is set to "1", that indicates an external reset has occurred. When any other reset occurs, this bit retains the value that has existed before such reset occurs. • A read access to this bit clears it to "0". • This bit is read-only. Writing a value to it has no effect on operation.
bit3	WDTR: Watchdog reset flag bit	When this bit is set to "1", that indicates a watchdog reset has occurred. When any other reset occurs, this bit retains the value that has existed before such reset occurs. • A read access to this bit clears it to "0". • This bit is read-only. Writing a value to it has no effect on operation.
bit2	PONR: Power-on reset flag bit	When this bit is set to "1", that indicates a power-on reset or a low-voltage detection reset (option) has occurred. When any other reset occurs, this bit retains the value that has existed before such reset occurs • The low-voltage detection reset function is available only in certain products. • A read access to this bit clears it to "0". • This bit is read-only. Writing a value to it has no effect on operation.
bit1	HWR: Hardware reset flag bit	When this bit is set to "1", that indicates a reset other than software reset has occurred. Therefore, when any of bit 2 to bit 5 is set to "1", this bit is set to "1" as well. When a software reset occurs, the bit retains the value that has existed before the software reset occurs. • A read access to this bit clears it to "0". • This bit is read-only. Writing a value to it has no effect on operation.
bit0	SWR: Software reset flag bit	When this bit is set to "1", that indicates a software reset has occurred. When a hardware reset (external reset, watchdog reset, power-on reset, low-voltage detection reset) occurs, the bit retains the value that has existed before the hardware reset occurs. • A read access to this bit or a power-on reset clears it to "0". • This bit is read-only. Writing a value to it has no effect on operation.

Note:

Since reading the reset source register clears its contents, save the contents of this register to the RAM before using those contents for calculation.

■ State of Reset Source Register (RSRR)

Table 7.2-2 State of Reset Source Register

Reset source	–	–	EXTS	WDTR	PONR	HWR	SWR
Power-on reset/Low-voltage detection reset	–	–	×	×	1	1	0
Software reset	–	–	△	△	△	△	1
Watchdog reset	–	–	△	1	△	1	△
External reset	–	–	1	△	△	1	△

1: Flag set

△: Previous state kept

×: Indeterminate

EXTS: When this bit is set to "1", that indicates an external reset has occurred.

WDTR: When this bit is set to "1", that indicates a watchdog reset has occurred.

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

HWR: When this bit is set to "1", that indicates one of the following reset has occurred: an external reset, a watchdog reset, a power-on reset or a low-voltage detection reset (option).

SWR: When this bit is set to "1", that indicates that a software reset has occurred.

7.3 Notes on Using Reset

This section provides notes on using the reset.

■ Notes on Using Reset

● Initialization of registers and bits by reset source

There are registers and bits that are not initialized by a reset source.

- The type of reset source determines which bit in the reset source register (RSRR) is to be initialized.
- The oscillation stabilization wait time setting register (WATR) of the clock controller is initialized only by a power-on reset.

CHAPTER 8
INTERRUPTS

This chapter describes the interrupts.

8.1 Interrupts

This section describes the interrupts.

■ Overview of Interrupts

The F^{2}MC-8FX family has 24 interrupt request inputs for respective peripheral functions, for each of which an interrupt level can be set independently to each other.

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 notifies the CPU of the generation of the interrupt. The CPU processes that interrupt according to the interrupt acceptance status. The device is released from standby mode by an interrupt request and resumes executing instructions.

■ Interrupt Requests from Peripheral Functions

Table 8.1-1 lists the interrupt requests of respective peripheral functions. When the CPU receives an interrupt request, it branches to the interrupt service routine with the interrupt vector table address corresponding to the interrupt request as the address of the branch destination.

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

While an interrupt is being processed in the interrupt service routine, if another interrupt whose interrupt request is of the same level or below the one of the interrupt being processed is generated, it is processed after the current interrupt service routine is completed. In addition, if multiple interrupt requests that are set to the same interrupt level are made, IRQ0 is at the top of the priority order.

Table 8.1-1 Interrupt Requests and Interrupt Vectors

Interrupt request	Vector table address		Bit name in interrupt level setting register	Priority order of iInterrupt requests of the same level (generated simultaneously)
	Upper	Lower		
IRQ0	$FFFA_H$	$FFFB_H$	L00 [1:0]	Highest
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]	Lowest

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

8.1.1 Interrupt Level Setting Registers (ILR0 to ILR5)

The interrupt level setting registers (ILR0 to ILR5) contain 24 pairs of 2-bit data assigned to the interrupt requests of different peripheral functions. Each pair of bits (interrupt level setting bits) is used to set the interrupt level of an interrupt request.

■ 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	00079H	L03	[1:0]	L02	[1:0]	L01	[1:0]	L00	[1:0]	R/W 11111111B
ILR1	0007AH	L07	[1:0]	L06	[1:0]	L05	[1:0]	L04	[1:0]	R/W 11111111B
ILR2	0007BH	L11	[1:0]	L10	[1:0]	L09	[1:0]	L08	[1:0]	R/W 11111111B
ILR3	0007CH	L15	[1:0]	L14	[1:0]	L13	[1:0]	L12	[1:0]	R/W 11111111B
ILR4	0007DH	L19	[1:0]	L18	[1:0]	L17	[1:0]	L16	[1:0]	R/W 11111111B
ILR5	0007EH	L23	[1:0]	L22	[1:0]	L21	[1:0]	L20	[1:0]	R/W 11111111B

The interrupt level setting registers assign a pair of bits to every interrupt request. The values of interrupt level setting bits in these registers represent the priority of an interrupt request (interrupt level: 0 to 3) in interrupt processing.

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

If the interrupt level of an interrupt request is 3, the CPU ignores that 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	Highest
01	1	↕
10	2	
11	3	Lowest (No interrupt)

XX:00 to 23 Number of an interrupt request

While the main program is being executed, the interrupt level bits in the condition code register (CCR: IL1, IL0) are "11_B".

8.1.2 Interrupt Processing

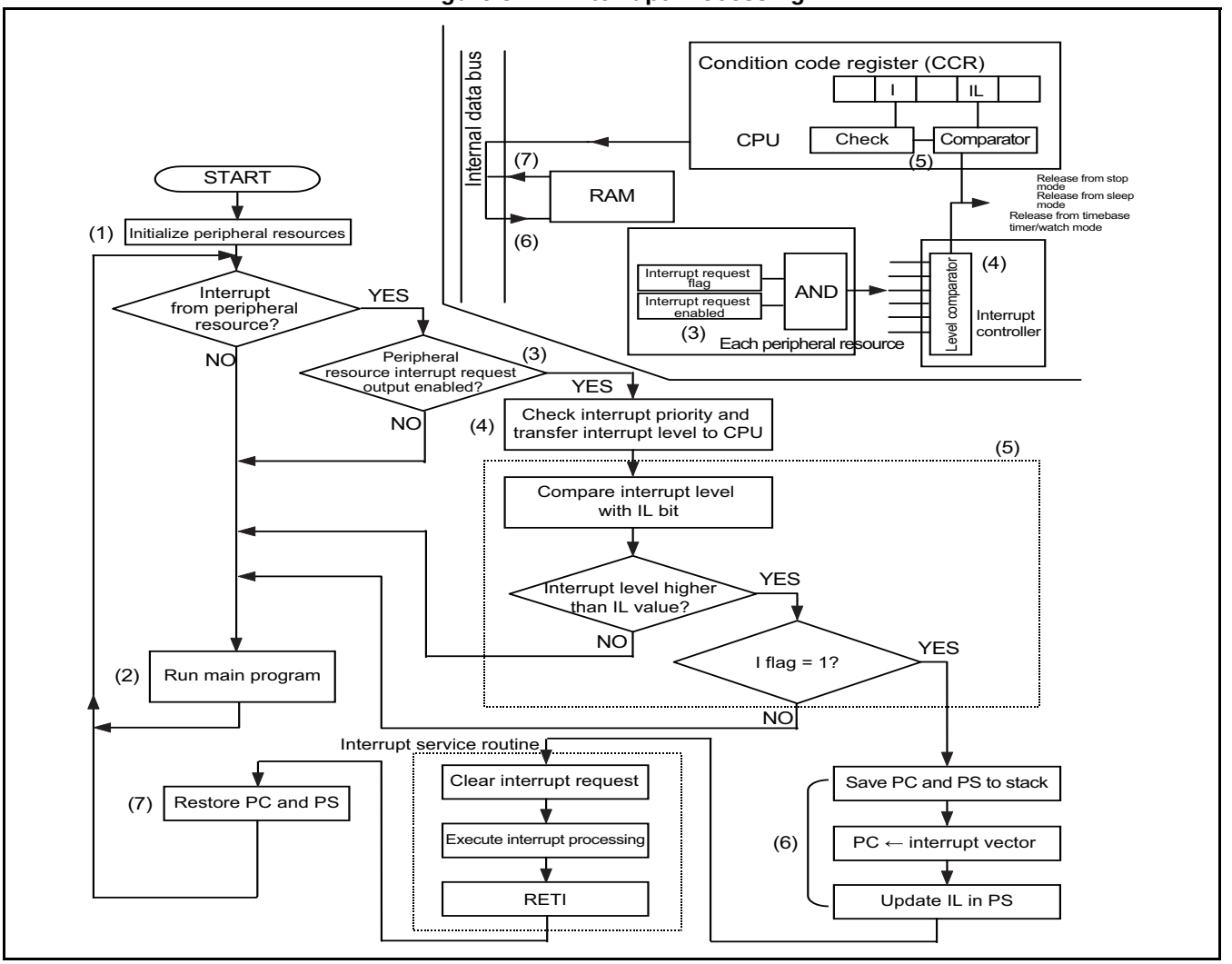
When an interrupt request is made by a peripheral resource, the interrupt controller notifies the CPU of the interrupt level of that interrupt request. When the CPU is ready to accept interrupts, it halts the program it is executing and executes an interrupt service routine.

■ Interrupt Processing

The procedure for processing an interrupt is as follows: the generation of an interrupt source 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 the interrupt level (ILR0 to ILR5 and CCR:IL1, IL0), the checking for interrupt requests of the same interrupt level made simultaneously, and the evaluation of the interrupt enable flag (CCR:I).

Figure 8.1-2 shows the interrupt processing.

Figure 8.1-2 Interrupt Processing



(1) All interrupt requests are disabled immediately after a reset. In the peripheral resource initialization program, initialize those peripheral functions that generate interrupts and set their interrupt levels in their respective interrupt level setting registers (ILR0 to ILR5) before starting operating such peripheral functions. 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. Assigning level 3 to a peripheral resource disables interrupts from that peripheral resource.

(2) Execute the main program (or the interrupt service routine in the case of nested interrupts).

(3) When an interrupt source is generated in a peripheral resource, the interrupt request flag bit for that peripheral resource is set to "1". Provided that the interrupt request enable bit for that peripheral resource has been set to the value that enables interrupts, an interrupt request of that peripheral resource is output to the interrupt controller.

(4) The interrupt controller keeps monitoring interrupt requests from individual peripheral functions and notifies the CPU of the interrupt level having priority over the others among interrupt levels already made. If there are interrupt requests having the same interrupt level, their positions in the priority order are also compared in the interrupt controller.

(5) If the interrupt level received has priority over (smaller interrupt level number) the level set in the interrupt level bits (CCR:IL1, IL0) in the condition code register, the CPU checks the content of the interrupt enable flag (CCR:I), and accepts the interrupt provided that interrupts have been enabled (CCR:I = 1).

(6) The CPU saves the contents of the program counter (PC) and the program status (PS) to the stack, captures the start address of the interrupt service routine from the corresponding interrupt vector table address, modifies the values of the interrupt level bits in the condition code register (CCR:IL1, IL0) to the values of the interrupt level received, then starts executing the interrupt service routine.

(7) Finally, the CPU uses the RETI instruction to restore the values of the program counter (PC) and the program status (PS) from the stack and resumes executing the instruction following the one executed just before the interrupt.

Note:

The interrupt request flag bit for a peripheral resource is not automatically cleared to "0" after an interrupt request is accepted. Therefore, such bit must be cleared to "0" by using a program (writing "0" to the interrupt request flag bit) in the interrupt service routine.

The low-power consumption (standby mode) is released by an interrupt. For details, see "6.8 Operations in Low-power Consumption Mode (Standby Mode)".

8.1.3 Nested Interrupts

Different interrupt levels can be assigned to multiple interrupt requests from peripheral functions in the interrupt level setting registers (ILR0 to ILR5) to process nested interrupts.

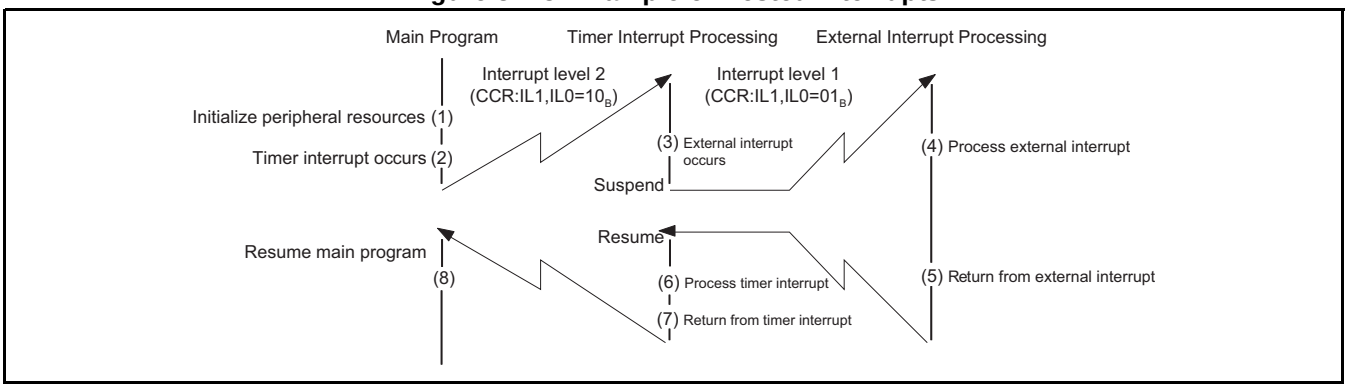
■ Nested Interrupts

During the execution of an interrupt service routine, if another interrupt request whose interrupt level has priority over the interrupt level of the interrupt being processed is made, the CPU suspends the current interrupt processing and accepts the interrupt request given priority. The interrupt level of an interrupt request can be set to 0 to 3. If it is set to 3, the CPU does not accept that interrupt request.

[Example: Nested interrupts]

In the following example of nested interrupts, assuming that the external interrupt is to be given priority over the timer interrupt, the interrupt level of the timer interrupt is set to 2 and that of the external interrupt to 1. If the external interrupt is generated while the timer interrupt is being processed, they are processed as shown in Figure 8.1-3.

Figure 8.1-3 Example of Nested Interrupts



- While the 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 timer interrupt (level 2 in this example). If an interrupt request whose interrupt level has priority over the interrupt level of the timer interrupt (level 1 in the example) is made, that interrupt is processed first.
- To temporarily disable nested interrupts processing while the timer interrupt is being processed, disable interrupts by setting the interrupt enable flag in the condition code register (CCR:I) to "0", or set the interrupt level bits (CCR:IL1, IL0) to "00_B".
- After the interrupt processing is completed, if the interrupt return instruction (RETI) is executed, the value of the program counter (PC) and that of the program status (PS) are restored, and the CPU resumes executing the program interrupted. In addition, the values of the condition code register (CCR) return to the ones existing before the interrupt due to the restoration of the value of the program status (PS).

8.1.4 Interrupt Processing Time

Before the CPU enters the interrupt service routine after an interrupt request is made, it needs to wait for the interrupt processing time, which consists of the time between the occurrence of an interrupt request and the end of the execution of the instruction being executed, and the interrupt handling time (the time required to initiate interrupt processing) to elapse. The maximum interrupt processing time is 26 machine clock cycles.

■ Interrupt Processing Time

Before executing the interrupt service routine after an interrupt request is made, the CPU needs to wait for the interrupt request sampling wait time and the interrupt handling time to elapse.

● Interrupt request sampling wait time

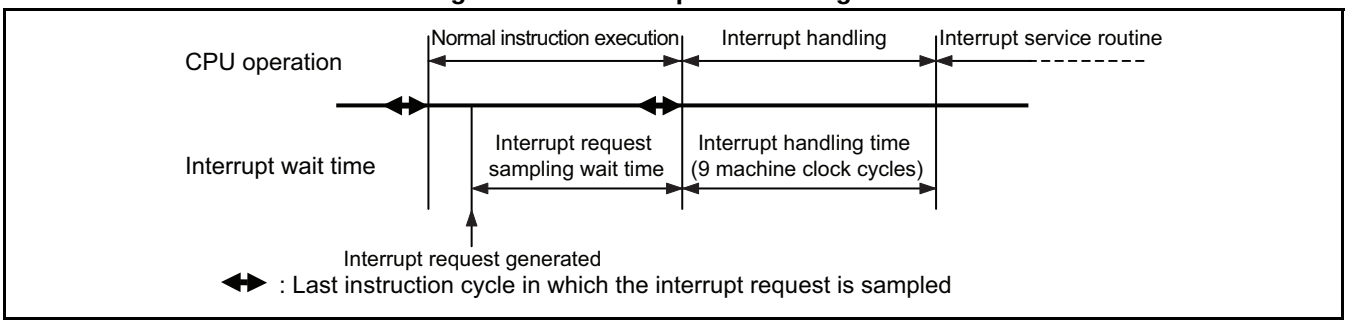
The CPU decides whether an interrupt request has occurred by sampling the interrupt request during the last cycle of an instruction. Therefore, the CPU cannot recognize interrupt requests while executing an instruction. This sampling wait time reaches its maximum when an interrupt request occurs immediately after the CPU starts executing the DIVU instruction, whose execution cycle is the longest (17 machine clock cycles).

● Interrupt handling time

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

- Saves the value of the program counter (PC) and that of the program status (PS) to the stack.
- 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).

Figure 8.1-4 Interrupt Processing Time



When an interrupt request occurs immediately after the CPU starts executing the DIVU instruction, whose execution cycle is the longest (17 machine clock cycles), the interrupt processing time spans 26 machine clock cycles.

The span of a machine clock cycle varies depending on the clock mode and main clock speed change (gear function). For details, see "CHAPTER 6 CLOCK CONTROLLER".

8.1.5 Stack Operation During Interrupt Processing

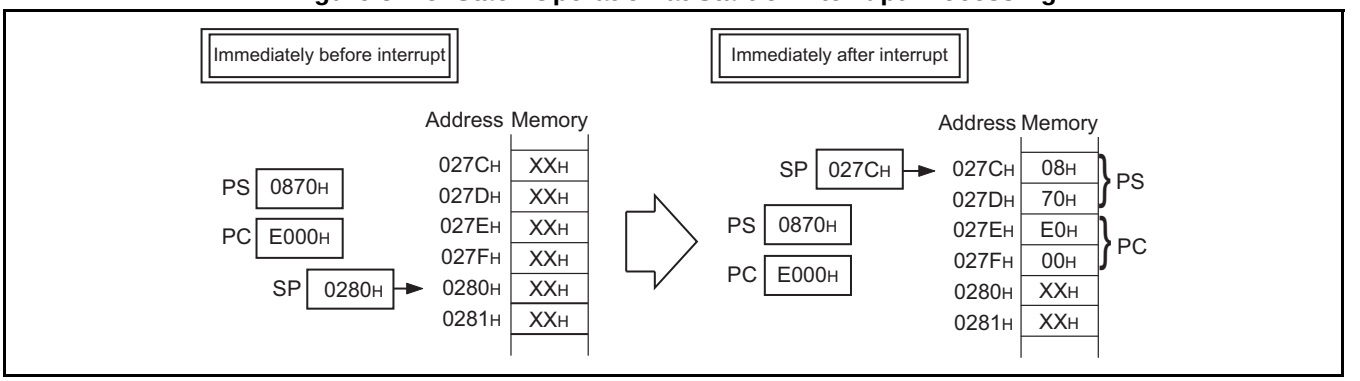
This section describes how the contents of a register are saved and restored during interrupt processing.

■ Stack Operation at the Start of Interrupt Processing

Once the CPU accepts an interrupt, it automatically saves the current value of the program counter (PC) and that of the program status (PS) values to the stack.

Figure 8.1-5 shows the stack operation at the start of interrupt processing.

Figure 8.1-5 Stack Operation at Start of Interrupt Processing



■ Stack Operation after Returning from Interrupt

When the CPU executes the interrupt return instruction (RETI) at the end of interrupt processing, it restores from the stack the value of the program status (PS) first and that of the program counter (PC), which is opposite to the sequence of saving the two values to the stack. After the restoration, both PS and PC return to their states prior to the start of interrupt processing.

Note:

Since the value of the accumulator (A) and that of the temporary accumulator (T) are not automatically saved to the stack, use the PUSHW and POPW instructions to save and restore the values of A and T.

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.

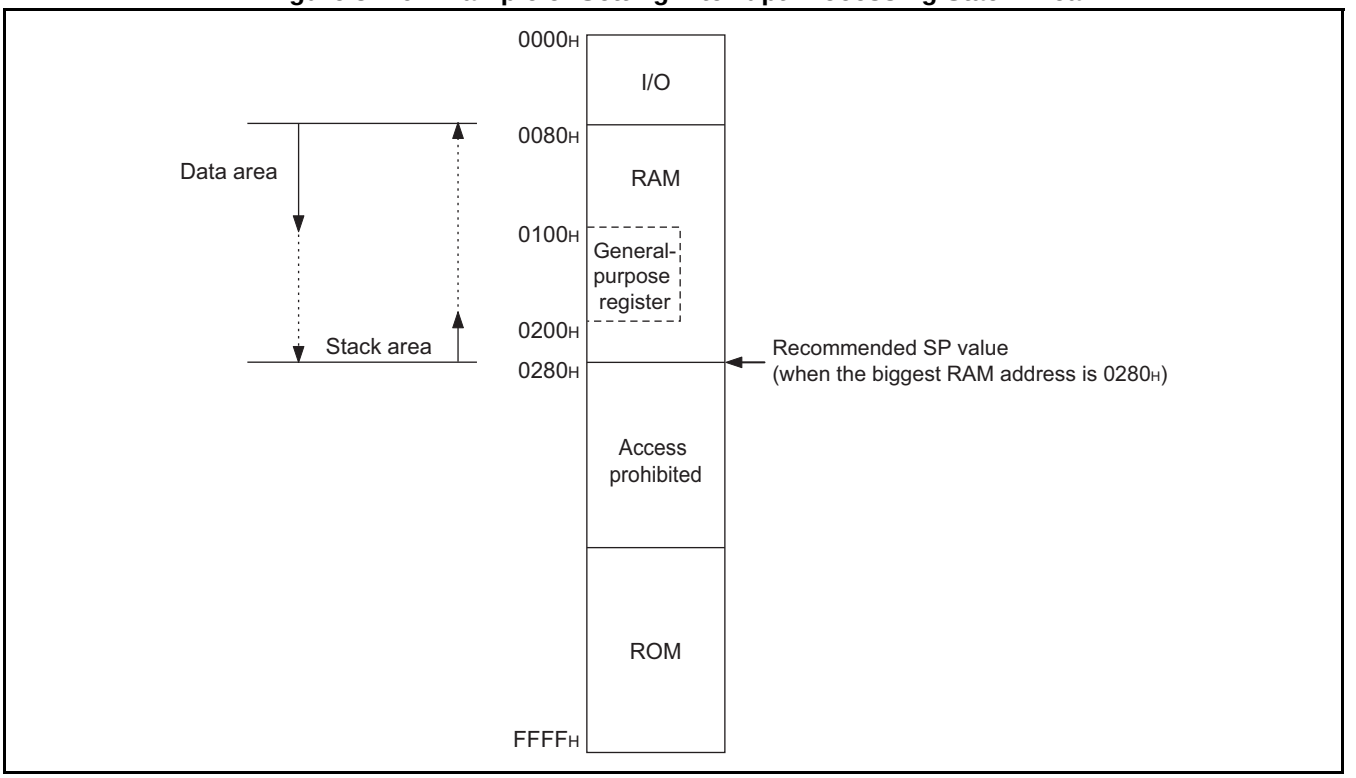
■ Interrupt Processing Stack Area

The stack area is also used for saving and restoring the program counter (PC) when the subroutine call instruction (CALL) or the vector call instruction (CALLV) is executed, and for saving temporarily and restoring register contents by the PUSHW and POPW instructions.

- The stack area is secured on the RAM together with the data area.
- Initialize the stack pointer (SP) so that it indicates the biggest RAM address and make the data area start from the smallest RAM address.

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

Figure 8.1-6 Example of Setting Interrupt Processing Stack Area



Note:

The stack area is utilized by interrupts, sub-routine calls, the PUSHW instruction, etc. in descending of addresses. It is released by return instructions (RETI, RET), the POPW instruction, etc. in ascending order of addresses. If the address value of the stack area used decreases due to nested interrupts or subroutine calls, do not let the stack area overlap the data area and the general-register area, both of which retain other data.

CHAPTER 9
I/O PORTS

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 capture input signals into the CPU with the port data register (PDR). The I/O direction of an individual I/O pin can be set as desired by using the corresponding to that I/O pin in the port direction register (DDR).

Table 9.1-1 and list the registers for each port.

Table 9.1-1 List of Port Registers (MB95200H Series)

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 F data register	(PDRF)	R, RM/W	00000000_B
Port F direction register	(DDRF)	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 register	(PUL0)	R/W	00000000_B
Port G pull-up register	(PULG)	R/W	00000000_B
A/D input disable register (Lower)	(AIDRL)	R/W	00000000_B
Input level select register	(ILSR)	R/W	00000000_B

Table 9.1-2 List of Port Registers (MB95210H Series

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 F data register	(PDRF)	R, RM/W	00000000_B
Port F direction register	(DDRF)	R/W	00000000_B
Port 0 pull-up register	(PUL0)	R/W	00000000_B
A/D input disable register (Lower)	(AIDRL)	R/W	00000000_B
Input level select register	(ILSR)	R/W	00000000_B

R/W: Readable/writable (The read value is the same as the write value.)

R, RM/W: Readable/writable (The read value is different from the write value. The write value is read by the read-modify-write type of instruction.)

9.2 Port 0

Port 0 is a general-purpose I/O port.
This section focuses on its functions as a general-purpose I/O port.
For details of peripheral functions, see their respective chapters.

■ 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 register (PUL0)
- A/D input disable register lower (AIDRL)
- Input level select register (ILSR)

■ Port 0 Pins

Port 0 has eight I/O pins.

Table 9.2-1 and Table 9.2-2 list the port 0 pins.

Table 9.2-1 Port 0 Pins (MB95200H Series)

Pin name	Function	Shared peripheral function	I/O type			
			Input*	Output	OD	PU
P00/AN00	P00 general-purpose I/O	AN00 analog input	Hysteresis/ Analog	CMOS	-	❍
P01/AN01	P01 general-purpose I/O	AN01 analog input	Hysteresis/ Analog	CMOS	-	❍
P02/INT02/ AN02/SCK	P02 general-purpose I/O	AN02 analog input INT02 external interrupt input LIN-UART clock I/O	Hysteresis/ Analog	CMOS	-	❍
P03/INT03/ AN03/SOT	P03 general-purpose I/O	AN03 analog input INT03 external interrupt input LIN-UART data output	Hysteresis/ Analog	CMOS	-	❍
P04/INT04/ AN04/SIN/ EC0/HCLK1*1	P04 general-purpose I/O	AN04 analog input INT04 external interrupt input 8/16-bit composite timer ch.0 clock input External clock input LIN-UART data input	Hysteresis/CMOS/ Analog	CMOS	-	❍
P05/INT05/ AN05/TO00/ HCLK2*2	P05 general-purpose I/O	AN05 analog input INT05 external interrupt input External clock input 8/16-bit composite timer 0 ch.0 output	Hysteresis/ Analog	CMOS	-	❍
P06/INT06/ TO01	P06 general-purpose I/O	INT06 external interrupt input 8/16-bit composite timer 0 ch.1 output	Hysteresis	CMOS	-	❍
P07/INT07	P07 general-purpose I/O	INT07 external interrupt input	Hysteresis	CMOS	-	❍

Table 9.2-2 Port 0 Pins (MB95210H Series)

Pin name	Function	Shared peripheral function	I/O type			
			Input*	Output	OD	PU
P04/INT04/ AN04/EC0/ HCLK1*1	P04 general-purpose I/O	AN04 analog input	Hysteresis/CMOS/ Analog	CMOS	-	❍
		INT04 external interrupt input				
		8/16-bit composite timer ch.0 clock input				
		External clock input				
P05/AN05/ TO00/ HCLK2*2	P05 general-purpose I/O	AN05 analog input	Hysteresis/ Analog	CMOS	-	❍
		External clock input				
		8/16-bit composite timer 0 ch.0 output				
P06/INT06/ TO01	P06 general-purpose I/O	INT06 external interrupt input	Hysteresis	CMOS	-	❍
		8/16-bit composite timer 0 ch.1 output				

OD: Open drain, PU: Pull-up

*1: If the external clock input is selected (SYSC:EXCK[1:0]=01_B), other functions cannot be selected.

*2: If the external clock input is selected (SYSC:EXCK[1:0]=10_B), other functions cannot be selected.

■ Block Diagrams of Port 0

Figure 9.2-1 Block Diagram of P00 and P01

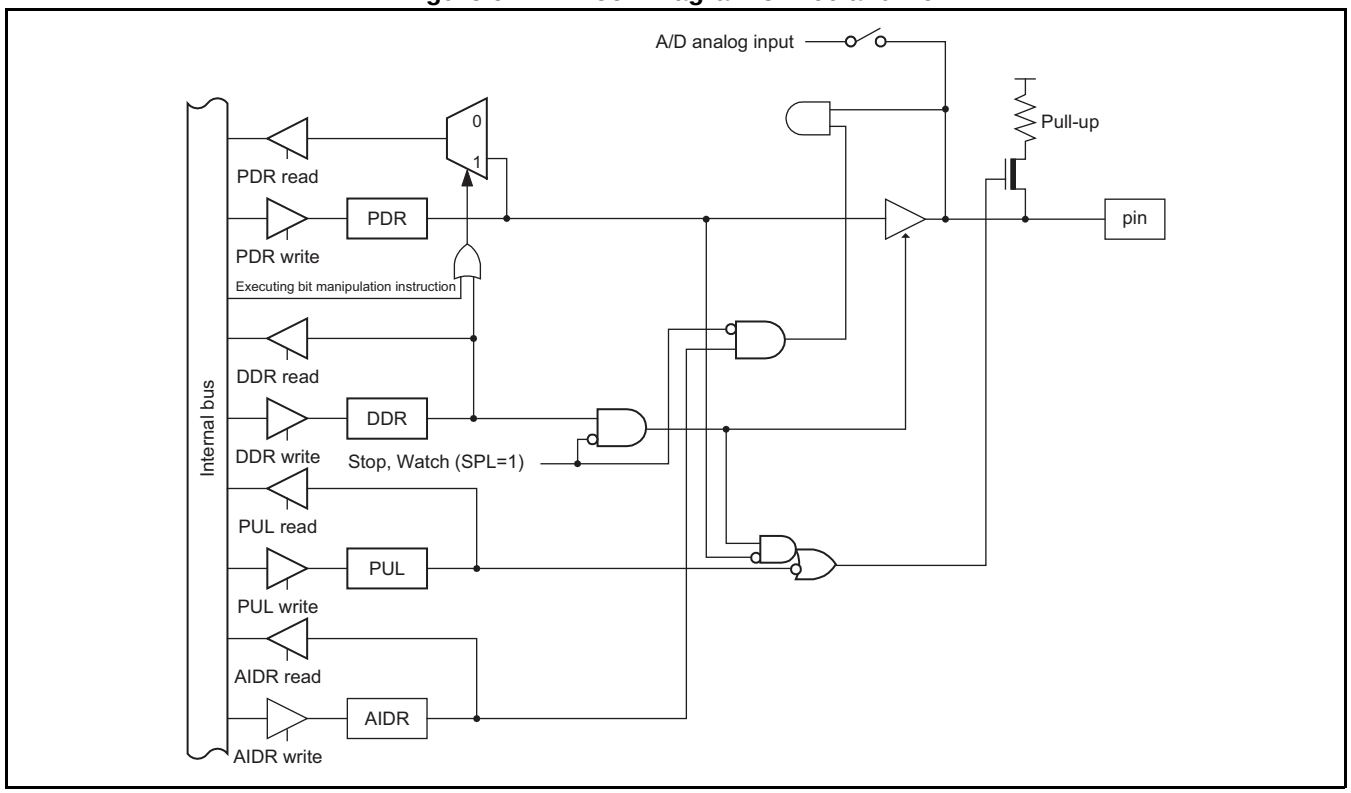


Figure 9.2-2 Block Diagram of P02, P03 and P05

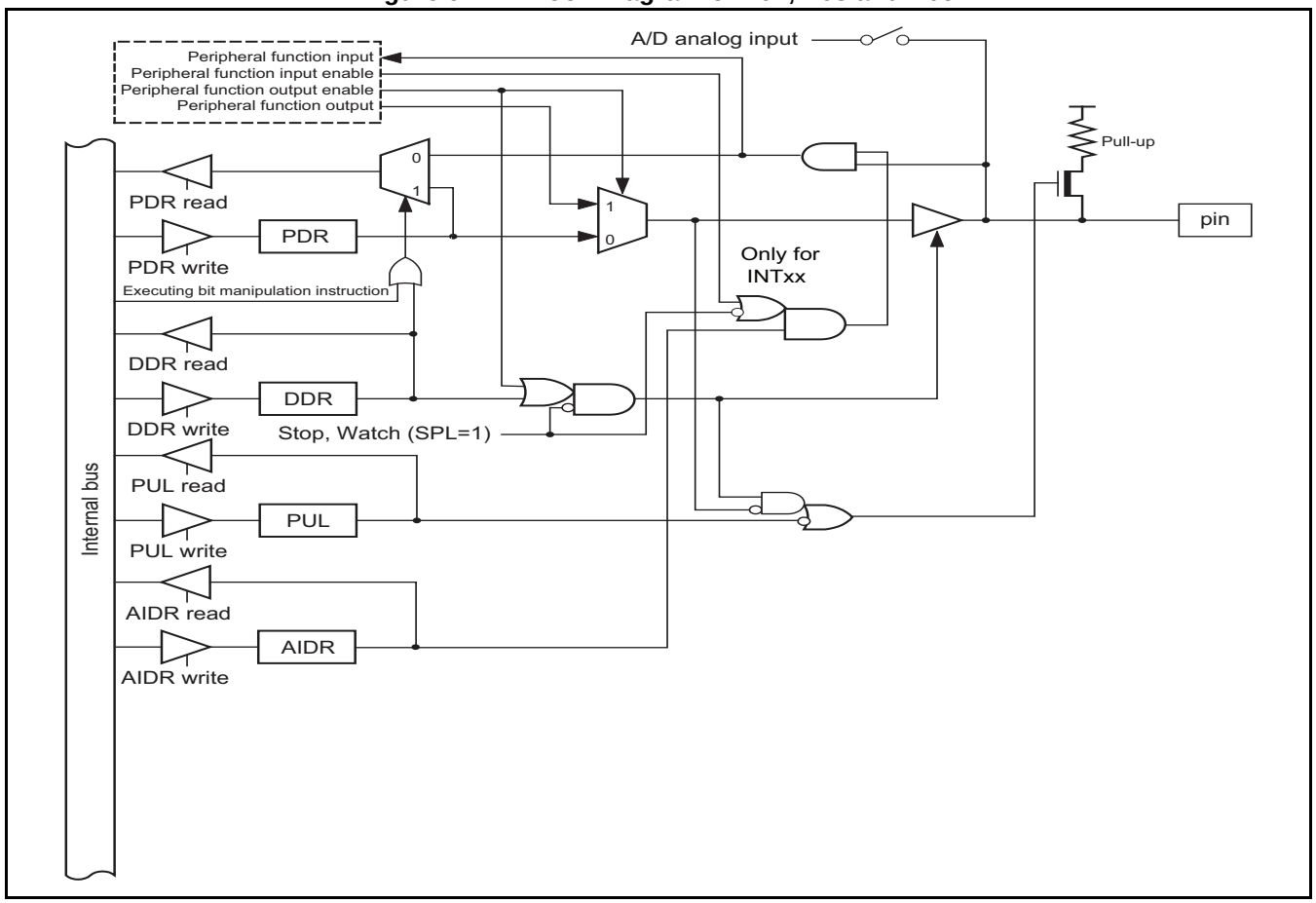


Figure 9.2-3 Block Diagram of P04

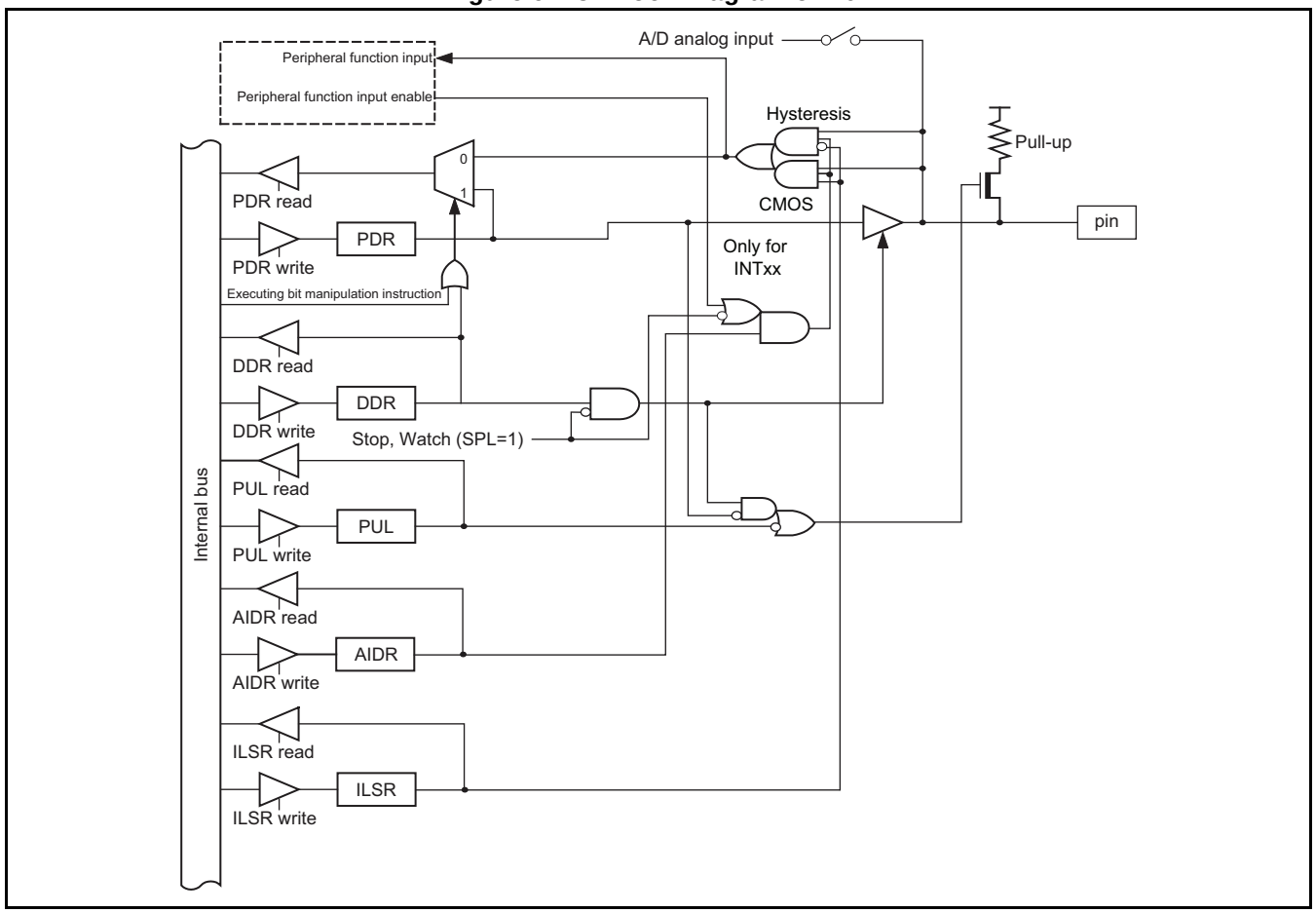


Figure 9.2-4 Block Diagram of P06

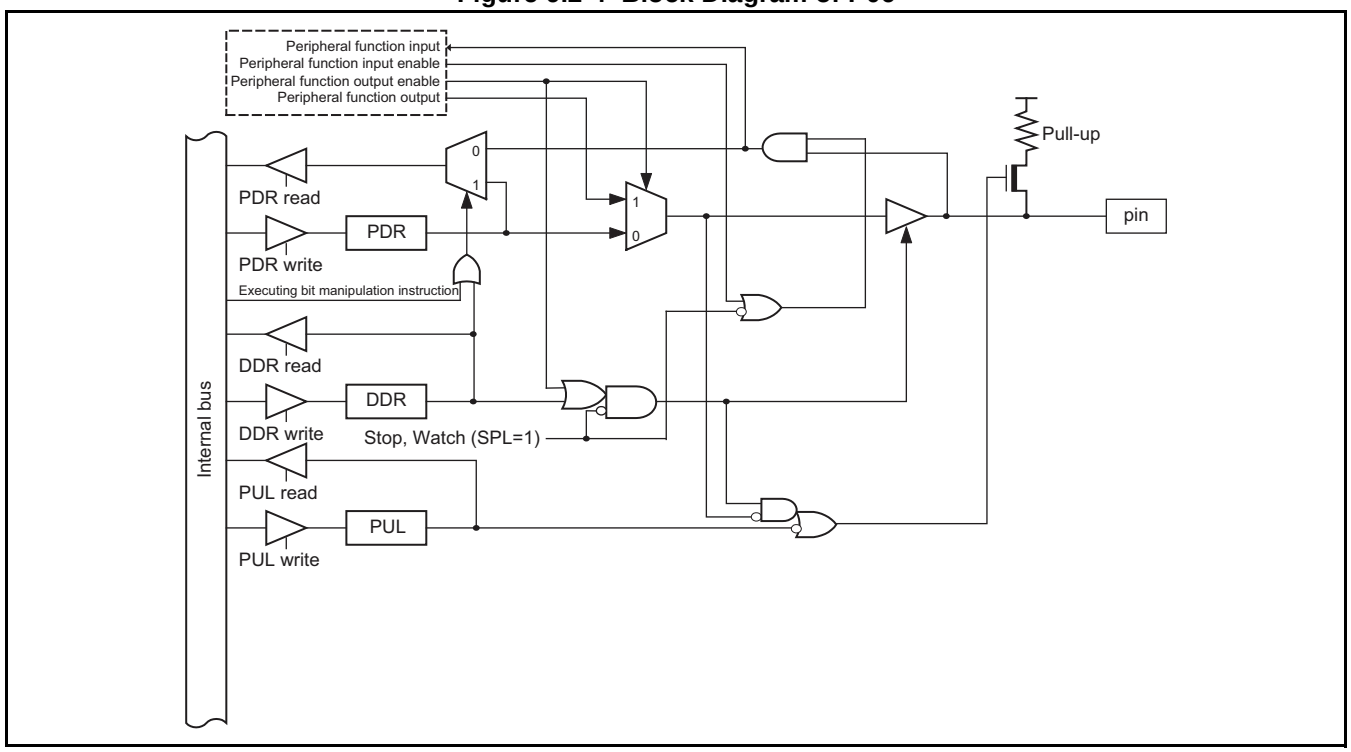
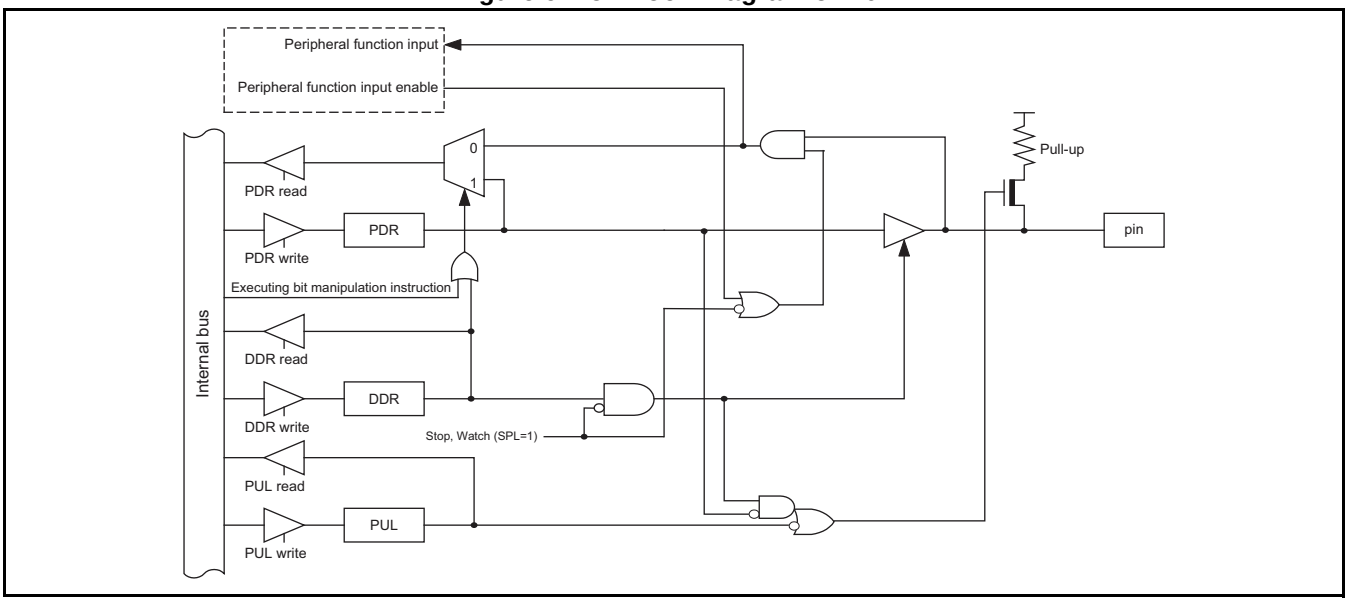


Figure 9.2-5 Block Diagram of P07



9.2.1 Port 0 Registers

This section describes the registers of port 0.

■ Port 0 Register Functions

Table 9.2-3 lists the functions of the port 0 register.

Table 9.2-3 Port 0 Register Functions

Register name	Data	Read	Read by read-modify-write instruction	Write
PDR0	0	Pin state is "L" level.	PDR value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR value is "1".	As output port, outputs "H" level.
DDR0	0	Port input enabled		
	1	Port output enabled		
PUL0	0	Pull-up disabled		
	1	Pull-up enabled		
AIDRL	0	Analog input enabled		
	1	Port input enabled		
ILSR	0	Hysteresis input level selected		
	1	CMOS input level selected		

Table 9.2-4 and Table 9.2-5 list the correspondence between port 0 pins and each register bit.

Table 9.2-4 Correspondence between Registers and Pins for Port 0 (MB95200H Series)

	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	-	-						
ILSR	-	-	-	bit2	-	-	-	-

Table 9.2-5 Correspondence between Registers and Pins for Port 0 (MB95210H Series)

	Correspondence between related register bits and pins		
Pin name	P06	P05	P04
PDR0	bit6	bit5	bit4
DDR0			
PUL0			
AIDRL	-		
ILSR	-	-	bit2

9.2.2 Operations of Port 0

This section describes the operations of port 0.

■ Operations of port 0

● Operation as an output port

- A pin will become an output port if the bit in the DDR register corresponding to that pin is set to "1".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When a pin is used as an output port, it outputs the value of the PDR register to external pins.
- If data is written to the PDR register, the value is stored in the output latch and is output to the pin set as an output port as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When using an analog input shared pin as an input port, set the corresponding bit in the A/D input disable register lower (AIDRL) to "1".
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function output pin

- A pin will become a peripheral function output pin if the peripheral output function is enabled by setting the output enable bit of a peripheral function corresponding to that pin.
- 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 the PDR register. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function input pin

- To set a pin as an input port, set the bit in the DDR register bit corresponding to the input pin of a peripheral function to "0".
- When using the analog input shared pin as another peripheral function input pin, configure it as an input port, which is the same as the operation as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses that pin as its input pin. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation at reset

- If the CPU is reset, all bits in the DDR register are initialized to "0" and port input is enabled. As for a pin shared with analog input, its port input is disabled because the A/D input disable register lower (AIDRL) is initialized to "0".

● Operation in stop mode and watch mode

- If the pin state setting bit in the standby control register (STBC:SPL) is set to "1" and the device transits to stop mode or watch mode, the pin is compulsorily made to enter the high impedance state regardless of the DDR register value. The input of that pin is locked to "L" level and blocked in order to prevent leaks due to input open. However, if the interrupt input is enabled for the external interrupt (INT07 to INT02), the input is enabled and not blocked.
- If the pin state setting bit is "0", the state of the port I/O or that of the peripheral function I/O remains unchanged and the output level is maintained.

● Operation as an analog input pin

- Set the bit in the DDR register bit corresponding to the analog input pin to "0" and the bit corresponding to that pin in the AIDRL register to "0".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions. In addition, set the corresponding bit in the PUL register to "0".

● Operation as an external interrupt input pin

- Set the bit in the DDR register corresponding to the external interrupt input pin to "0".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- The pin value is always input to the external interrupt circuit. When using a pin for a function other than the interrupt, disable the external interrupt function corresponding to that pin.

● Operation of the pull-up control register

- Setting the bit in the PUL register to "1" makes the pull-up resistor be internally connected the pull-up register to the pin. When the pin output is "L" level, the pull-up resistor is disconnected regardless of the value of the PUL register.

● Operation of the input level select register

- Setting the bit in ILSR to "1" changes only P04 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P04 should become the hysteresis input level.
- For pins other than P04, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.
- When changing the input level of P04, ensure that the peripheral function (LIN-UART/External interrupt/ 8/16-bit composite timer ch.0 clock input / External clock input) has been stopped.

Table 9.2-6 shows the pin states of port 0.

Table 9.2-6 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 (the pull-up setting is enabled) Input cutoff (If the external interrupt function is enabled, the external interrupt can be input.)	Hi-Z Input disabled*

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

Hi-Z: High impedance

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

9.3 Port 1

Port 1 is a general-purpose I/O port.
This section focuses on its functions as a general-purpose I/O port.
For details of peripheral functions, see their respective chapters.

■ 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 Pin

Port 1 has one I/O pin.

Table 9.3-1 lists the port 1 pin.

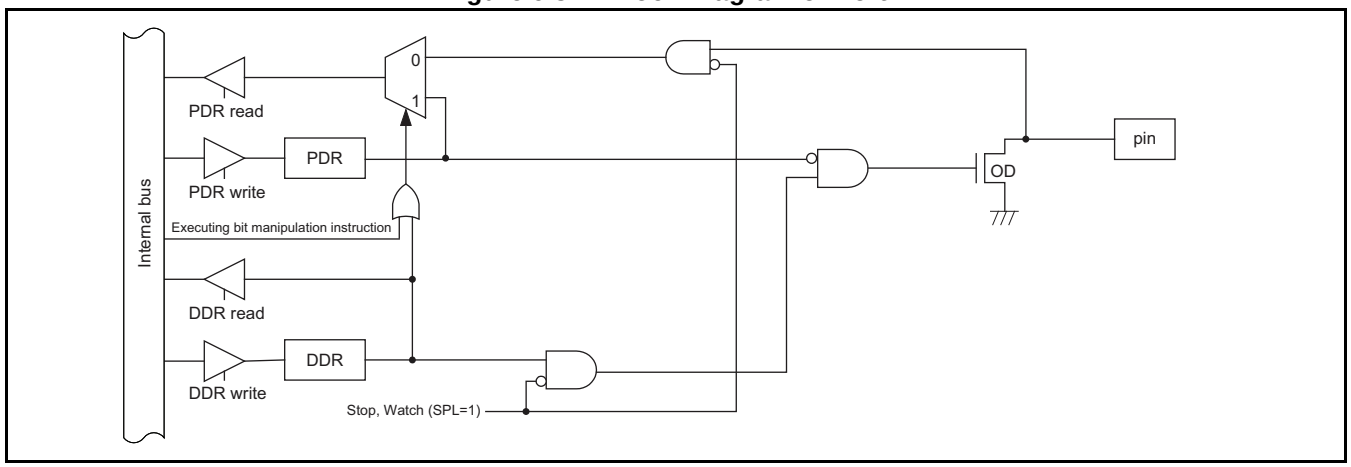
Table 9.3-1 Port 1 Pin

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P12/ DBG/EC0	P12 general-purpose I/O	On-chip debug communication pin	Hysteresis	CMOS	❍	-
		EC0 8/16-bit composite timer ch.0 clock input				

OD: Open drain, PU: Pull-up

■ Block Diagram of Port 1

Figure 9.3-1 Block Diagram of Port 1



9.3.1 Port 1 Registers

This section describes the registers of port 1.

■ Port 1 Register Functions

Table 9.3-2 lists the port 1 register functions.

Table 9.3-2 Port 1 Register Functions

Register name	Data	Read	Read by read-modify-write instruction	Write
PDR1	0	Pin state is "L" level.	PDR value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR value is "1".	As output port, outputs "H" level.
DDR1	0	Port input enabled		
	1	Port output enabled		

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	-	-	-	-	-	P12	-	-
PDR1	-	-	-	-	-	bit2	-	-
DDR1								

9.3.2 Operations of Port 1

This section describes the operations of port 1.

■ Operations of Port 1

● Operation as an output port

- A pin will become an output port if the bit in the DDR register corresponding to that pin is set to "1".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When a pin is used as an output port, it outputs the value of the PDR register to external pins.
- If data is written to the PDR register, the value is stored in the output latch and is output to the pin set as an output port as it is.
- Reading the PDR register returns the PDR register value.

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function output pin

- A pin will become a peripheral function output pin if the peripheral output function is enabled by setting the output enable bit of a peripheral function corresponding to that pin.
- 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 the PDR register. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function input pin

- To set a pin as an input port, set the bit in the DDR register corresponding to the input pin of a peripheral function to "0".
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses that pin as its input pin. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation at reset

- If the CPU is reset, all bits in the DDR register are initialized to "0" and port input is enabled.

● Operation in stop mode and watch mode

- If the pin state setting bit in the standby control register (STBC:SPL) is set to "1" and the device transits to stop mode or watch mode, the pin is compulsorily made to enter the high impedance state regardless of the DDR register value. The input of that pin is locked to "L" level and blocked in order to prevent leaks due to input open.
- If the pin state setting bit is "0", the state of the port I/O or that of the peripheral function I/O remains unchanged and the output level is maintained.

Table 9.3-4 shows the pin states of 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 Input cutoff	Hi-Z Input enabled* (Not functional)

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

Hi-Z: High impedance

*: "Input enabled" means that the input function is enabled. While the input function is enabled, prevent leak by pulling up the pin or pulling down the pin or supplying voltage from an external component to the pin, etc. If a pin is used as an output port, the pin state is the same as that of other ports.

9.4 Port 6

Port 6 is a general-purpose I/O port.
This section focuses on its functions as a general-purpose I/O port.
For details of peripheral functions, see their respective chapters.
This port is only available in the MB95200H Series.

■ 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)

■ Port 6 Pins

Port 6 has three I/O pins.

Table 9.4-1 lists the port 6 pins.

Table 9.4-1 Port 6 Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P62/TO10	P62 general-purpose I/O	TO10 8/16-bit composite timer 10 output	Hysteresis	CMOS	-	-
P63/TO11	P63 general-purpose I/O	TO11 8/16-bit composite timer 11 output	Hysteresis	CMOS	-	-
P64/EC1	P64 general-purpose I/O	EC1 8/16-bit composite timer ch.1 clock output	Hysteresis	CMOS	-	-

OD: Open drain, PU: Pull-up

■ Block Diagrams of Port 6

Figure 9.4-1 Block Diagram of P62 and P63

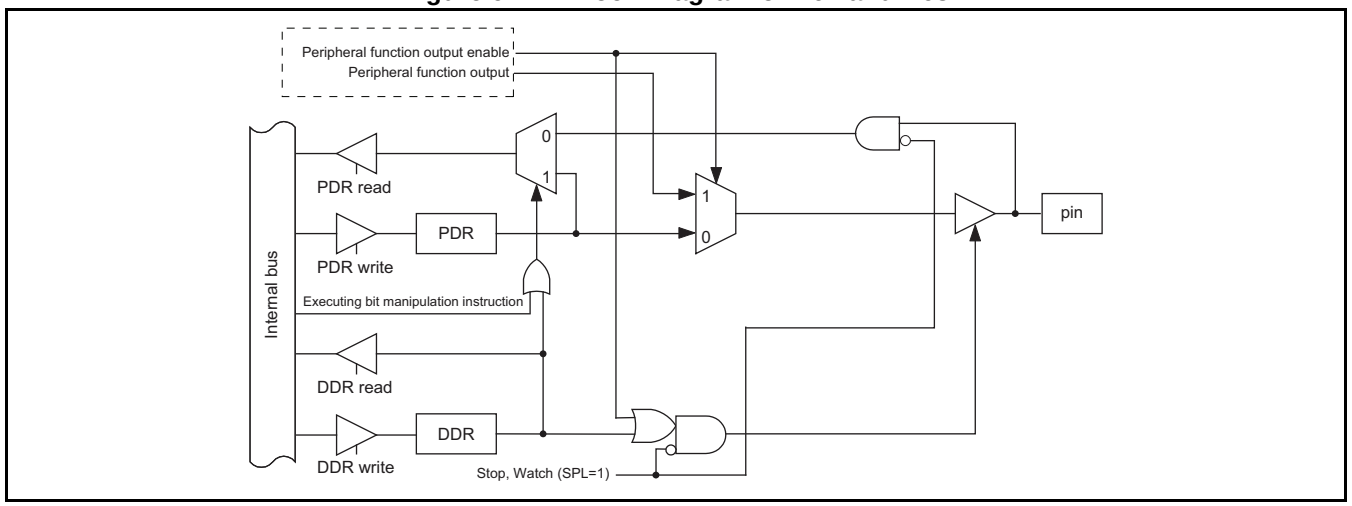
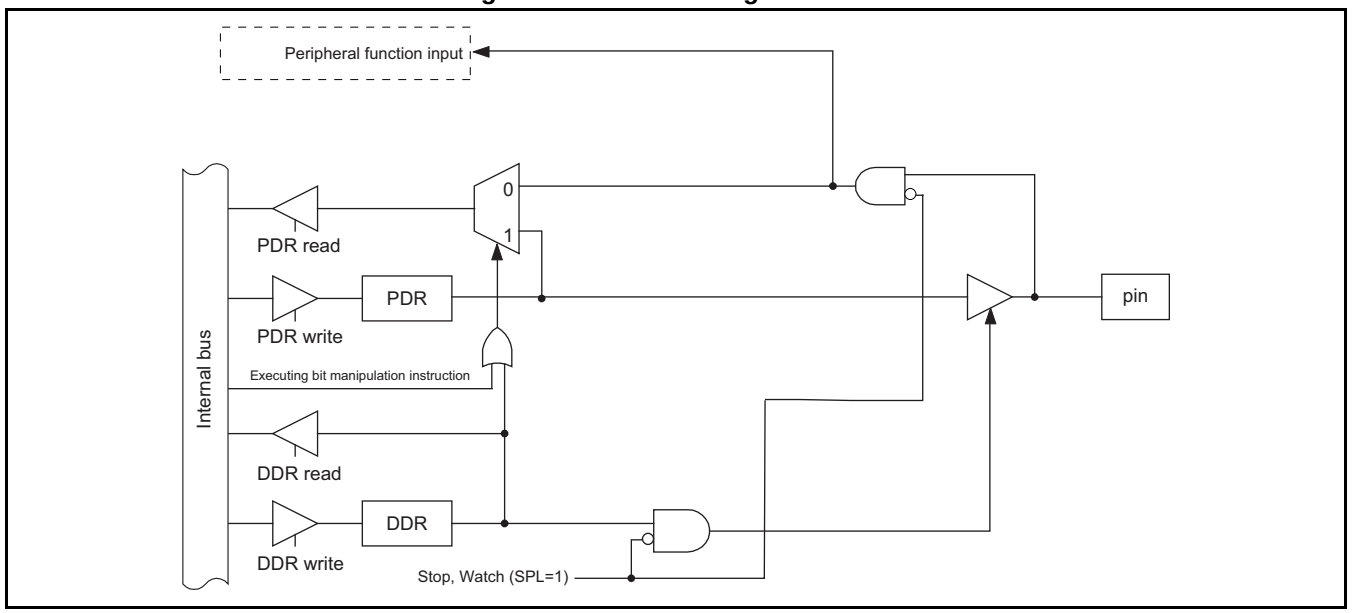


Figure 9.4-2 Block Diagram of P64



9.4.1 Port 6 Registers

This section describes the registers of port 6.

■ Port 6 Register Functions

Table 9.4-2 lists the port 6 register functions.

Table 9.4-2 Port 6 Register Functions

Register name	Data	Read	Read by read-modify-write instruction	Write
PDR6	0	Pin state is "L" level.	PDR value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR value is "1".	As output port, outputs "H" level*.
DDR6	0	Port input enabled		
	1	Port output enabled		

*: For N-ch. open drain pin, this should be Hi-Z.

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	-	-	-	P64	P63	P62	-	-
PDR6	-	-	-	bit4	bit3	bit2	-	-
DDR6								

9.4.2 Operations of Port 6

This section describes the operations of port 6.

■ Operations of Port 6

● Operation as an output port

- A pin will become an output port if the bit in the DDR register corresponding to that pin is set to "1".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When a pin is used as an output port, it outputs the value of the PDR register to external pins.
- If data is written to the PDR register, the value is stored in the output latch and is output to the pin set as an output port as it is.
- Reading the PDR register returns the PDR value.

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function output pin

- A pin will become a peripheral function output pin if the peripheral output function is enabled by setting the output enable bit of a peripheral function corresponding to that pin.
- 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 the PDR register. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation as a peripheral function input pin

- To set a pin as an input port, set the bit in the DDR register corresponding to the input pin of a peripheral function to "0".
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses that pin as its input pin. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation at reset

- If the CPU is reset, all bits in the DDR register are initialized to "0" and port input is enabled.

● Operation in stop mode and watch mode

- If the pin state setting bit in the standby control register (STBC:SPL) is set to "1" and the device transits to stop mode or watch mode, the pin is compulsorily made to enter the high impedance state regardless of the DDR register value. The input of that pin is locked to "L" level and blocked in order to prevent leaks due to input open.

- If the pin state setting bit is "0", the state of the port I/O or that of the peripheral function I/O remains unchanged and the output level is maintained.

Table 9.4-4 shows the pin states of 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 setting bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is enabled. While the input function is enabled, pull-up or pull-down operation has to be performed in order to prevent leaks due to external input. If a pin is used as an output port, the pin state is the same as that of other ports.

9.5 Port F

Port F is a general-purpose I/O port.
This section focuses on its functions as a general-purpose I/O port.
For details of peripheral functions, see their respective chapters.

■ Port F Configuration

Port F is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port F data register (PDRF)
- Port F direction register (DDRF)

■ Port F Pins

Port F has three I/O pins.

Table 9.5-1 and Table 9.5-2 list the port F pins.

Table 9.5-1 Port F Pins (MB95200H Series)

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PF0/X0*1	PF0 general-purpose I/O	Main clock oscillation pin	Hysteresis	CMOS	-	-
PF1/X1*1	PF1 general-purpose I/O	Main clock oscillation pin	Hysteresis	CMOS	-	-
PF2/RSTX*2	PF2 general-purpose I/O	External reset pin	Hysteresis	CMOS	-	-

Table 9.5-2 Port F Pins (MB95210H Series)

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PF2/RSTX*2	PF2 general-purpose I/O	External reset pin	Hysteresis	CMOS	-	-

OD: Open drain, PU: Pull-up

*1: If the main oscillation clock is selected (SYSC:PFSEL=0_B), the port function cannot be used.

*2: If the external reset is selected (SYSC:RSTEN=1_B), the port function cannot be used.

This pin is a dedicated reset pin in MB95F204H/F214H/F203H/F213H/F202H/F212H.

■ Block Diagrams of Port F

Figure 9.5-1 Block Diagram of PF0 and PF1

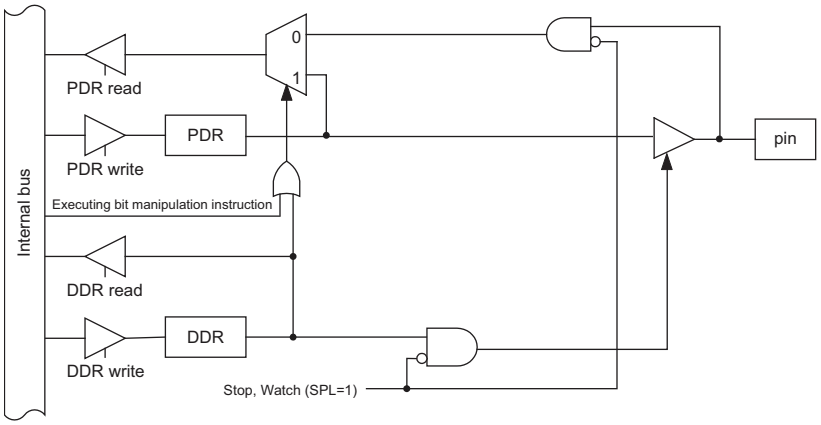
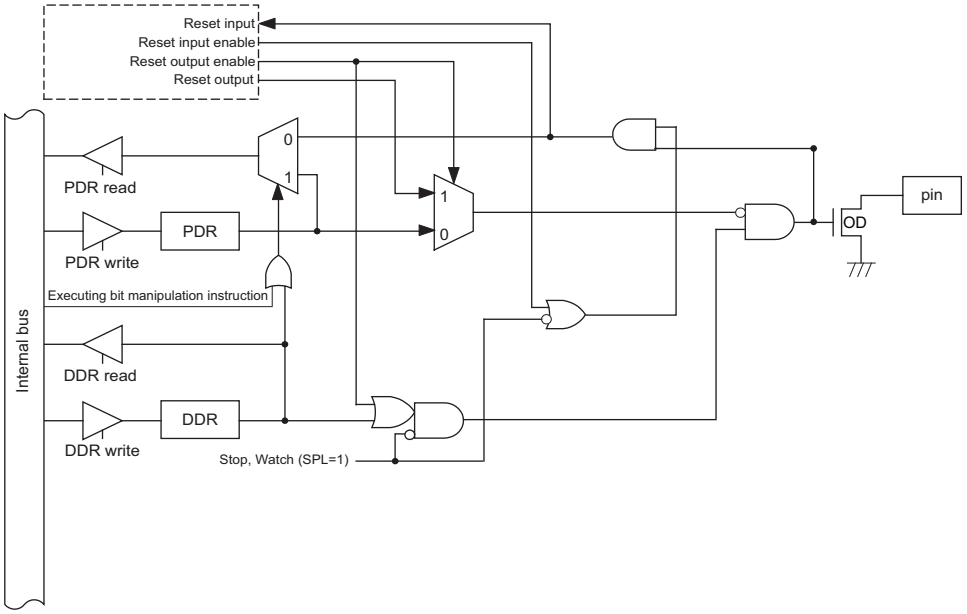


Figure 9.5-2 Block Diagram of PF2



9.5.1 Port F Registers

This section describes the registers of port F.

■ Port F Register Functions

Table 9.5-3 lists the port F register functions.

Table 9.5-3 Port F Register Functions

Register name	Data	Read	Read by read-modify-write instruction	Write
PDRF	0	Pin state is "L" level.	PDR value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR value is "1".	As output port, outputs "H" level.
DDRF	0	Port input enabled		
	1	Port output enabled		

Table 9.5-4 and Table 9.5-5 list the correspondence between port F pins and each register bit.

Table 9.5-4 Correspondence between Registers and Pins for Port F (MB95200H Series)

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	-	PF2	PF1	PF0
PDRF	-	-	-	-	-	bit2	bit1	bit0
DDRF								

Table 9.5-5 Correspondence between Registers and Pins for Port F (MB95210H Series)

	Correspondence between related register bits and pins					
Pin name	-	-	-	-	-	PF2
PDRF	-	-	-	-	-	bit2
DDRF						

*: PF2/RSTX is a dedicated reset pin in MB95F204H/F214H/F203H/F213H/F202H/F212H.

9.5.2 Operations of Port F

This section describes the operations of port F.

■ Operations of Port F

● Operation as an output port

- A pin will become an output port if the bit in the DDR register corresponding to that pin is set to "1".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When a pin is used as an output port, it outputs the value of the PDR register to external pins.
- If data is written to the PDR register, the value is stored in the output latch and is output to the pin set as an output port as it is.
- Reading the PDR register returns the PDR value.

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation at reset

- If the CPU is reset, all bits in the DDR register are initialized to "0" and port input is enabled.

● Operation in stop mode and watch mode

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

Table 9.5-6 shows the pin states of port F.

Table 9.5-6 Pin State of Port F

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*1 (Not functional) Low*2

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

Hi-Z: High impedance

*1: "Input enabled" means that the input function is enabled. While the input function is enabled, pull-up or pull-down operation has to be performed in order to prevent leaks due to external input. If a pin is used as an output port, the pin state is the same as that of other ports.

*2: Only for PF2 at power-on reset.

9.6 Port G

Port G is a general-purpose I/O port.
This section focuses on its functions as a general-purpose I/O port.
For details of peripheral functions, see their respective chapters.
This port is only available in the MB95200H series.

■ 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 register (PULG)

■ Port G Pins

Port G has two I/O pins.

Table 9.6-1 lists the port G pins.

Table 9.6-1 Port G Pins

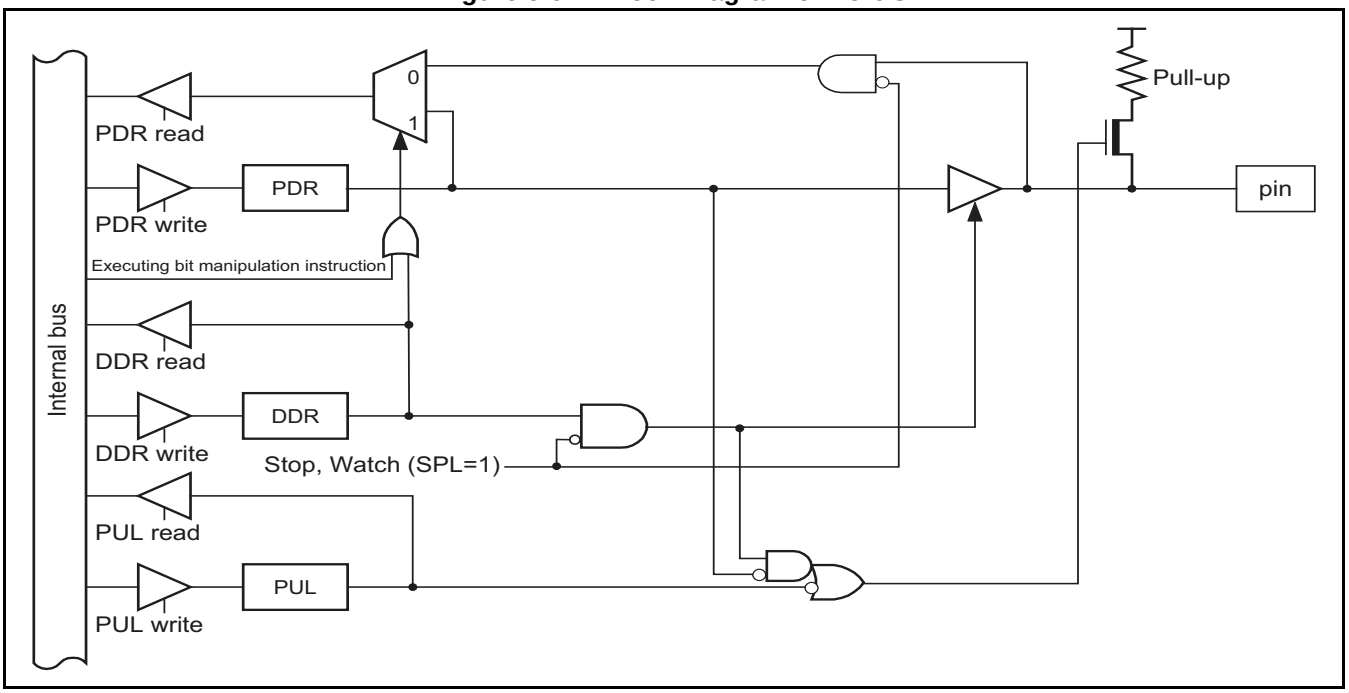
Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PG1/X0A*1	PG1 general-purpose I/O	Subclock oscillation pin	Hysteresis	CMOS	-	❍
PG2/X1A*1	PG2 general-purpose I/O	Subclock oscillation pin	Hysteresis	CMOS	-	❍

OD: Open drain, PU: Pull-up

*1: If the sub-oscillation clock is selected (SYSC:PGSEL=0_B), the port function cannot be used.

■ Block Diagram of Port G

Figure 9.6-1 Block Diagram of Port G



9.6.1 Port G Registers

This section describes the registers of port G.

■ Port G Register Functions

Table 9.6-2 lists the port G register functions.

Table 9.6-2 Port G Register Functions

Register name	Data	Read	Read by read-modify-write instruction	Write
PDRG	0	Pin state is "L" level.	PDR value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR 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 9.6-3 lists the correspondence between port G pins and each register bit.

Table 9.6-3 Correspondence between Registers and Pins for Port G

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	-	PG2	PG1	-
PDRG	-	-	-	-	-	bit2	bit1	-
DDRG								
PULG								

9.6.2 Operations of Port G

This section describes the operations of port G.

■ Operations of Port G

● Operation as an output port

- A pin will become an output port if the bit in the DDR register corresponding to that pin is set to "1".
- For a pin shared with other peripheral functions, disable the output of such peripheral functions.
- When a pin is used as an output port, it outputs the value of the PDR register to external pins.
- If data is written to the PDR register, the value is stored in the output latch and is output to the pin set as an output port as it is.
- Reading the PDR register returns the PDR value.

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write command is used to read the PDR register, the PDR register value is returned.

● Operation at reset

- If the CPU is reset, all bits in the DDR register are initialized to "0" and port input is enabled.

● Operation in stop mode and watch mode

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

● Operation of the pull-up register

- Setting the bit in the PUL register to "1" makes the pull-up resistor be internally connected to the pin. When the pin output is "L" level, the pull-up resistor is disconnected regardless of the value of the PUL register.

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

Table 9.6-4 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 setting bit in standby control register (STBC:SPL)

Hi-Z: High impedance

*: "Input enabled" means that the input function is enabled. While the input function is enabled, pull-up or pull-down operation has to be performed in order to prevent leaks due to external input. If a pin is used as an output port, the pin state is the same as that of other ports.

CHAPTER 10

TIMEBASE TIMER

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

10.1 Overview of Timebase Timer

The timebase timer is a 24-bit free-run down-counting counter. It is synchronized with the main clock divided by two or with the main CR clock. The clock can be selected by the RCM1 bit and RCM0 bit in the SYCC2 register. The timebase timer has an interval timer function that 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 or using the main CR clock as the count clock.

- The counter of the timebase timer counts down so that an interrupt request is generated every time a selected interval time elapses.
- The length of an interval time can be selected from the following 16 types.

Table 10.1-1 shows the interval times available for the timebase timer.

Table 10.1-1 Interval Times of Timebase Timer

	Interval time if the main CR clock is used ($2^n \times 1/F_{CRH}$*1)	Interval time if the main clock is used ($2^n \times 2/F_{CH}$*2)
n=9	64 [µs]	256 [µs]
n=10	128 [µs]	512 [µs]
n=11	256 [µs]	1.024 [ms]
n=12	512 [µs]	2.048 [ms]
n=13	1.024 [ms]	4.096 [ms]
n=14	2.048 [ms]	8.192 [ms]
n=15	4.096 [ms]	16.384 [ms]
n=16	8.192 [ms]	32.768 [ms]
n=17	16.384 [ms]	65.536 [ms]
n=18	32.768 [ms]	131.072 [ms]
n=19	65.536 [ms]	262.144 [ms]
n=20	131.072 [ms]	524.288 [ms]
n=21	262.144 [ms]	1.049 [s]
n=22	524.288 [ms]	2.097 [s]
n=23	1.049 [s]	4.194 [s]
n=24	2.097 [s]	8.389 [s]

*1: $1/F_{CRH}$ = 0.125 [µs] when F_{CRH} = 8 [MHz]
*2: $2/F_{CH}$ = 0.5 [µs] when F_{CH} = 4 [MHz]

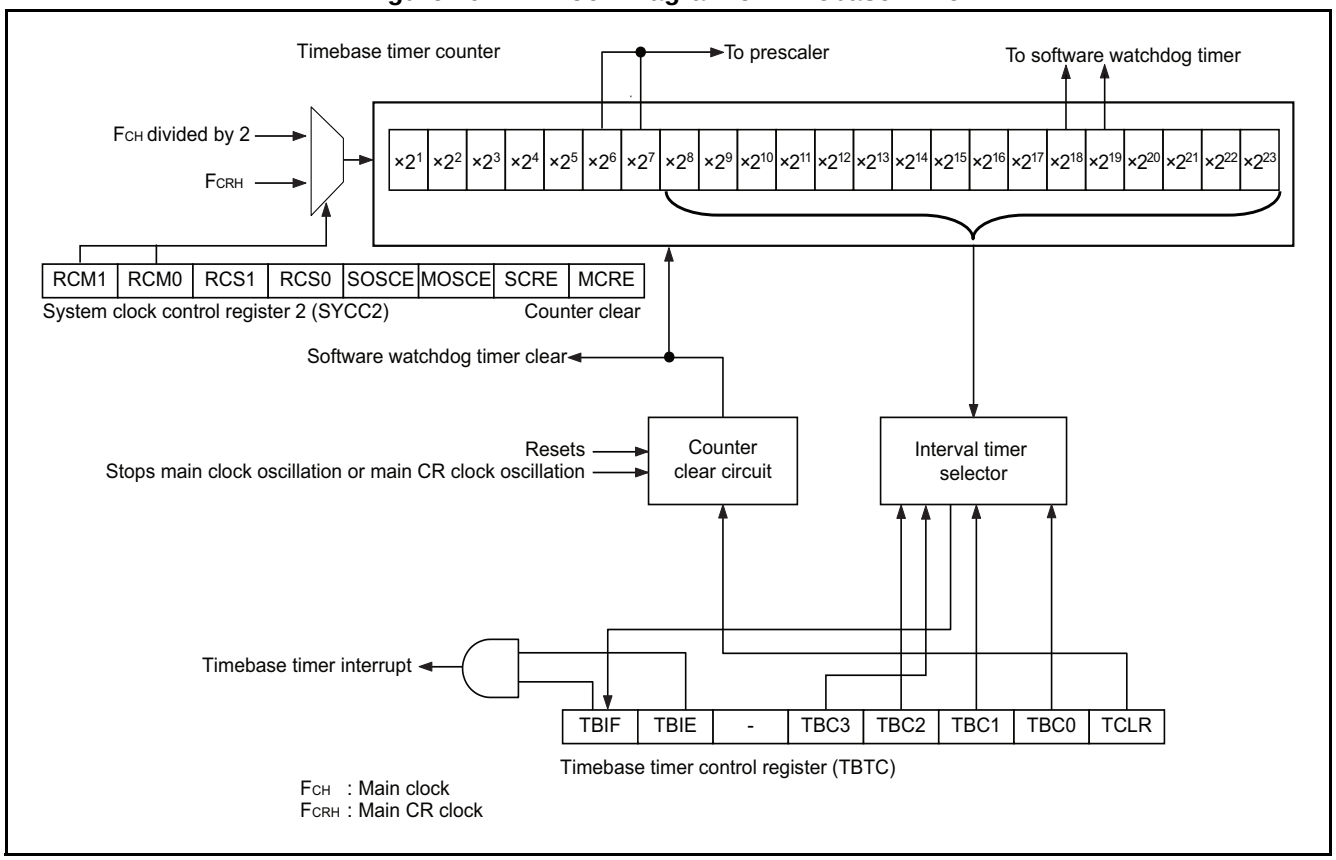
10.2 Configuration of Timebase Timer

The timebase timer consists of the following blocks:

- **Timebase timer counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Timebase timer control register (TBTC)**

■ Block Diagram of Timebase Timer

Figure 10.2-1 Block Diagram of Timebase Timer



- Timebase timer counter

  This is a 24-bit down-counter using the main clock divided by two or the main CR clock as its count clock.

- Counter clear circuit

  This circuit controls the clearing of the timebase timer counter.

- Interval timer selector

  This circuit selects one bit out of 16 bits in the 24 bits of the timebase timer counter as the interval timer.

- Timebase timer control register (TBTC)

  This register selects the interval time, clears the counter, controls interrupts and checks the state of the timebase timer.

■ Input Clock

The timebase timer uses the main clock divided by two or the main CR clock as its input clock (count clock).

■ Output Clock

The timebase timer supplies clocks to the main clock, the software watchdog timer and the prescaler.

10.3 Register of Timebase Timer

Figure 10.3-1 shows the register of the timebase timer.

■ Register of Timebase Timer

Figure 10.3-1 Register of Timebase Timer

Timebase timer control register

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000A$_H$	TBIF	TBIE	-	TBC3	TBC2	TBC1	TBC0	TCLR	00000000$_B$
	R(RM1),W	R/W	R0/WX	R/W	R/W	R/W	R/W	R/W	

R(RM1),W : Readable/writable (The read value is different from the write value. “1" is read by the read-modify-write type of instruction.)
R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0,W : Write only (Writable. "0" is read.)

10.3.1 Timebase Timer Control Register (TBTC)

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

■ Timebase Timer Control Register (TBTC)

Figure 10.3-2 Timebase Timer Control Register (TBTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000AH	TBIF	TBIE	-	TBC3	TBC2	TBC1	TBC0	TCLR	00000000B
	R(RM1),W	R/W	R0/WX	R/W	R/W	R/W	R/W	R0,W	

TCLR	Timebase timer initialization bit	
	Read	Write
0	"0" is always read	Has no effect on operation
1	-	Clears the counter of timebase timer

TBC3	TBC2	TBC1	TBC0	Interval time (Main clock F_{CH} = 4 MHz)	Interval time (Main CR clock F_{CRH} = 8 MHz)
0	1	0	0	2^9 x 2/F_{CH} (256 µs)	2^9 x 1/F_{CRH} (64 µs)
0	0	0	0	2^{10} x 2/F_{CH} (512 µs)	2^{10} x 1/F_{CRH} (128 µs)
0	1	0	1	2^{11} x 2/F_{CH} (1.024 ms)	2^{11} x 1/F_{CRH} (256 µs)
0	0	0	1	2^{12} x 2/F_{CH} (2.048 ms)	2^{12} x 1/F_{CRH} (512 µs)
0	1	1	0	2^{13} x 2/F_{CH} (4.096 ms)	2^{13} x 1/F_{CRH} (1.024 ms)
0	0	1	0	2^{14} x 2/F_{CH} (8.192 ms)	2^{14} x 1/F_{CRH} (2.048 ms)
0	1	1	1	2^{15} x 2/F_{CH} (16.384 ms)	2^{15} x 1/F_{CRH} (4.096 ms)
0	0	1	1	2^{16} x 2/F_{CH} (32.768 ms)	2^{16} x 1/F_{CRH} (8.192 ms)
1	0	0	0	2^{17} x 2/F_{CH} (65.536 ms)	2^{17} x 1/F_{CRH} (16.384 ms)
1	0	0	1	2^{18} x 2/F_{CH} (131.072 ms)	2^{18} x 1/F_{CRH} (32.768 ms)
1	0	1	0	2^{19} x 2/F_{CH} (262.144 ms)	2^{19} x 1/F_{CRH} (65.536 ms)
1	0	1	1	2^{20} x 2/F_{CH} (524.288 ms)	2^{20} x 1/F_{CRH} (131.072 ms)
1	1	0	0	2^{21} x 2/F_{CH} (1.049 s)	2^{21} x 1/F_{CRH} (262.144 ms)
1	1	0	1	2^{22} x 2/F_{CH} (2.197 s)	2^{22} x 1/F_{CRH} (524.288 ms)
1	1	1	0	2^{23} x 2/F_{CH} (4.194 s)	2^{23} x 1/F_{CRH} (1.049 s)
1	1	1	1	2^{24} x 2/F_{CH} (8.389 s)	2^{24} x 1/F_{CRH} (2.097 s)

TBIE	Timebase timer interrupt request enable bit
0	Disables output of interrupt request
1	Enables output of interrupt request

TBIF	Timebase timer interrupt request enable bit	
	Read	Write
0	Interval time has not elapsed	Clears the bit
1	Interval time has elapsed	Has no effect on operation

R(RM1),W : Readable/Writable (The read value is different from the write value. "1" is read by the read-modify-write type of instruction.)
R/W : Readable/Writable (The read value is the same as the write value.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0,W : Write only (Writable. "0" is read.)
: Undefined
: Initial value

Table 10.3-1 Functions of Bits in Timebase Timer Control Register (TBTC)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>TBIF:
Timebase timer interrupt request flag bit</td><td>This flag is set to "1" when the interval time selected by the timebase timer elapses.
An interrupt request is output if this bit and the timebase timer interrupt request enable bit (TBIE) are set to "1".
Writing "0": clears the bit.
Writing "1": has no effect on operation.
If this bit is read by the read-modify-write (RMW) type of instruction, it always returns "1".</td></tr>
<tr><td>bit6</td><td>TBIE:
Timebase timer interrupt request enable bit</td><td>This bit enables/disables output of interrupt requests to interrupt controller.
Writing "0": disables the output of timebase timer interrupt requests.
Writing "1": enables the output of timebase timer interrupt requests.
An interrupt request is output if this bit and the timebase timer interrupt request flag bit (TBIF) are set to "1".</td></tr>
<tr><td>bit5</td><td>Undefined bit</td><td>This bit is undefined.
• When this bit is read, it always returns "0".
• Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit4
to
bit1</td><td>TBC3 to TBC0:
Interval time select bits</td><td>These bits select interval time.
<table>
<tr><th>TBC3</th><th>TBC2</th><th>TBC1</th><th>TBC0</th><th>Interval time
(Main clock F_{CH} = 4 MHz)</th><th>Interval time
(Main CR clock F_{CRH} = 8 MHz)</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>$2^9 \times 2/F_{CH}$ (256 µs)</td><td>$2^9 \times 1/F_{CRH}$ (64 µs)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>$2^{10} \times 2/F_{CH}$ (512 µs)</td><td>$2^{10} \times 1/F_{CRH}$ (128 µs)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>$2^{11} \times 2/F_{CH}$ (1.024 ms)</td><td>$2^{11} \times 1/F_{CRH}$ (256 µs)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>$2^{12} \times 2/F_{CH}$ (2.048 ms)</td><td>$2^{12} \times 1/F_{CRH}$ (512 µs)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>$2^{13} \times 2/F_{CH}$ (4.096 ms)</td><td>$2^{13} \times 1/F_{CRH}$ (1.024 ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>$2^{14} \times 2/F_{CH}$ (8.192 ms)</td><td>$2^{14} \times 1/F_{CRH}$ (2.048 ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>$2^{15} \times 2/F_{CH}$ (16.384 ms)</td><td>$2^{15} \times 1/F_{CRH}$ (4.096 ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>$2^{16} \times 2/F_{CH}$ (32.768 ms)</td><td>$2^{16} \times 1/F_{CRH}$ (8.192 ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>$2^{17} \times 2/F_{CH}$ (65.536 ms)</td><td>$2^{17} \times 1/F_{CRH}$ (16.384 ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>$2^{18} \times 2/F_{CH}$ (131.072 ms)</td><td>$2^{18} \times 1/F_{CRH}$ (32.768 ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>$2^{19} \times 2/F_{CH}$ (262.144 ms)</td><td>$2^{19} \times 1/F_{CRH}$ (65.536 ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>$2^{20} \times 2/F_{CH}$ (524.288 ms)</td><td>$2^{20} \times 1/F_{CRH}$ (131.072 ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>$2^{21} \times 2/F_{CH}$ (1.049 s)</td><td>$2^{21} \times 1/F_{CRH}$ (262.144 ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>$2^{22} \times 2/F_{CH}$ (2.097 s)</td><td>$2^{22} \times 1/F_{CRH}$ (524.288 ms)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>$2^{23} \times 2/F_{CH}$ (4.194 s)</td><td>$2^{23} \times 1/F_{CRH}$ (1.049 s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>$2^{24} \times 2/F_{CH}$ (8.389 s)</td><td>$2^{24} \times 1/F_{CRH}$ (2.097 s)</td></tr>
</table></td></tr>
<tr><td>bit0</td><td>TCLR:
Timebase timer initialization bit</td><td>This bit clears the timebase timer counter.
Writing "0": is ignored and has no effect on the operation.
Writing "1": initializes all counter bits to "1".
When this bit is read, it always returns "0".
Note: When the output of the timebase timer is selected as the count clock for the watchdog timer, using this bit to clear the timebase timer also clears the software watchdog timer.</td></tr>
</table>

10.4 Interrupts of Timebase Timer

An interrupt request is generated when the interval time selected by the timebase timer elapses (interval timer function).

■ Interrupts when Interval Function is in Operation

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

- Regardless of the value of the TBIE bit, the TBIF bit is set to "1" when the selected bit underflows.
- With the TBIF bit having been set to "1", if the TBIE bit is changed from the disable state to the enable state ($0 \rightarrow 1$), an interrupt request is generated immediately.
- The TBIF bit will not be set to "1" if the counter is cleared (TBTC:TCLR = 1) at the same time as the timebase timer counter underflows.
- In the interrupt service routine, write "0" to the TBIF bit to clear an interrupt request.

Note:

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

Table 10.4-1 Interrupts of Timebase Timer

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

■ Register and Vector Table Addresses for Interrupts of Timebase Timer

Table 10.4-2 Register and Vector Table Addresses for Interrupts of Timebase Timer

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Timebase timer	IRQ19	ILR4	L19	$FFD4_H$	$FFD5_H$

See "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector table addresses of all peripheral functions.

Note:

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

10.5 Operations of Timebase Timer and Setting Procedure Example

This section describes the operations of the interval timer function of the timebase timer.

■ Operations of Timebase Timer

The counter of the timebase timer is initialized to "FFFFFF$_H$" after a reset and starts counting while being synchronized with the main clock divided by two.

The timebase 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 "FFFFFF$_H$".

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

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
000A$_H$	TBTC	TBIF	TBIE	-	TBC3	TBC2	TBC1	TBC0	TCLR
		0	1		◎	◎	◎	◎	0

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

When the timebase timer initialization bit in the timebase timer control register (TBTC:TCLR) is set to "1", the counter of the timebase timer is initialized to "FFFFFF$_H$" and continues to count down. When the selected interval time has elapsed, the timebase timer interrupt request flag bit of the timebase 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 Timebase Timer

If the timebase timer is cleared when the output of the timebase 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 timebase timer initialization bit (TBTC:TCLR), modify the settings of other peripheral functions whenever necessary so that clearing the counter does not have any unexpected effect on them.

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

The timebase timer is cleared not only by the timebase timer initialization bit (TBTC:TCLR), but also when the main clock is stopped and the oscillation stabilization wait time is necessary. The timebase timer is cleared in the following situations:

- When the device transits from the main clock mode or main CR clock mode to the stop mode
- When the device transits from the main clock mode or main CR clock mode to the subclock mode or sub-CR clock mode
- At power on
- At low-voltage detection reset

After the main clock oscillation stabilization wait time elapses, if a reset occurs with the main clock running, the counter of the timebase timer is cleared and stops operating. However, the counter continues to operate even during a reset if the oscillation stabilization wait time is necessary.

■ Operation Examples of Timebase Timer

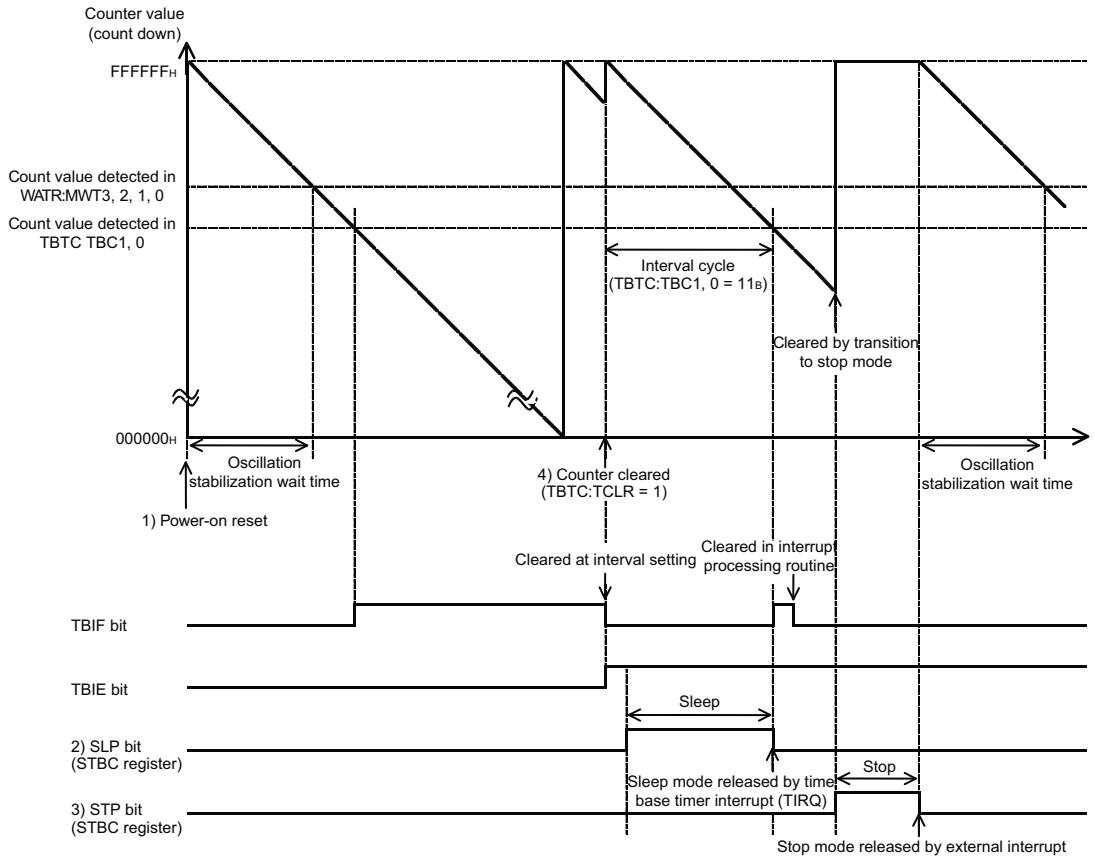
Figure 10.5-2 shows examples of operations under the following conditions:

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

If the device transits to the timebase time mode, the same operations are executed as those executed when the device transits to the sleep mode.

In stop mode in which the clock mode is subclock mode, sub-CR clock mode, main clock mode or main CR clock mode, the timer operation stops because it is cleared and the main clock stops. After the device wakes up from the stop mode, the timebase timer is used to count the oscillation stabilization wait time.

Figure 10.5-2 Operations of Timebase Timer



• When setting the interval time select bits in timebase timer control register (TBTC:TBC3 to TBC0) to "11_B" (2^{16} x $2/F_{CH}$)

• TBTC:TBC3 to TBC0: Interval time select bits in timebase timer control register
• TBTC:TCLR : Timebase timer initialization bit in timebase timer control register
• TBTC:TBIF : Timebase timer interrupt request flag bit in timebase timer control register
• TBTC:TBIE : Timebase timer interrupt request enable bit in timebase timer control register
• STBC:SLP : Sleep bit in standby control register
• STBC:STP : Stop bit in standby control register
• WATR:MWT : Main clcok oscillation stabilization wait time select bit in oscillation stabilization wait time setting register

■ Setting Procedure Example

Below is an example of procedure for setting the timebase timer.

● Initial settings

1) Disable interrupts. (TBTC:TBIE = 0)
2) Set the interval time. (TBTC:TBC3 to TBC0)
3) Enable interrupts. (TBTC:TBIE = 1)
4) Clear the counter. (TBTC:TCLR = 1)

● Processing interrupts

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

10.6 Notes on Using Timebase Timer

This section provides notes on using the timebase timer.

■ Notes on Using Timebase Timer

● When setting the timer by program

The timer cannot be waken up from interrupt processing when the timebase timer interrupt request flag bit (TBTC:TBIF) is set to "1" and the interrupt request enable bit is enabled (TBTC:TBIE = 1). Always clear the TBIF bit in the interrupt service routine.

● Clearing Timebase Timer

The timebase timer is cleared not only by the timebase timer initialization bit (TBTC:TCLR = 1) but also when the oscillation stabilization wait time of the main clock is required. When the timebase timer is selected as the count clock of the software watchdog timer (WDTC:CS2, CS1 = 00_B or CS2, CS1, = 01_B), clearing the timebase timer also clears the software watchdog timer.

● Peripheral functions receiving clock from timebase timer

In the mode where the source oscillation of the main clock is stopped, the counter is cleared and the timebase timer stops operating. In addition, if the counter of the timebase timer is cleared with the output of the timebase timer being used in other peripheral functions, that will affect the operations of such peripheral operations such as the changing of their operating cycles.

After the counter of the timebase timer is cleared, the clock that is output from the timebase timer for the software watchdog timer returns to the initial state. However, since the software watchdog timer counter is also cleared at the same time as the clock for the software watchdog timer returns to the initial state, the software watchdog timer operates in its normal cycle.

CHAPTER 11

HARDWARE/SOFTWARE WATCHDOG TIMER

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

11.1 Overview of Watchdog Timer

The watchdog timer serves 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 a program entering an infinite loop.

The output of the timebase timer or of the watch prescaler or of the sub-CR timer can be selected as the count clock for the watchdog timer.

The watchdog timer is activated according to the values at the addresses $FFBE_H$ and $FFBF_H$ on the flash memory, which are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$/$0FEC_H$). In the case of software activation (software watchdog), the watchdog timer register (WDTC) must be set to start the watchdog timer function. In the case of hardware activation (hardware watchdog), the watchdog timer starts automatically after a reset. It can also stop or run in stop mode according to the values at the addresses $FFBE_H$ and $FFBF_H$ on the flash memory, which are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$/$0FEC_H$). See "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)" for details of the watchdog timer selection ID.

The intervals 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 bit CS[1:0], CSP	Interval time	
		Minimum time	Maximum time
Timebase timer output (main clock = 4 MHz)	000_B (SWWDT)	524 ms	1.05 s
	010_B (SWWDT)	262 ms	524 ms
Watch prescaler output (subclock = 32.768 kHz)	100_B (SWWDT)	500 ms	1.00 s
	110_B (SWWDT)	250 ms	500 ms
Sub-CR timer (sub-CR clock = 50-200 kHz)	$XX1_B$ (SWWDT) or HWWDT*1	328 ms	2.62 s

*1: CS[1:0]=00B, CSP=1B(read only)

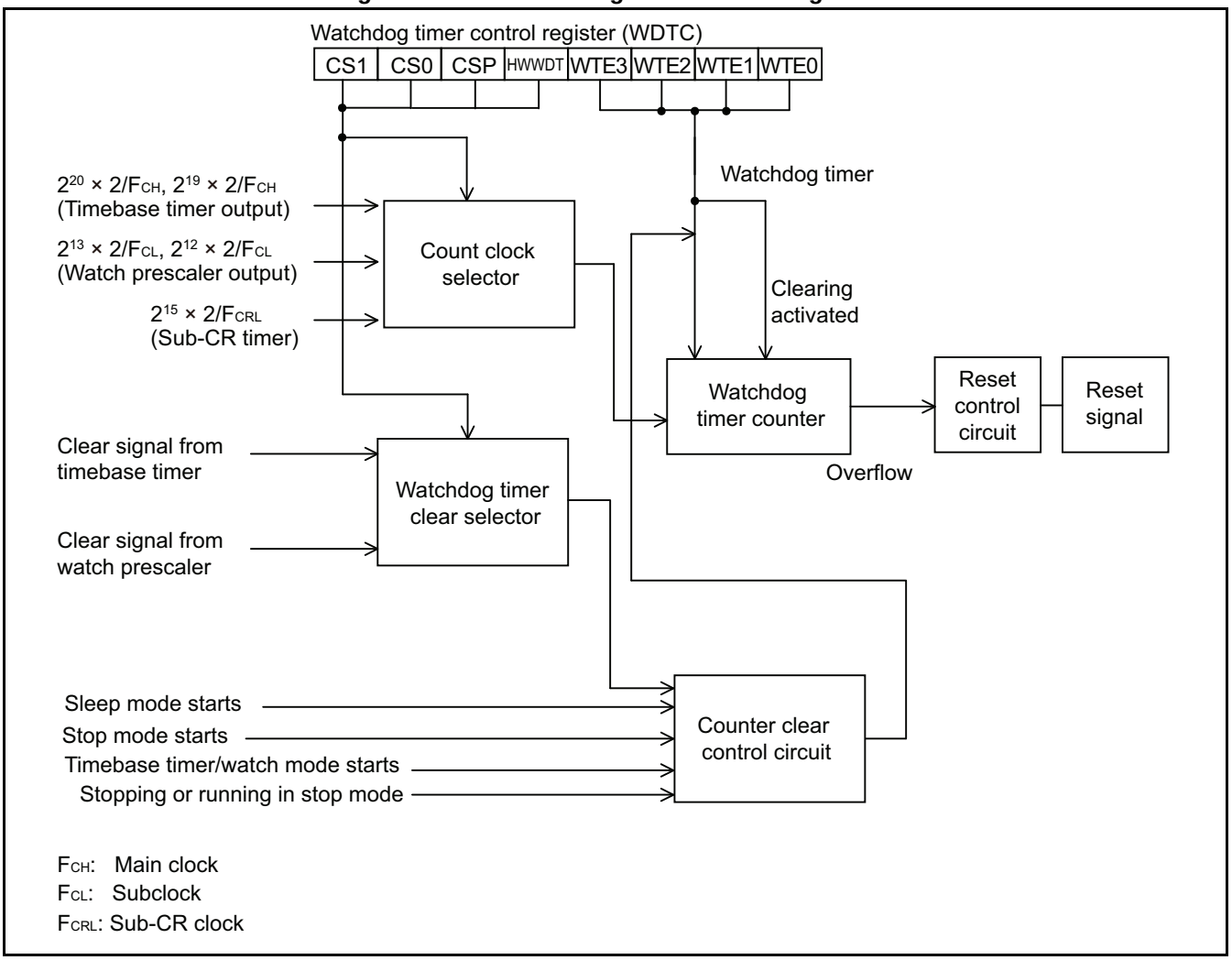
11.2 Configuration of Watchdog Timer

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 the timebase timer or of the watch prescaler or of the sub-CR timer 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 of the timebase timer or of the watch prescaler or of the sub-CR timer as the input clock (count clock).

11.3 Register of Watchdog Timer

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

■ Register of Watchdog Timer

Figure 11.3-1 Register of Watchdog Timer

Watchdog timer control register (WDTC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000C_H$	CS1	CS0	CSP	HWWDT	WTE3	WTE2	WTE1	WTE0	
Software	R/W	R/W	R/W	R0,WX	R0/W	R0/W	R0/W	R0/W	00000000_B
Hardware	R0/WX	R0/WX	R1/WX	R1,WX	R0/W	R0/W	R0/W	R0/W	00110000_B

R/W: Readable/writable (The read value is the same as the write value.)
R0/WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R1/WX: Undefined bit (The read value is "1". Writing a value to it has no effect on operation.)
R0,W: Write only (Writable. "0" is read.)
-: 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	CSP	HWWDT	WTE3	WTE2	WTE1	WTE0	
Software	R/W	R/W	R/W	R0,WX	R0/W	R0/W	R0/W	R0/W	00000000$_B$
Hardware	R0/WX	R0/WX	R1/WX	R1,WX	R0/W	R0/W	R0/W	R0/W	00110000$_B$

WTE3	WTE2	WTE1	WTE0	Watchdog control bits
0	1	0	1	• Activates software watchdog timer (at the first write access after a reset) • Clears watchdog timer Software: from the second write access after a reset Hardware: from the first write access after a reset
Other than above				No effect on operation

HWWDT	Hardware watchdog timer activation bit
1	Hardware watchdog timer is activated
0	Hardware watchdog timer stops (software watchdog timer can be activated)

CS1	CS0	CSP	Count clock switch bits
0	0	0	Output cycle of timebase timer ($2^{21}/F_{CH}$)
0	1	0	Output cycle of timebase timer ($2^{20}/F_{CH}$)
1	0	0	Output cycle of watch prescaler ($2^{14}/F_{CL}$)
1	1	0	Output cycle of watch prescaler ($2^{13}/F_{CL}$)
X	X	1	Output cycle of sub-CR timer ($2^{16}/F_{CRL}$)

R/W : Readable/writable (The read value is the same as the write value.)
R0,W : Write only (Writable. "0" is read.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R1/WX : Undefined bit (The read value is "1". Writing a value to it has no effect on operation.)
- : Undefined
□ : Initial value for Software watchdog timer
F_{CH} : Main clock
F_{CL} : Subclock
F_{CRL} : Sub-CR clock

Table 11.3-1 Functions of Bits in 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 rowspan="2">These bits select the count clock of the watchdog timer.
<table>
<tr><th>CS1</th><th>CS0</th><th>CSP</th><th>Count clock switch bits</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Output cycle of timebase timer ($2^{21}/F_{CH}$)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Output cycle of timebase timer ($2^{20}/F_{CH}$)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Output cycle of watch prescaler ($2^{14}/F_{CL}$)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Output cycle of watch prescaler ($2^{13}/F_{CL}$)</td></tr>
<tr><td>X</td><td>X</td><td>1</td><td>Output cycle of sub-CR timer ($2^{16}/F_{CRL}$)</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: Since the timebase timer in stopped in subclock mode, always select the output of the watch prescaler in subclock mode. Do not select the output of the watch prescaler for a single external clock system.</td></tr>
<tr><td>bit5</td><td>CSP:
Count clock select sub-CR selector bit</td></tr>
<tr><td>bit4</td><td>Hardware watchdog activation bit</td><td>The bit is a read-only bit, used to confirm the start/stop of the hardware watchdog timer.
"1": The hardware watchdog timer has been activated.
"0": The hardware watchdog timer has stopped (The software watchdog timer can be activated).</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 (from second write after reset).
Writing other than "0101_B": has no effect on operation.
• When these bits are read, they always return "0000_B".</td></tr>
</table>

The read-modify-write type of instruction cannot be used.

11.4 Operations of Watchdog Timer and Setting 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

Software watchdog

- 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,CSP) should also be set at the same time.
- Once the watchdog timer is activated, a reset is the only way to stop its operation.

Hardware watchdog

- Write "$A597_H$" (the hardware watchdog time is enabled except in standby mode) or any other value (the hardware watchdog time is enabled in every mode) except "$A596_H$" and "$A597_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the flash memory, which are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$/$0FEC_H$). See "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)" for details of the watchdog timer selection ID.
- Start operation after a reset.
- CS1,CS0,CSP bits are read-only bits, fixed at "001_B".
- The timer is cleared by a reset and resumes operation after the reset is released.

● 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 hardware watchdog timer is cleared when "0101_B" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0). The counter of the software 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 time and from the second time onward.
- The watchdog timer is cleared at the same time as the timer selected as the count clock (timebase timer or watch prescaler) is cleared.

● Operation in standby mode

Regardless of the clock mode selected, the watchdog timer clears its counter and stops the operation when transiting to standby mode (sleep/stop/timebase timer/watch), except in the case of selecting the hardware activation with the hardware watchdog timer running in standby mode.

Once released from standby mode, the timer restarts the operation, except in the case of selecting the hardware activation with the hardware watchdog timer running in standby mode.

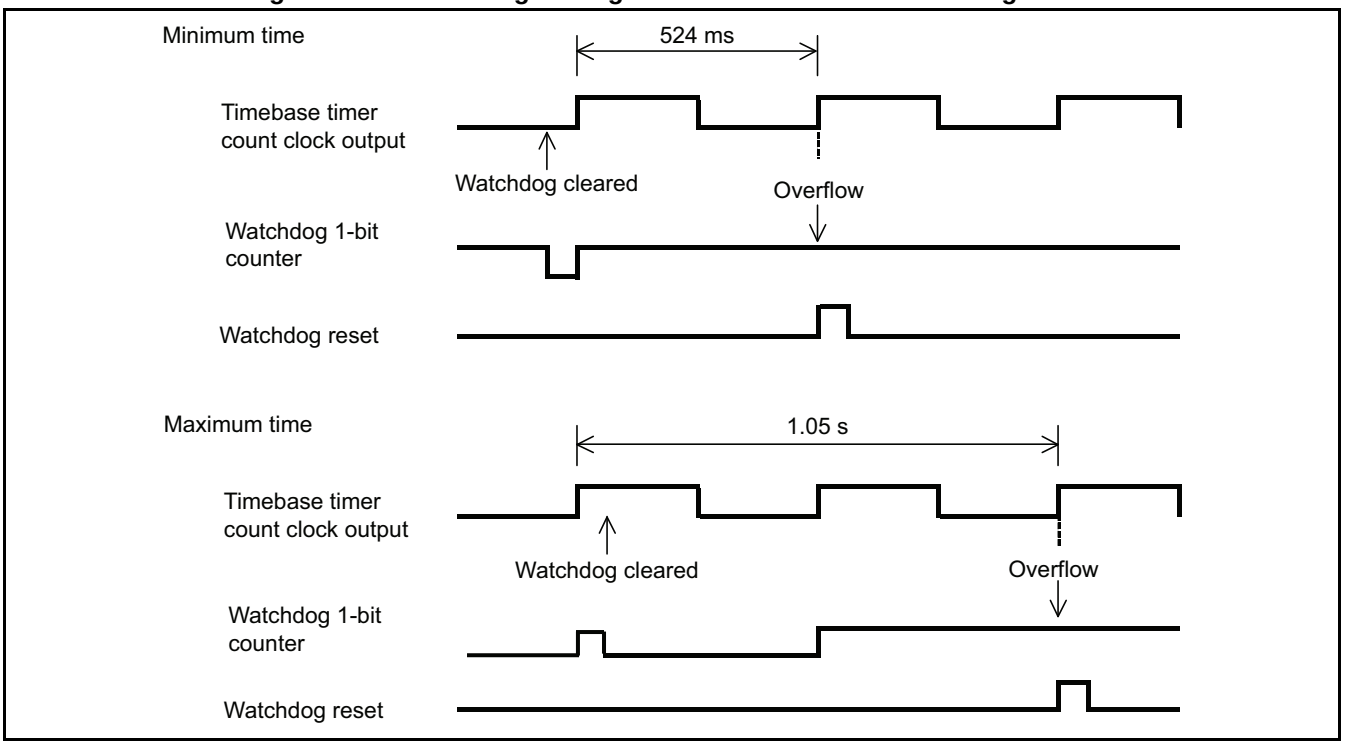
Note:

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

● Interval time

The interval time varies depending on the timing of clearing the watchdog timer. Figure 11.4-1 shows the correlation between the timing of clearing timing the watchdog timer and the interval time when the timebase timer output $2^{21}F_{CH}$ (F_{CH}: main clock) is selected as the count clock (main clock = 4 MHz).

Figure 11.4-1 Clearing Timing and Interval Time of Watchdog Timer



● Operation in subclock mode

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

■ Setting Procedure Example

Below is the procedure for setting the software watchdog timer.

1) Select the count clock. (WDTC:CS1, CS0, CSP)
2) Activate the watchdog timer. (WDTC:WTE3 to WTE0 = 0101_B)
3) Clear the watchdog timer. (WDTC:WTE3 to WTE0 = 0101_B)

Below is the procedure for setting the hardware watchdog timer.

1) Write "$A597_H$" (the hardware watchdog time is enabled except in standby mode) or any other value (the hardware watchdog time is enabled in every mode) except "$A596_H$" and "$A597_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the flash memory, which are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$/$0FEC_H$). See "CHAPTER 22 NON-VOLATILE REGISTER FUNCTION (NVR)" for details of the watchdog timer selection ID.
2) Clear the watchdog timer.(WDTC:WTE3 to WTE0 = 0101_B)

11.5 Notes on Using Watchdog Timer

This section provides notes on using the watchdog timer.

■ Notes on Using Watchdog Timer

● Stopping the watchdog timer

Software watchdog timer

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

● Selecting the count clock

Software watchdog timer

The count clock switch bits (WDTC:CS1, CS0, CSP) can be modified only when the watchdog control bits (WDTC:WTE3 to WTE0) are set to "0101_B" after the activation of the watchdog timer. The count clock switch bits cannot be set by a bit manipulation instruction. Moreover, the bit settings should not be changed once the timer is activated.

In subclock mode, the timebase timer does not operate because the main clock stops oscillating.

In order to make the watchdog timer operate in subclock mode, it is necessary to select the watch prescaler as the count clock beforehand and set "WDTC:CS1,CS0,CSP" to "100_B" or "110_B" or "$XX1_B$".

● Clearing the watchdog timer

Clearing the counter used as the count clock of the watchdog timer (timebase timer or watch prescaler or sub-CR timer) also clears the counter of the watchdog timer.

The counter of the watchdog timer is cleared when the watchdog timer transits to the sleep mode, stop mode or watch mode, except in the case of selecting the hardware activation with the hardware watchdog timer running in standby 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.

● Hardware watchdog (with timer running in standby mode)

The watchdog timer does not stop in stop mode, sleep mode, timebase timer mode or watch mode. Therefore, the watchdog timer is not to be cleared by the CPU even if the internal clock stops. (in stop mode, sleep mode, timebase timer mode or watch mode).

Regularly release the device from standby mode and clear the watchdog timer. However, depending on the setting of the oscillation stabilization wait time setting register, a watchdog reset may be generated after the CPU wakes up from stop mode in subclock mode or sub-CR clock mode.

Take account of the setting of the subclock stabilization wait time when selecting the subclock.

CHAPTER 12

WATCH PRESCALER

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

12.1 Overview of Watch Prescaler

The watch prescaler is a 16-bit down-counting, free-run counter, which is synchronized with the subclock divided by two or the sub-CR 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 subclock divided by two as its count clock.

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

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

Table 12.1-1 Interval Times of Watch Prescaler

	Interval time (Sub-CR clock) ($2^n \times 2/F_{CRL}$ *1)	Interval time (Subclock) ($2^n \times 2/F_{CL}$ *2)
n=10	20.48 [ms]	62.5 [ms]
n=11	40.96 [ms]	125 [ms]
n=12	81.92 [ms]	250 [ms]
n=13	163.84 [ms]	500 [ms]
n=14	327.68 [ms]	1 [s]
n=15	655.36 [ms]	2 [s]
n=16	1.311 [s]	4 [s]
n=17	2.621 [s]	8 [s]

*1: $2/F_{CRL}$=20 [µs] when F_{CRL}=100 [kHz]

*2: $2/F_{CL}$=61.035 [µs] when F_{CL}=32.768 [kHz]

Note:

Refer to the data sheet of the MB95200H/210H Series for the accuracy of the sub-CR clock frequency.

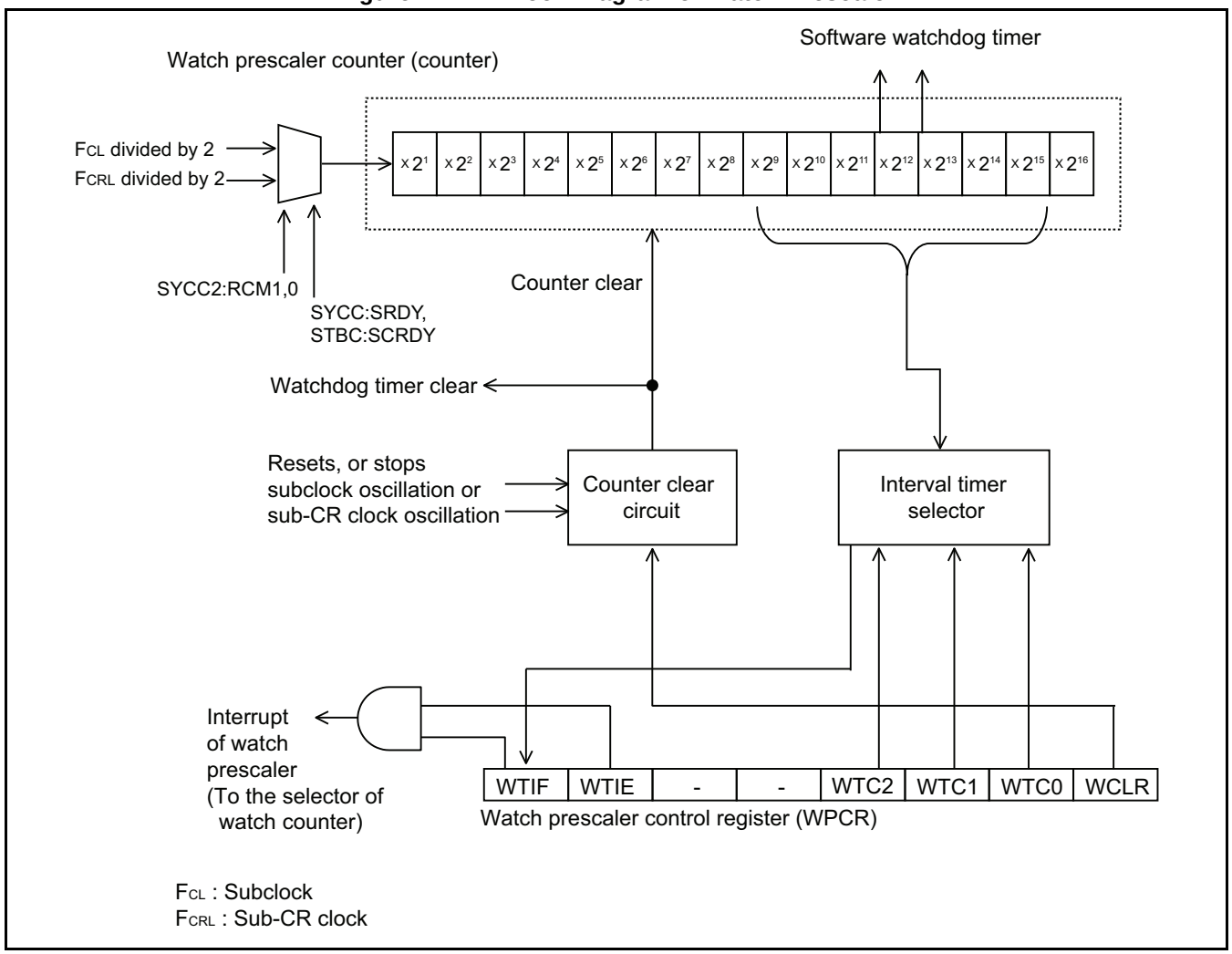
12.2 Configuration of Watch Prescaler

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 16-bit down-counter that uses the subclock divided by two or the sub-CR 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 eight bits used for the interval timer among 16 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 subclock divided by two or the sub-CR clock divided by two as its input clock (count clock).

■ Output Clock

The watch prescaler supplies its clock to the timer for the software watchdog timer and the watch counter.

12.3 Register of Watch Prescaler

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

■ Register of Watch Prescaler

Figure 12.3-1 Register of Watch Prescaler

Watch prescaler control register (WPCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$000B_H$	WTIF	WTIE	-	-	WTC2	WTC1	WTC0	WCLR	00000000_B
	R(RM1),W	R/W	R0/WX	R0/WX	R/W	R/W	R/W	R0,W	

R(RM1),W: Readable/writable (The read value is different from write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/W: Readable/writable (The read value is the same as the write value.)
R0/WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0,W: Write only (Writable. "0" is read)
-: 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 of the watch prescaler.

■ 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	-	-	WTC2	WTC1	WTC0	WCLR	00000000$_B$
	R(RM1),W	R/W	R0/WX	R0/WX	R/X	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.

WTC2	WTC1	WTC0	Interval time (Subclock F_{CL}=32.768 kHz)	Interval time (Sub-CR clock F_{CRL}=100 kHz)
1	0	0	$2^{10} \times 2/F_{CL}$ (62.5ms)	$2^{10} \times 2/F_{CRL}$ (20.48 ms)
0	0	0	$2^{11} \times 2/F_{CL}$ (125 ms)	$2^{11} \times 2/F_{CRL}$ (40.96 ms)
0	0	1	$2^{12} \times 2/F_{CL}$ (250 ms)	$2^{12} \times 2/F_{CRL}$ (81.92 ms)
0	1	0	$2^{13} \times 2/F_{CL}$ (500 ms)	$2^{13} \times 2/F_{CRL}$ (163.84 ms)
0	1	1	$2^{14} \times 2/F_{CL}$ (1 s)	$2^{14} \times 2/F_{CRL}$ (327.68 ms)
1	0	1	$2^{15} \times 2/F_{CL}$ (2 s)	$2^{15} \times 2/F_{CRL}$ (655.36 ms)
1	1	0	$2^{16} \times 2/F_{CL}$ (4 s)	$2^{16} \times 2/F_{CRL}$ (1.311 s)
1	1	1	$2^{17} \times 2/F_{CL}$ (8 s)	$2^{17} \times 2/F_{CRL}$ (2.621 s)

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(RM1),W: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0,W : Write only (Writable. "0" is read.)
- : Undefined
□ : Initial value

Table 12.3-1 Functions of Bits in 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.
• An interrupt request is generated when this bit and the interrupt request enable bit (WTIE) are set to "1".
Writing "0": sets this bit to "0".
Writing "1": is ignored and has no effect on operation.
• If this bit is read by the read-modify-write (RMW) type of instruction, it always returns "1".</td></tr>
<tr><td>bit6</td><td>WTIE:
Interrupt request enable bit</td><td>This bit enables or disables the output of interrupt requests to interrupt controller.
Writing "0": disables the interrupt request output of the watch prescaler.
Writing "1": enables the interrupt request output of the watch prescaler.
An interrupt request is output when this bit and the watch interrupt request flag bit (WTIF) are set to "1".</td></tr>
<tr><td>bit5, bit4</td><td>Undefined bits</td><td>These bits are undefined.
• When this bit is read, it always returns "0".
• Writing a value to this bit has no effect on operation.</td></tr>
<tr><td>bit3 to bit1</td><td>WTC2 to WTC0:
Watch interrupt interval time select bits</td><td>These bits select the interval time.
<table>
<tr><th>WTC2</th><th>WTC1</th><th>WTC0</th><th>Interval time
(Subclock F_{CL} = 32.768 kHz)</th><th>Interval time
(Sub-CR clock F_{CRL} = 100 kHz)</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$2^{10} \times 2/F_{CL}$ (62.5 ms)</td><td>$2^{10} \times 2/F_{CRL}$ (20.48 ms)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$2^{11} \times 2/F_{CL}$ (125. ms)</td><td>$2^{11} \times 2/F_{CRL}$ (40.96 ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$2^{12} \times 2/F_{CL}$ (250. ms)</td><td>$2^{12} \times 2/F_{CRL}$ (81.92 ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$2^{13} \times 2/F_{CL}$ (500. ms)</td><td>$2^{13} \times 2/F_{CRL}$ (163.84 ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$2^{14} \times 2/F_{CL}$ (1 s)</td><td>$2^{14} \times 2/F_{CRL}$ (327.68 ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$2^{15} \times 2/F_{CL}$ (2 s)</td><td>$2^{15} \times 2/F_{CRL}$ (655.36 ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$2^{16} \times 2/F_{CL}$ (4 s)</td><td>$2^{16} \times 2/F_{CRL}$ (1.311 s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$2^{17} \times 2/F_{CL}$ (8 s)</td><td>$2^{17} \times 2/F_{CRL}$ (2.621 s)</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": is ignored and has no effect on operation.
Writing "1": initializes all counter bits to "1".
When this bit is read, it always returns "0".
Note: When the output of the watch prescaler is selected as the count clock of the software watchdog timer, clearing the watch prescaler with this bit also clears the software 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 except the stop mode in which the main clock is used, if the watch prescaler counter counts up using the source oscillation of the subclock and the time of the interval timer has elapsed, the watch interrupt request flag bit is set to "1" (WPCR:WTIF = 1). At that time, if the interrupt request enable bit has been enabled (WPCR:WTIE = 1), an interrupt request (IRQ20) is output from the watch prescaler to the interrupt controller.

- Regardless of the value in the WTIE bit, the WTIF bit is set to "1" as soon as the time set by the watch interrupt interval time select bits has elapsed.
- When the WTIF bit is set to "1", changing the WTIE bit from the disable state to the enable state (WPCR:WTIE = 0 → 1) immediately generates an interrupt request.
- The WTIF bit will not be set to "1" if 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 service routine to clear an interrupt request to "0".

Note:

To enable the output of interrupt requests after releasing a reset, set the WTIE bit in the WPCR register to "1" and clear the WTIF bit in the same register simultaneously.

■ Interrupts of Watch Prescaler

Table 12.4-1 Interrupts of Watch Prescaler

Item	Description
Interrupt condition	Interval time set by "WPCR:WTC2 to WTC0" has elapsed.
Interrupt flag	WPCR:WTIF
Interrupt enable	WPCR:WTIE

■ Register and Vector Table Addresses Related to Interrupts of Watch Prescaler

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

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Watch prescaler*	IRQ20	ILR5	L20	$FFD2_H$	$FFD3_H$

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

See "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector table addresses of all

peripheral functions.

Note:

In the case of the interval time set for the watch prescaler being shorter than the oscillation stabilization wait time of the subclock, when the device transits to stop mode in subclock mode or sub-CR mode, an interrupt request is generated from the watch prescaler during the subclock oscillation stabilization wait time required for the wake-up from stop mode caused by an external interrupt. To prevent the interrupt request from being generated, set the interrupt request enable bit (WPCR:WTIE) in the watch prescaler control register to "0" to disable interrupts of the watch prescaler when the device transits to stop mode in subclock mode or sub-CR clock mode.

12.5 Operations of Watch Prescaler and Setting 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 subclock divided by two as its count clock as long as the subclock oscillates.

When cleared (WPCR:WCLR = 1), the counter starts counting down from "$FFFF_H$". Once it reaches "0000_H", it returns to "$FFFF_H$" to continue counting. As soon as the time set by the interrupt interval time select bits has elapsed during the counting down, the watch interrupt request flag bit (WPCR:WTIF) is set to "1" in any mode except the stop mode in which the main clock is used. In other words, a watch interrupt request is generated at every selected interval time, based on the time when the counter was last cleared.

■ Clearing Watch Prescaler

If the watch prescaler is cleared, other peripheral functions that are using the watch prescaler output are affected by changes in count time and by other factors.

When clearing the counter using the watch prescaler initialization bit (WTCR:WCLR), modify the settings of other peripheral functions so that clearing the counter does not have any unexpected effect on them.

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 subclock is stopped and the oscillation stabilization wait time has to be wai.

- When the device transits from the subclock mode or sub-CR clock mode to the stop mode
- When the subclock oscillation enable bits in the system clock control register2 (SYCC2:SOSCE or SCRE) is set to "0" in main clock mode or main CR clock mode

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

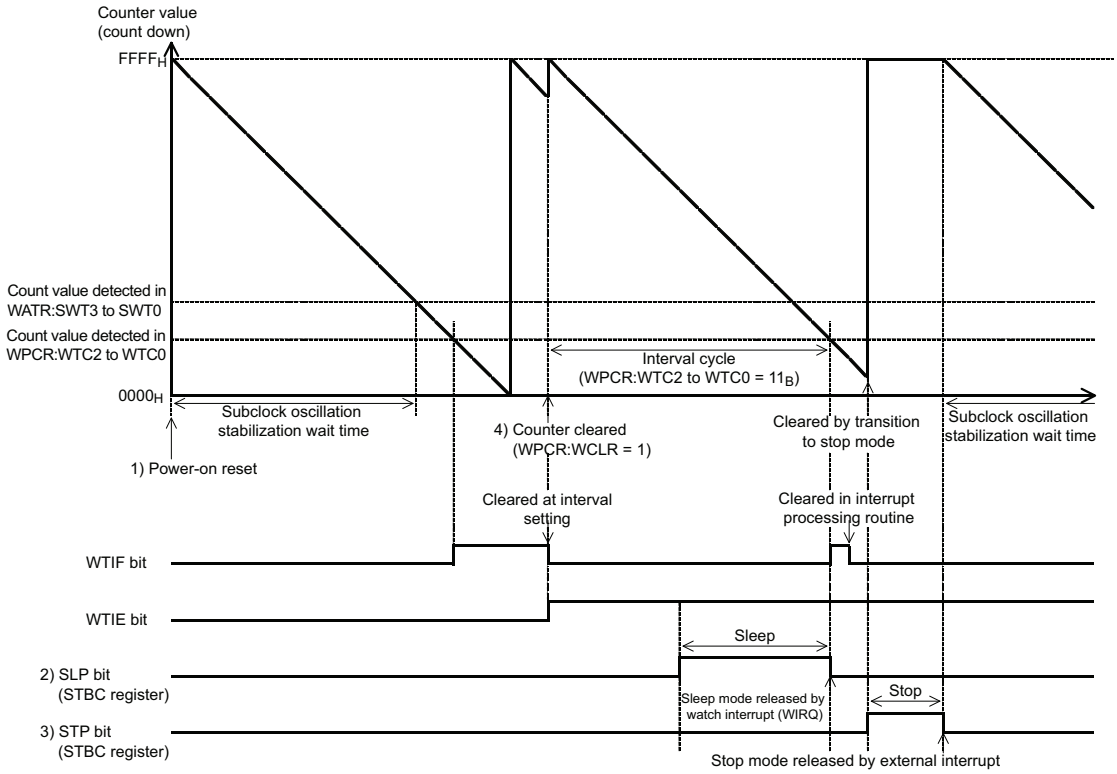
■ Operation Examples of Watch Prescaler

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

1) When a power-on reset occurs
2) When the device transits to the sleep mode during the operation of the interval timer function in subclock mode or sub-CR clock mode
3) When the device transits to the stop mode during the operation of the interval timer function in subclock mode or sub-CR clock mode
4) When a request for clearing the counter is issued

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

Figure 12.5-1 Operation Examples of Watch Prescaler



• When setting interval time select bits in the watch prescaler control register (WPCR:WTC2 to WTC0) to "11_B" (2^{14} x 2/F_{CL})

• WPCR:WTC2 to 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 : Subclock oscillation stabilization wait time select bits in oscillation stabilization wait time setting register

■ Setting Procedure Example

Below is an example of procedure for setting the watch prescaler.

● Initial settings

1) Set the interrupt level. (ILR5)
2) Set the interval time. (WPCR:WTC2 to WTC0)
3) Enable interrupts. (WPCR:WTIE = 1)
4) Clear the counter. (WPCR:WCLR = 1)

● Processing interrupts

1) Clear the interrupt request flag. (WPCR:WTIF = 0)
2) Process an interrupt.

12.6 Notes on Using Watch Prescaler

This section provides notes on using the watch prescaler.

■ Notes on Using Watch Prescaler

- When setting interrupt processing in a program

  The watch prescaler cannot be waken up from interrupt processing if the watch interrupt request flag bit (WPCR:WTIF) is set to "1" and the interrupt request is enabled (WPCR:WTIE = 1). Always clear the WTIF bit in the interrupt routine.

- Clearing the watch prescaler

  When the watch prescaler is selected as the count clock of the software watchdog timer (WDTC:CS1, CS0, CSP = 100_B or 110_B), clearing the watch prescaler also clears the software watchdog timer.

- Watch interrupts

  In stop mode in which the main clock is used, the watch prescaler performs counting but does not generate the watch prescaler interrupt (IRQ20).

- Peripheral functions receiving clock from the watch prescaler

  If the counter of the watch prescaler is cleared when the output of the watch prescaler is used in other peripheral functions, the operations of such peripheral functions may be affected such as the changing of their operating cycles.

  After the counter of the watch prescaler is cleared, the clock for the software watchdog timer output from the watch prescaler returns to the initial state. However, since the software watchdog timer counter is also cleared at the same time as the clock for the software watchdog timer returns to the initial state, the software watchdog timer operates in its normal cycle.

12.7 Sample Programs for Watch Prescaler

Fujitsu Microelectronics provides 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, see "■ Sample programs" in "Preface".

■ Setting Methods other than Sample Programs

● How to initialize the watch prescaler

The watch timer initialization bit (WPCR:WCLR) is used.

Action	Watch timer initialization bit (WCLR)
To initialize the watch prescaler	Write "1" to this bit

● How to select the interval time

The watch interrupt interval time select bits (WPCR:WTC2 to WTC0) are used to select the interval time.

● Interrupt-related register

The interrupt level register shown in the following table is used to select the interrupt level.

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

Interrupt request enable bit, Watch interrupt request flag

The interrupt request enable bit (WPCR:WTIE) is used to enable interrupts.

Action	Interrupt request enable bit (WTIE)
To disable interrupt requests	Write "0" to this bit
To enable interrupt requests	Write "1" to this bit

The watch interrupt request flag (WPCR:WTIF) is used to clear interrupt requests.

Action	Watch interrupt request flag (WTIF)
To clear interrupt requests	Write "0" to this bit

CHAPTER 13

WILD REGISTER FUNCTION

This chapter describes the functions and operations of the wild register function.

13.1 Overview of Wild Register Function

**The wild register function can be used to patch bugs in a program with addresses and amendment data, both of which are to be set in built-in registers.
This section describes the wild register function.**

■ Wild Register Function

The wild register consists of three wild register data setting registers, three wild register address setting registers, a 1-byte address compare enable register and a 1-byte wild register data test setting register. If addresses and data that are to be modified are set to these registers, the ROM data can be replaced with modification data set in the registers. Data of up to three different addresses can be modified.

The wild register function can be used to debug a program after creating the mask and to patch bugs in the program.

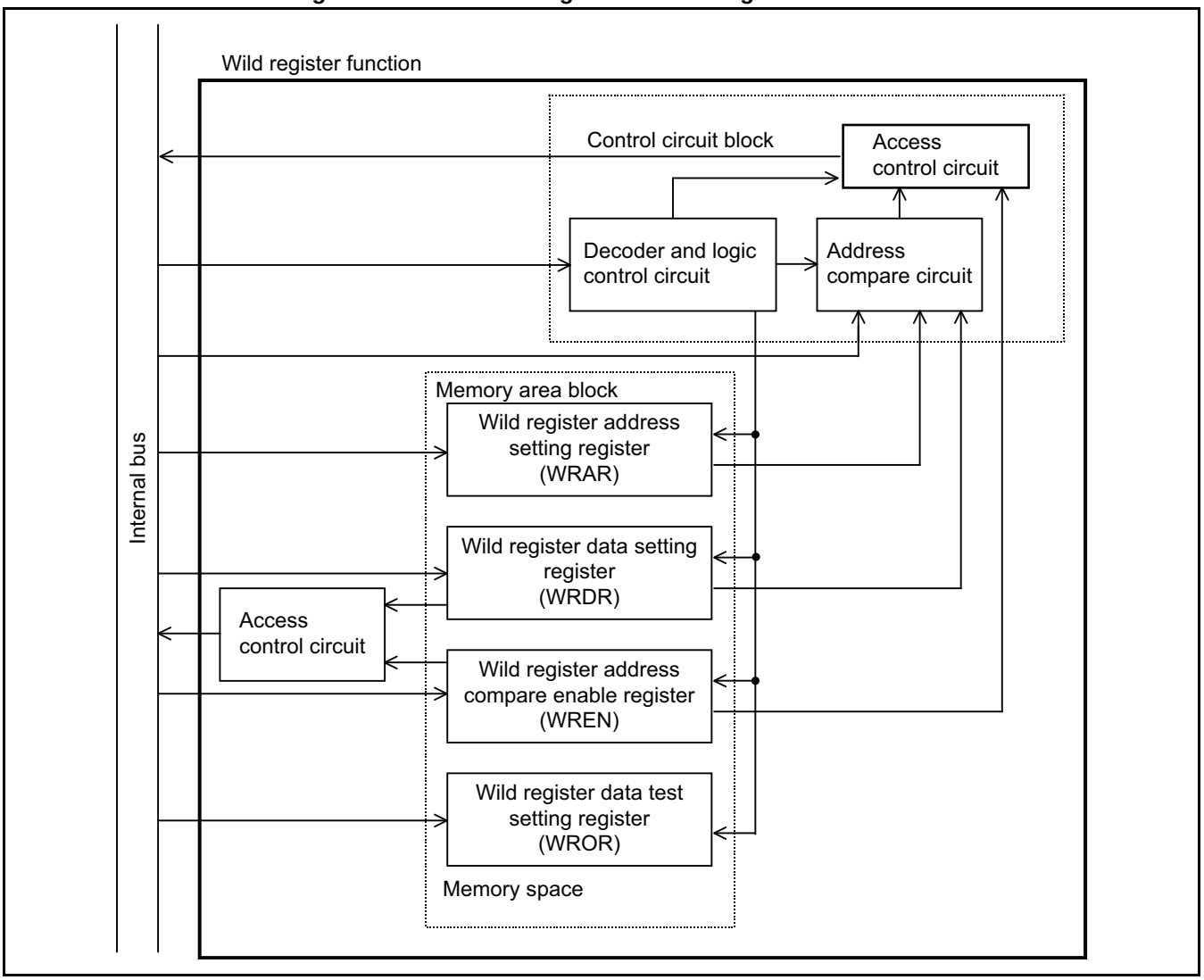
13.2 Configuration of Wild Register Function

The block diagram of the wild register is shown below. The wild register consists of the following blocks:

- **Memory area block**
  Wild register data setting register (WRDR0 to WRDR2)
  Wild register address setting register (WRAR0 to WRAR2)
  Wild register address compare enable register (WREN)
  Wild register data test setting register (WROR)
- **Control circuit block**

■ Block Diagram of Wild Register Function

Figure 13.2-1 Block Diagram of Wild Register Function



- Memory area block

  The memory area block consists of the wild register data setting registers (WRDR), wild register address setting registers (WRAR), wild register address compare enable register (WREN) and wild register data test setting 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 setting register (WRDR). In addition, the wild register data test setting register (WROR) enables the normal read function for each wild register data setting register (WRDR).

- Control circuit block

  This circuit compares the actual address data with addresses set in the wild register address setting registers (WRAR). If they match, the circuit outputs the data from the wild register data setting register (WRDR) to the data bus. The operation of the control circuit block is controlled by the wild register address compare enable register (WREN).

13.3 Registers of Wild Register Function

The registers of the wild register function include the wild register data setting registers (WRDR), wild register address setting registers (WRAR), wild register address compare enable register (WREN) and wild register data test setting register (WROR).

■ Registers of Wild Register Function

Figure 13.3-1 Registers of Wild Register Function

Wild register data setting registers (WRDR0 to WRDR2)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F82_H$	WRDR0	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
$0F85_H$	WRDR1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
$0F88_H$	WRDR2									

Wild register address setting registers (WRAR0 to WRAR2)

Address		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
$0F80_H$, $0F81_H$	WRAR0	RA15	…	…	…	…	…	…	RA8	00000000_B
$0F83_H$, $0F84_H$	WRAR1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F86_H$, $0F87_H$	WRAR2	RA7	…	…	…	…	…	…	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)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0076_H	WREN	-	-	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 setting register (WROR)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0077_H	WROR	-	-	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 (The read value is the same as the write value.)
R0/WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0/W0: Reserved bit (The write value is "0"; the read value is "0".)
-: Unused

■ Wild Register Number

A wild register number is assigned to each wild register address setting register (WRAR) and each wild register data setting register (WRDR).

Table 13.3-1 Wild Register Numbers Corresponding to Wild Register Address Setting Registers and Wild Register Data Setting Registers

Wild register number	Wild register address setting register (WRAR)	Wild register data setting register (WRDR)
0	WRAR0	WRDR0
1	WRAR1	WRDR1
2	WRAR2	WRDR2

13.3.1 Wild Register Data Setting Registers (WRDR0 to WRDR2)

The wild register data setting registers (WRDR0 to WRDR2) use the wild register function to specify the data to be amended.

■ Wild Register Data Setting Registers (WRDR0 to WRDR2)

Figure 13.3-2 Wild Register Data Setting 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 (The read value is the same as the write value.)

Table 13.3-2 Functions of Bits in Wild Register Data Setting Register (WRDR)

Bit name		Function
bit7 to bit0	RD7 to RD0: Wild register data setting 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 setting register (WRAR). Data is valid at an address corresponding to one of the wild register numbers. • The read access to one of these bits is enabled only when the data test setting bit in the wild register data test setting register (WROR) corresponding to the bit to be read is set to "1".

13.3.2 Wild Register Address Setting Registers (WRAR0 to WRAR2)

The wild register address setting registers (WRAR0 to WRAR2) set the address to be amended by the wild register function.

■ Wild Register Address Setting Registers (WRAR0 to WRAR2)

Figure 13.3-3 Wild Register Address Setting 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 (The read value is the same as the write value)

Table 13.3-3 Functions of Bits in Wild Register Address Setting Register (WRAR)

Bit name		Function
bit15 to bit0	RA15 to RA0: Wild register address setting bits	These bits set the address to be amended by the wild register function. The address to be assigned to amendment data is set to these bits. The address is to be specified according to the wild register number corresponding to a wild register address setting register.

13.3.3 Wild Register Address Compare Enable Register (WREN)

The wild register address compare enable register (WREN) enables/disables the operations of wild register functions using their respective wild register numbers.

■ Wild Register Address Compare Enable Register (WREN)

Figure 13.3-4 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	

R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0/W0 : Reserved bit (The write value is "0"; the read value is "0".)
R/W : Readable/writable (The read value is the same as the write value.)
-: Unused

Table 13.3-4 Functions of Bits in Wild Register Address Compare Enable Register (WREN)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. • When these bits are read, they always return "0". • Writing values to these bits has no effect on operation.
bit5 to bit3	Reserved bits	These bits are reserved. • When these bits are read, they always return "0". • Always set these bits to "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. **Writing "0"**: disables the operation of the wild register function. **Writing "1"**: enables the operation of the wild register function.

13.3.4 Wild Register Data Test Setting Register (WROR)

The wild register data test setting register (WROR) enables/disables reading data from the corresponding wild register data setting register (WRDR0 to WRDR2).

■ Wild Register Data Test Setting Register (WROR)

Figure 13.3-5 Wild Register Data Test Setting 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	

R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R0/W0 : Reserved bit (The write value is "0"; the read value is "0".)
R/W : Readable/writable (The read value is the same as the write value.)
-: Unused

Table 13.3-5 Functions of Bits in Wild Register Data Test Setting Register (WROR)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. • When these bits are read, they always return "0". • Writing values to these bits has no effect on operation.
bit5 to bit3	Reserved bits	These bits are reserved. • When these bits are read, they always return "0". • Always set these bits to "0".
bit2 to bit0	DRR2, DRR1, DRR0: Wild register data test setting bits	These bits enable/disable the normal reading from the corresponding data setting register of the wild register. • DRR0 enables/disables reading from the wild register data setting register (WRDR0). • DRR1 enables/disables reading from the wild register data setting register (WRDR1). • DRR2 enables/disables reading from the wild register data setting register (WRDR2). **Writing "0"**: disables reading. **Writing "1"**: enables reading.

13.4 Operations of Wild Register Function

This section describes the procedure for setting the wild register function.

■ Procedure for Setting Wild Register Function

Prepare a 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 using the wild register function. The setting method for the wild register is shown below.

This section does not include information on the method of communications 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 setting register (WRAR0 to WRAR2).
- Write a new code to the wild register data setting register (WRDR0 to WRDR2) corresponding to the wild register address setting register to which the address has been written.
- Write "1" to the EN bit in the wild register address compare enable register (WREN) corresponding to the wild register number to enable the wild register function represented by that wild register number.

Table 13.4-1 shows the procedure for setting the registers of the wild register function.

Table 13.4-1 Procedure for Setting Registers of Wild Register Function

Step	Operation	Operation example
1	Read replacement data from a peripheral function outside through a certain communication method.	Suppose the built-in ROM code to be modified is at the address $F011_H$ and the data to be modified at "$B5_H$", and there are three built-in ROM codes to be modified.
2	Write the replacement address to a wild register address setting register (WRAR0 to WRAR2).	Set wild register address setting registers (WRAR0 = $F011_H$, WRAR1 = ..., WRAR2 = ...).
3	Write a new ROM code (replacement for the built-in ROM code) to a wild register data setting register (WRDR0 to WRDR2).	Set the wild register data setting registers (WRDR0 = $B5_H$, WRDR1 =..., WRDR2 =...).
4	Enable the EN bit in the wild register address compare enable register (WREN) corresponding to the wild register number of the wild register function used.	Setting bit 0 of the address compare enable register (WREN) to "1" enables the wild register function of the wild register number 0. If the address matches the value set in the wild register address setting register (WRAR), the value of the wild register data setting register (WRDR) will be replaced with the built-in ROM code. When replacing more than one built-in ROM code, enable the related EN bits in the wild register address compare enable register (WREN) corresponding to respective built-in ROM codes.

■ Wild Register Function Applicable Addresses

The wild register function can be applied to all address space except the address "0078_H".

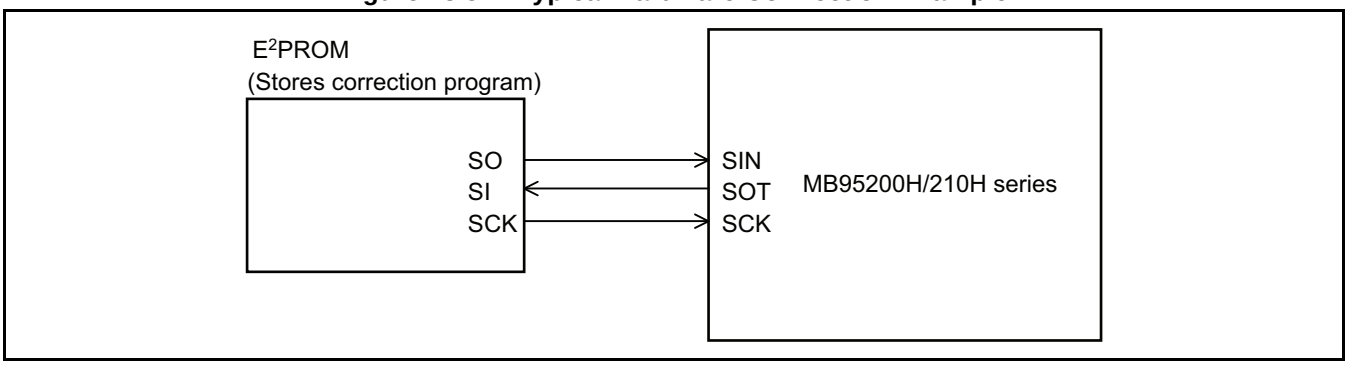
Since the address "0078_H" is used as a mirror address for the register bank pointer and the direct bank pointer, this address cannot be patched.

13.5 Typical Hardware Connection Example

Below is an example of typical hardware connection for the application of the wild register function.

■ Hardware Connection Example

Figure 13.5-1 Typical Hardware Connection Example



CHAPTER 14

8/16-BIT COMPOSITE TIMER

This chapter describes the functions and operations of the 8/16-bit composite timer.

14.1 Overview of 8/16-bit Composite Timer

The 8/16-bit composite timer consists of two 8-bit counters. It can be used as two 8-bit timers, or as a 16-bit timer if the two counters are connected in cascade.
The 8/16-bit composite 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 value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted, an interrupt request occurs, and the counter stops counting.

■ 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 value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted, an interrupt request occurs, and the counter counts from "00_H" again. The timer outputs 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 value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted and an interrupt request occurs. Under these conditions, if the counter continues to count and reaches "FF_H", it restarts counting from "00_H". The timer outputs 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 at "FF_H" in 8-bit operation or at "$FFFF_H$" in 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by setting a specific 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 of variable cycle and duty depending on the cycle and "L" pulse width specified by registers.

In this operating mode, since the two 8-bit counters have to be used separately, the composite timer cannot operate as a 16-bit counter.

■ PWC Timer Function

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured.

In this operating mode, the counter starts counting from "00_H" immediately after a count start edge of an external input signal is detected. Afterward, when a count end edge is detected, the counter transfers its value to a register to generate an interrupt.

■ Input Capture Function

When the input capture function is selected, the counter value is stored in a register immediately after the detection of an edge of an external input signal.

This function is available in either free-run mode or clear mode for count operation.

In clear mode, the counter starts counting from "00_H", and transfers its value to a register to generate an interrupt after an edge is detected. Afterward, the counter restarts counting from "00_H".

In free-run mode, the counter transfers its value to a register to generate an interrupt immediately after the detection of an edge. Afterward, unlike in clear mode, the counter continues to count without being cleared to "00_H".

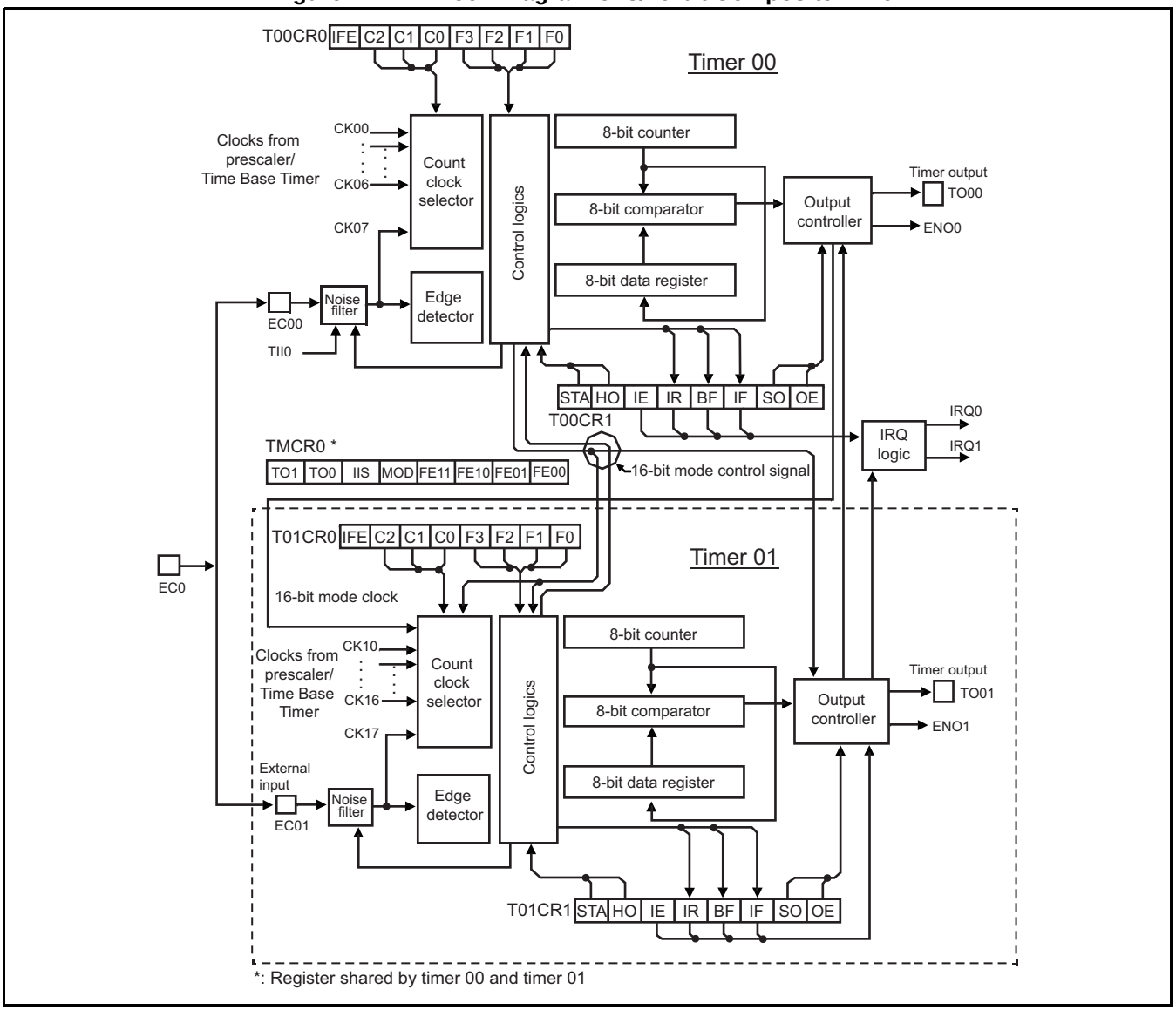
14.2 Configuration of 8/16-bit Composite Timer

The 8/16-bit composite timer consists of the following blocks:

- **8-bit counter × 2 channels**
- **8-bit comparator (including a temporary latch) × 2 channels**
- **8/16-bit composite timer 00/01 data register × 2 channels (T00DR/T01DR)**
- **8/16-bit composite timer 00/01 status control register 0 × 2 channels (T00CR0/T01CR0)**
- **8/16-bit composite timer 00/01 status control register 1 × 2 channels (T00CR1/T01CR1)**
- **8/16-bit composite 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 Composite Timer

Figure 14.2-1 Block Diagram of 8/16-bit Composite 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 value in the 8/16-bit composite timer 00/01 data register and that in the counter. It incorporates a latch that temporarily stores the 8/16-bit composite timer 00/01 data register value.

● 8/16-bit composite timer 00/01 data register

The 8/16-bit composite timer 00/01 data register is used to write the maximum value counted during interval timer operation or PWM timer operation and to read the count value during PWC timer operation

or input capture operation.

- 8/16-bit composite timer 00/01 status control registers 0 (T00CR0/T01CR0)

  These registers are used to select the timer operating mode and the count clock, and to enable or disable IF flag interrupts.

- 8/16-bit composite timer 00/01 status control registers 1 (T00CR1/T01CR1)

  These registers are used to control interrupt flags, timer output, and timer operation.

- 8/16-bit composite timer 00/01 timer mode control register (TMCR0)

  This register is used to select the noise filter function, 8-bit or16-bit operating 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 operating clock signal from different prescaler output signals (divided machine clock signal and timebase timer output signal).

- 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. The filter function can be selected from "H" pulse noise elimination, "L" pulse noise elimination, and "H"/"L"-pulse noise elimination.

- TII0 internal pin (internally connected to the LIN-UART, available only on channel 0 of MB95200H Series)

  The TII0 pin serves as the signal input pin for timer 00. Nonetheless, it is connected to the LIN-UART inside the chip. For information about how to use the pin, see "CHAPTER 16 LIN-UART". In addition, the TII0 pin in channel 1 is internally fixed to "0".

■ Input Clock

The 8/16-bit composite timer uses the output clock from the prescaler as its input clock (count clock).

14.3 Channels of 8/16-bit Composite Timer

This section describes the channels of the 8/16-bit composite timer.

■ Channels of 8/16-bit Composite Timer

The MB95200H Series contains two channels of 8/16-bit composite timer and the MB95210H Series contains one channel of 8/16-bit composite timer.

In a channel, there are two 8-bit counters. They 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 14.3-1 8/16-bit Composite 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 14.3-2 8/16-bit Composite Timer Channels and Corresponding Registers

Channel	Pin name	Pin function
0	T00CR0	Timer 00 status control register 0
	T01CR0	Timer 01 status control register 0
	T00CR1	Timer 00 status control register 1
	T01CR1	Timer 01 status control register 1
	T00DR	Timer 00 data register
	T01DR	Timer 01 data register
	TMCR0	Timer 00/01 timer mode control register
1	T10CR0	Timer 10 status control register 0
	T11CR0	Timer 11 status control register 0
	T10CR1	Timer 10 status control register 1
	T11CR1	Timer 11 status control register 1
	T10DR	Timer 10 data register
	T11DR	Timer 11 data register
	TMCR1	Timer 10/11 timer mode control register

In the following sections in this chapter, only details of channel 0 of the 8/16-bit composite timer are provided.

Channel 0 and channel 1 are the same. The 2-digit number in the pin names and register names corresponds to channel and timer. The first number represents the channel and the second one the timer.

The MB95210H Series contains one channel of 8/16-bit composite timer.

In a channel, there are two 8-bit counters. They 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 14.3-3 8/16-bit Composite 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

Table 14.3-4 8/16-bit Composite Timer Channels and Corresponding Registers

Channel	Register name	Register
0	T00CR0	Timer 00 status control register 0
	T01CR0	Timer 01 status control register 0
	T00CR1	Timer 00 status control register 1
	T01CR1	Timer 01 status control register 1
	T00DR	Timer 00 data register
	T01DR	Timer 01 data register
	TMCR0	Timer 00/01 timer mode control register

The 2-digit number in the pin names and register names corresponds to channel and timer. The first number represents the channel and the second one the timer.

14.4 Pins Related to 8/16-bit Composite Timer

This section describes the pins related to the 8/16-bit composite timer.

■ Pins Related to 8/16-bit Composite Timer

The external pins related to the 8/16-bit composite timer are TO00, TO01, and EC0. TII0 is for internal chip connection.

- TO00 pin

TO00:

This pin serves as the timer output pin for timer 00 in 8-bit operation or for timers 00 and 01 in 16-bit operation. When the output is enabled (T00CR1:OE = 1) in the interval timer function, PWM timer function, or PWC timer function, this pin becomes an output pin automatically regardless of the port direction register (DDR2:bit2) and functions as the timer output TO00 pin.

The output becomes undetermined if output is enabled with the input capture function in use.

- TO01 pin

TO01:

This pin serves as the timer output pin for timer 01 in 8-bit operation. When the output is enabled (T01CR1:OE = 1) in interval timer, PWM timer function (fixed-cycle mode), or PWC timer function, the pin becomes an output pin automatically regardless of the port direction register (DDR2:bit3) and functions as the timer output TO01 pin.

In 16-bit operation, if output is enabled with the PWM timer function (variable-cycle mode) or input capture function in use, the output becomes undetermined.

- 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 function or PWM timer function is selected, or as the signal input pin for timer 00 when the PWC timer function or input capture function is selected. The pin cannot be set to serve as the external count clock input pin when the PWC timer function or input capture function is selected.

To use the input function mentioned above, set the bit in the port direction register corresponding to EC0 pin to "0" to make 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 function or PWM timer function is selected, or as the signal input pin for timer 01 when the PWC timer function or input capture function is selected. The pin cannot be set to serve as the external count clock input pin when the PWC timer function or input capture function is selected.

In 16-bit operation, the input function of this pin is not used. If the PWM timer function (variable-cycle mode) is selected, the input function of this pin can also be used.

To use the input function mentioned above, set the bit in the port direction register corresponding to EC0 pin to "0" to make the pin as an input port.

■ Block Diagrams of Pins Related to 8/16-bit Composite Timer

Figure 14.4-1 Block Diagram of Pin EC0 (P12/EC0/DBG) Related to 8/16-bit Composite Timer

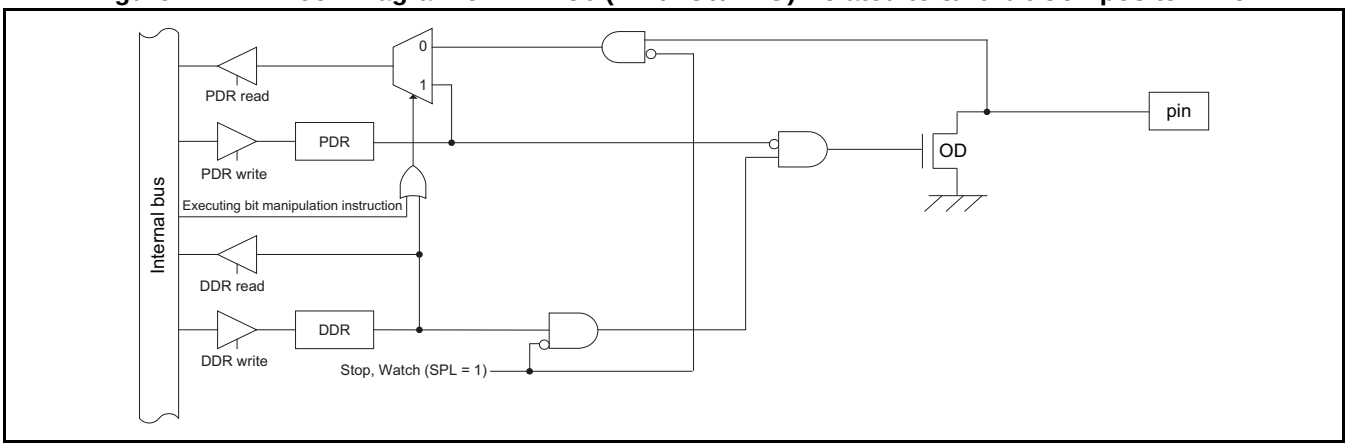


Figure 14.4-2 Block Diagram of Pin EC0 (P04/INT04/AN04/SIN/HCLK1/EC0) Related to 8/16-bit Composite Timer

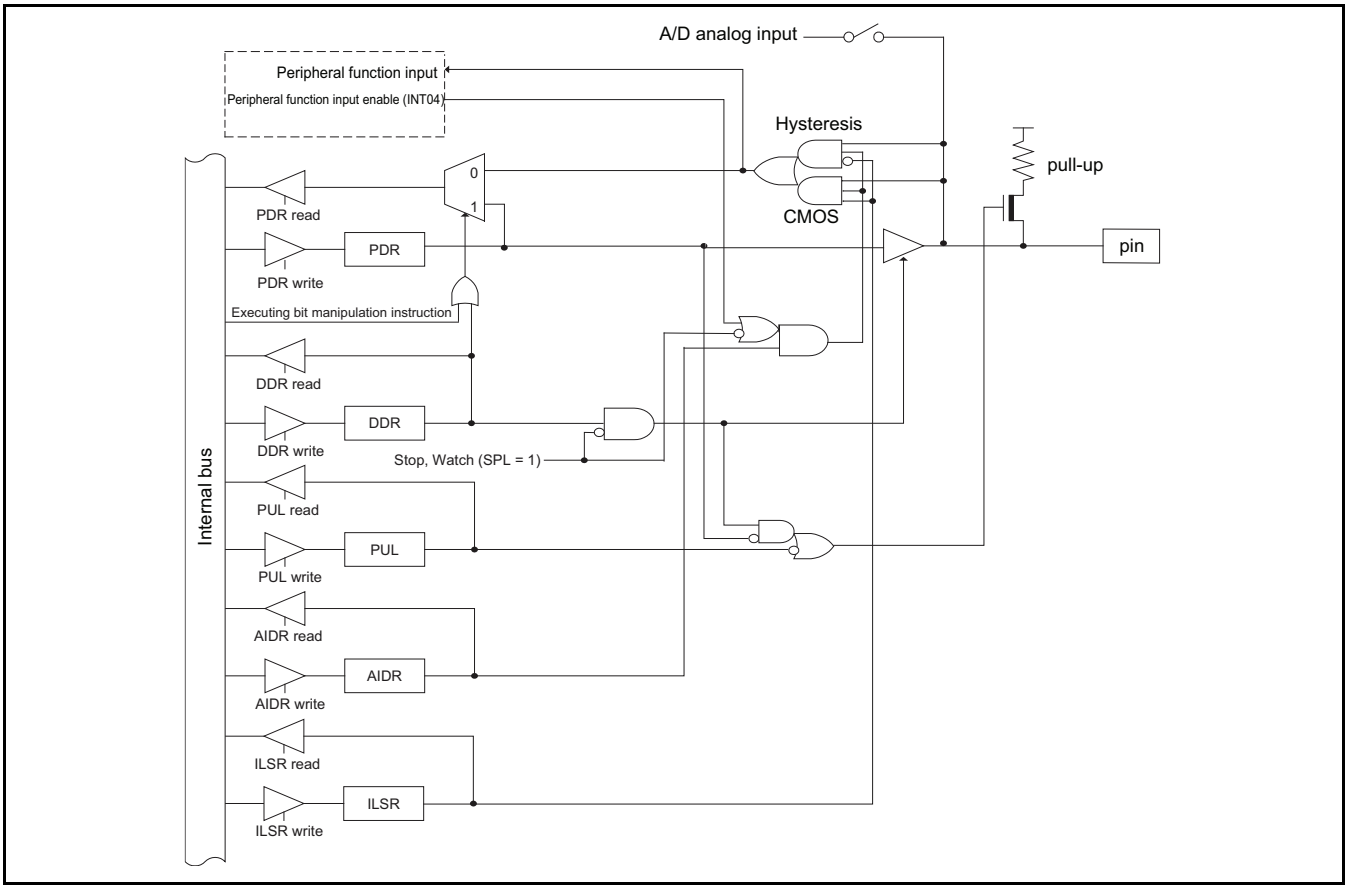


Figure 14.4-3 Block Diagram of Pin TO00 (P05/INT05/AN05/TO00/HCLK2) Related to 8/16-bit Composite Timer

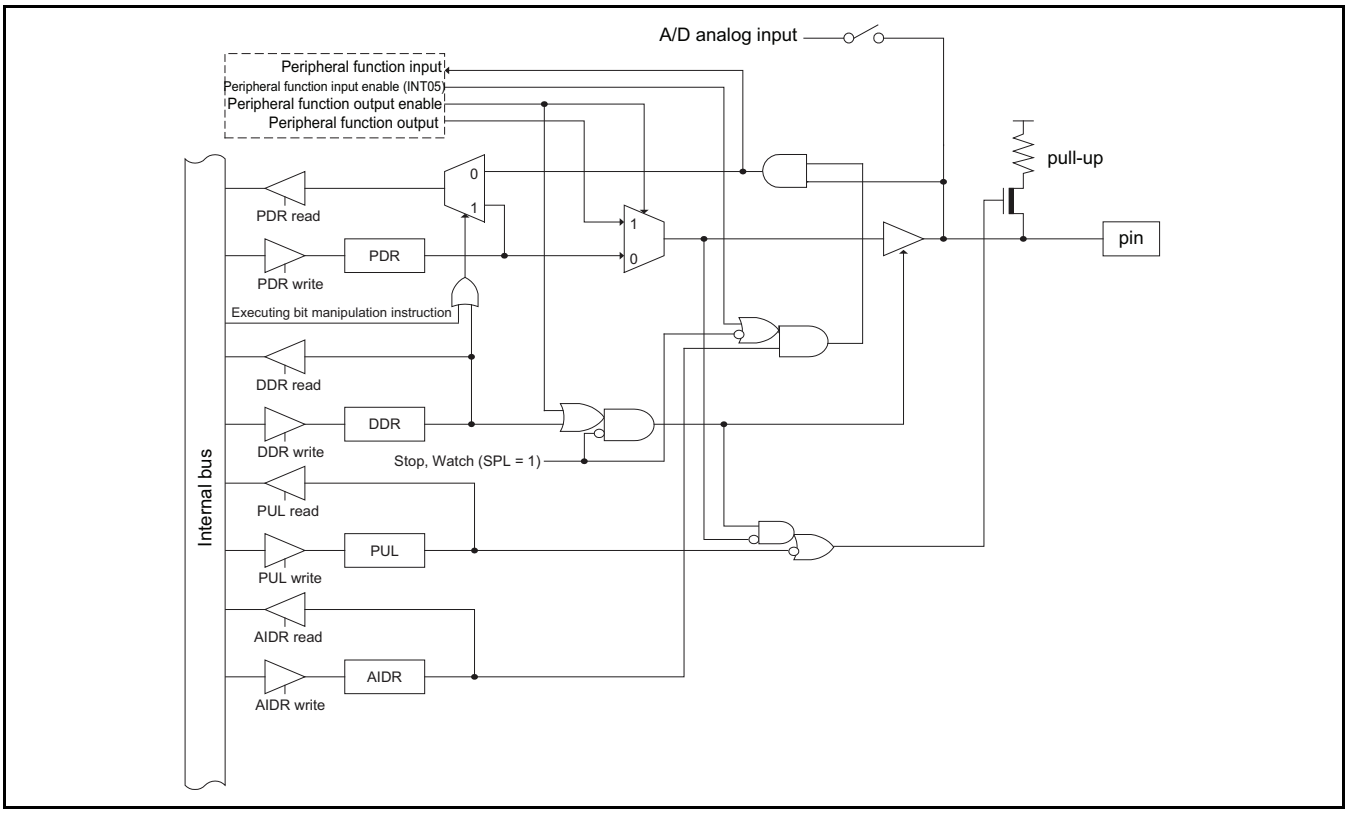


Figure 14.4-4 Block Diagram of Pin TO01 (P06/INT06/TO01) Related to 8/16-bit Composite Timer

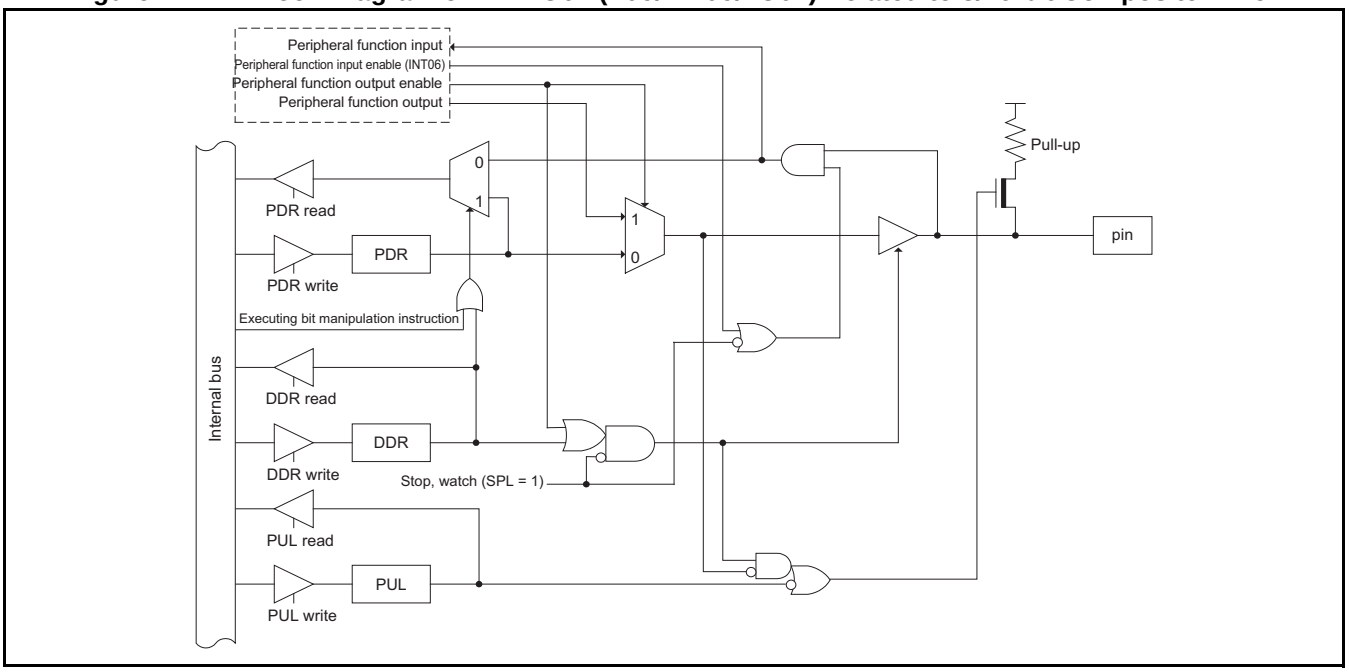


Figure 14.4-5 Block Diagram of Pins TO10, TO11 (P62/TO10, P63/TO11) Related to 8/16-bit Composite Timer

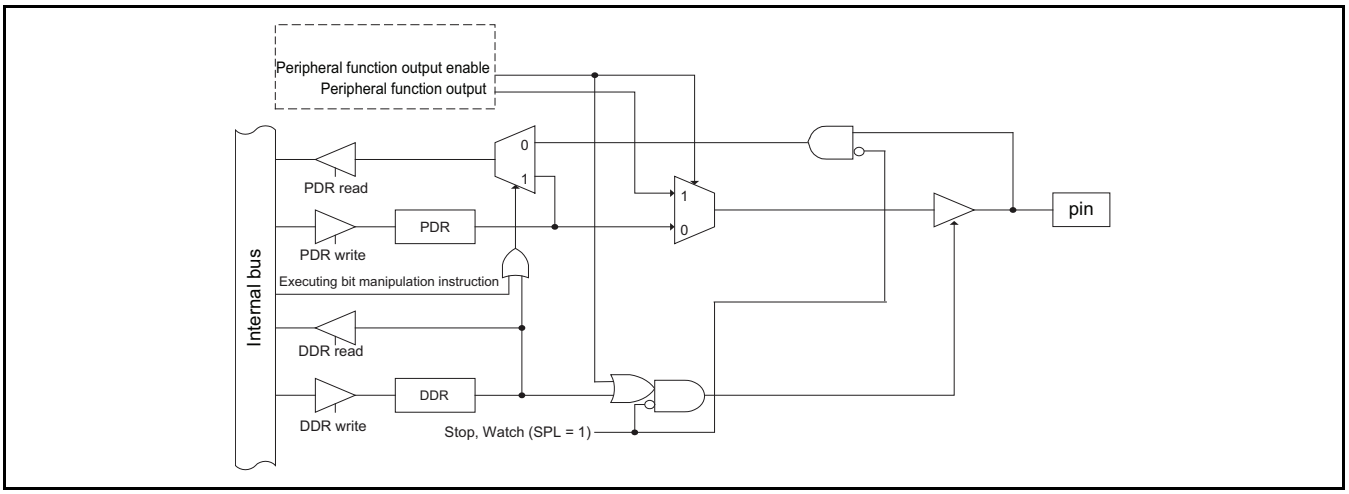
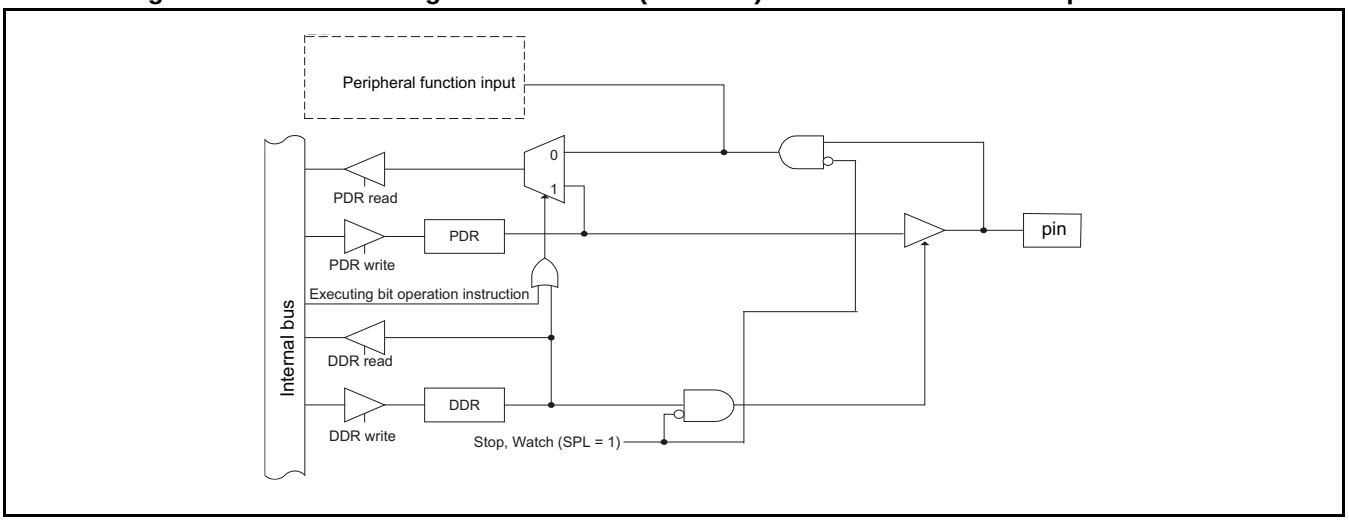


Figure 14.4-6 Block Diagram of Pin EC1 (P64/EC1) Related to 8/16-bit Composite Timer



14.5 Registers of 8/16-bit Composite Timer

This section describes the registers of the 8/16-bit composite timer.

■ Registers of 8/16-bit Composite Timer

Figure 14.5-1 Registers Of 8/16-bit Composite Timer

8/16-bit composite timer 00/01 status control register 0 (T00CR0/T01CR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
$0F92_H$ T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0	00000000_B
$0F93_H$ T00CR0	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	

8/16-bit composite timer 00/01 status control register 1 (T00CR1/T01CR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
0036_H T01CR1	STA	HO	IE	IR	BF	IF	SO	OE	00000000_B
0037_H T00CR1	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

8/16-bit composite timer 00/01 data register (T00DR/T01DR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
$0F94_H$ T01DR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000_B
$0F95_H$ T00DR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit composite timer 00/01 timer mode control register (TMCR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
$0F96_H$ TMCR0	T01	T00	TIS	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 (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R,W : Readable/writable (The read value is different from the write value.)

14.5.1 8/16-bit Composite Timer 00/01 Status Control Register 0 (T00CR0/T01CR0)

The 8/16-bit composite timer 00/01 status control 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 Composite Timer 00/01 Status Control Register 0 (T00CR0/T01CR0)

Figure 14.5-2 8/16-bit Composite Timer 00/01 Status Control Register 0 (T00CR0/T01CR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F92H T01CR0 0F93H T00CR0	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 (The read value is the same as the write value.)
: Initial value

Table 14.5-1 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control 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.
Writing "0": disables IF flag interrupts.
Writing "1": an IF flag interrupt request is output 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. See "6.12 Operation of Prescaler".
• Write access to these bits is nullified in timer operation (T00CR1/T01CR1:STA = 1).
• The clock selection of T01CR0 (timer 01) is nullified in 16-bit operation.
• These bits cannot be set to "111_B" when the PWC function or input capture function is used. An attempt to write "111_B" with the PWC function 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 14.5-1 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control 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 operating mode select bits</td><td>These bits select the timer operating 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. If one of the timers starts operating (T00CR1/T01CR1: STA= 1), the F3, F2, F1 and F0 bits of the other timer are automatically set to "0100_B".
• With the 16-bit operation having been selected (TMCR0:MOD = 1), if the composite timer starts operating using the PWM timer function (variable-cycle mode) (T00CR1/T01CR1:STA = 1), the MOD bit is set to "0" automatically.
• Write access to these bits is nullified in timer operation (T00CR1/T01CR1:STA = 1).
<table>
<tr><th>F3</th><th>F2</th><th>F1</th><th>F0</th><th>Timer operating 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 rising)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>PWC timer (cycle = falling to falling)</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>

14.5.2 8/16-bit Composite Timer 00/01 Status Control Register 1 (T00CR1/T01CR1)

The 8/16-bit composite timer 00/01 status control 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 Composite Timer 00/01 Status Control Register 1 (T00CR1/T01CR1)

Figure 14.5-3 8/16-bit Composite Timer 00/01 Status Control Register 1 (T00CR1/T01CR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0036H T01CR1 0037H T00CR1	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

OE	
0	Disables timer output
1	Enables timer output

SO — Timer output initial value bit

SO	
0	Timer initial value "0"
1	Timer initial value "1"

IF — Timer reload/overflow flag

IF	Read	Write
0	No reload or overflow	Flag clear
1	Reload and overflow	No effect on operation

BF — Data register full flag

BF	
0	No measurement data in data register
1	Measurement data present in data register

IR — Pulse width measurement complete and edge detection flag

IR	Read	Write
0	Measurement complete, edge undetected	Flag clear
1	Measurement complete, edge detected	No effect on operation

IE — Interrupt request bit

IE	
0	Disables interrupt requests
1	Enables interrupt requests

HO — Timer suspend bit

HO	
0	Resumes timer operation
1	Suspends timer

STA — Timer operation enable bit

STA	
0	Stops timer
1	Enables timer

R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
: Initial value

Table 14.5-2 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control Register 1 (1 / 2)

Bit name		Function
bit7	STA: Timer operation enable bit	This bit enables or stops the timer operation. **Writing "0"**: stops the timer operation and sets the count value to "00_H". • With the PWM timer function (variable-cycle mode) in use (T00CR0/T01CR0: F3, F2, F1, F0 = 0100_B), the STA bit in either the T00CR1 (timer 00) or the T01CR1 (timer 01) register can be used to enable or disable the timer operation. If the STA bit in one of the registers is set to "0", the STA bit in the other one is automatically set to the same value. • During 16-bit operation (TMCR0:MD = 1), use the STA bit in the T00CR1 (timer 00) register to enable or disable timer operation. If the STA bit of one of the timers is se to "0", the STA bit in the other one is automatically set to the same value. **Writing "1"**: allows timer operation to start from count value "00_H". • Before setting this bit to "1", set 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:MD), and filter function select bits (TMCR0:FE11, FE10, FE01, FE00).
bit6	HO: Timer suspend bit	This bit suspends or resumes the timer operation. • Writing "1" to this bit during timer operation suspends the timer operation. • When the timer operation has been enabled (T00CR1/T01CR1:STA = 1), writing "0" to the bit resumes the timer operation. • With the PWM timer function (variable-cycle mode) in used (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. If the HO bit in one of the registers is set to "0" or "1", the HO bit in the other one is automatically set to the same value. • In 16-bit operation (TMCR0:MD = 1), use the HO bit in the T00CR1 (timer 00) register to suspend or resume timer operation. If the HO bit in one of the registers is set to "0" or "1", the HO bit in the other one is automatically set to the same value.
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". However, an interrupt request from the timer reload/overflow flag (T00CR1/T01CR1:IF) is not output 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 indicates the completion of pulse width measurement or the detection of an edge. • With the PWC timer function in use, this bit is set to "1" immediately after pulse width measurement is complete. • With the input capture function in use, this bit is set to "1" immediately after an edge is detected. • The bit is set to "0" when the function of the composite timer selected is neither the PWC timer function nor the input capture function. • If this bit is read by the read-modify-write (RMW) type of instruction, it always returns "1". • The IR bit in the T01CR1 (timer 01) register is set to "0" in 16-bit operation. • Writing "0" to this bit sets it to "0". • Writing "1" to this bit is ignored.

Table 14.5-2 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control Register 1 (2 / 2)

Bit name		Function
bit3	BF: Data register full flag	• With the PWC timer function in use, this bit is set to "1" when a count value is stored in the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) immediately after pulse width measurement is complete. • In 8-bit operation, this bit is set to "0" when the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is read. • The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) holds data if this bit is set to "1". With this bit being "1", even when the next edge is detected, the count value is not transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), and the next measurement result is thus lost. Nonetheless, there is an exception. With the F3 bit to F0 bit in the T00CR0/T01CR0 register having been set to "1001_B", even though the BF bit is set to "1", the "H" pulse measurement result is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not 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 is selected. • Writing a value to this bit has no effect on operation.
bit2	IF: Timer reload/overflow flag	This bit is used to detect the count value match and the counter overflow. • With the interval timer function (one-shot or continuous) or the PWM timer function (variable-cycle mode) in use, this bit is set to "1" if the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) value matches the count value. • With the PWC timer function of the input capture function in use, this bit is set to "1" if a counter overflow occurs. • If this bit is read by a read-modify-write (RMW) instruction, it always returns "1". • Writing "0" to this bit sets it to "0". • Writing "1" to this bit has no effect on operation. • The bit becomes "0" if the PWM function (variable-cycle mode) is selected. • The IF bit in the T01CR1 (timer 01) register is "0" in 16-bit operation.
bit1	SO: Timer output initial value bit	The timer output (TMCR0:TO1/TO0) initial value is set by writing a value to this bit. 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". • In 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 SO bit in the other one has no effect on operation. • During timer operation (T00CR1:STA = 1 or T01CR1:STA = 1), the write access to this bit is invalid. However, in 16-bit operation, although a value can be written to the SO bit in the T01CR1 (timer 01) register even during timer operation, the value written has no direct effect on the timer output. • When the PWM timer function (fixed cycle mode or variable cycle mode) or the input capture function is in use, the value of this bit has no effect on operation.
bit0	OE: Timer output enable bit	This bit enables or disables timer output. **Writing "0"**: no timer output is supplied to the external pin. In this case, the external pin serves as a general-purpose port. **Writing "1"**: the time output (TMCR0:TO1/TO0) is supplied to the external pin.

14.5.3 8/16-bit Composite Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

The 8/16-bit composite timer 00/01 timer mode control register ch.0 (TMCR0) selects the filter function, 8-bit or 16-bit operating mode, and signal input to timer 00 and indicates the timer output value. This register serves both timer 00 and timer 01.

■ 8/16-bit Composite Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

Figure 14.5-4 8/16-bit Composite Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

TMCR0 Address 0F96H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
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 operating 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
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 (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
: Initial value

*1: The EC0 input can be assigned to P12 or P04 by setting the SYSC register. For details, see "CHAPTER 23 CLOCK & RESET SYSTEM CONFIGURATION CONTROLLER".

Table 14.5-3 Functions of Bits in 8/16-bit Composite 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 timer function selected.
• Writing a value to this bit has no effect on operation.
• In 16-bit operation, if the PWM timer function (variable-cycle mode) or the input capture function is selected, the value in the bit becomes undefined.
• With the interval timer function or the PWC timer function having been selected, if the timer stops operating (T00CR1/T01CR1:STA = 0), this bit holds the last value.
• With the PWM timer function (variable-cycle mode) having been selected, if the timer stops operating (T00CR1/T01CR1:STA = 0), this bit holds the last value.
• When the timer operating mode select bits (T00CR0/T01CR0: F3, F2, F1, F0) are modified with the timer stopping operating, this bit indicates the last value of timer operation if the same timer operation has been performed; otherwise it indicates "0", its initial value.</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 a value to this bit has no effect on operation.
• If the input capture function is selected, the value in the bit becomes undefined.
• With the interval timer function or the PWM timer (variable-cycle mode) or the PWC timer function having been selected, if the timer stops operating (T00CR1/T01CR1:STA = 0), this bit holds the last value.
• With the PWM timer function (variable-cycle mode) having been selected, if the timer stops operating (T00CR1/T01CR1:STA = 0), this bit holds the last value.
• When the timer operating mode select bits (T00CR0/T01CR0: F3, F2, F1, F0) are modified with the timer stopping operating, this bit indicates the last value of timer operation if the same timer operation has been performed; otherwise it indicates "0", its initial value.</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 function or input capture function is 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.
The EC0 input can be assigned to P12 or P04 by setting the SYSC register. For details, see "23.2 System Configuration Register (SYSC)" in "CHAPTER 23".</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.
• While this bit is "1", if the timer starts operating (T00CR1/T01CR1:STA = 1) with the PWM timer function (variable-cycle mode), this bit is automatically set to "0".
• During timer operation (T00CR1:STA = 1 or T01CR1:STA = 1), the write access to this bit is invalid.</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 function or the input capture function is selected.
<table>
<tr><th>FE11</th><th>FE10</th><th>Timer 01 filter</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>
• During timer operation (T00CR1:STA = 1), the write access to these bits is invalid.
• The settings of the bits have no effect on operation when the interval timer function or the PWM timer function is selected (the filter function does not operate.).</td></tr>
</table>

Table 14.5-3 Functions of Bits in 8/16-bit Composite 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 function or the input capture function is selected.
<table>
<tr><th>FE01</th><th>FE00</th><th>Timer 00 filter</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>
• During timer operation (T00CR1:STA = 1), the write access to these bits is invalid.
• The settings of these bits have no effect on operation when the interval timer function or the PWM timer function is selected (the filter function does not operate.).</td></tr>
</table>

14.5.4 8/16-bit Composite Timer 00/01 Data Register ch.0 (T00DR/T01DR)

The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is used to set the maximum count value during the interval timer operation or the PWM timer operation and to read the count value during the PWC timer operation or the input capture operation. The T00DR and T01DR registers correspond to timers 00 and 01 respectively.

■ 8/16-bit Composite Timer 00/01 Data Register (T00DR/T01DR)

Figure 14.5-5 8/16-bit Composite Timer 00/01 Data Register (T00DR/T01DR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0F94_H$ T01DR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000_B
$0F95_H$ T00DR	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R/W	

R,W: Readable/writable (The read value is different from the write value.)

● Interval timer function

The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is used to set the interval time. When the timer starts operating (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 counter returns to "00_H" and continues 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 timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

● PWM timer functions (fixed-cycle)

The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is used to set "H" pulse width time. When the timer starts operating (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 transferred to 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 timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

- PWM timer functions (variable-cycle)

  The 8/16-bit composite timer 00 data register (T00DR) and 8/16-bit composite timer 01 data register (T01DR) are used to set "L" pulse width timer and cycle respectively. When the timer starts operating (T00CR1/T01CR1:STA = 1), the value of each register is transferred to the latch in the 8-bit comparator and the two counters start counting from timer output "L". When the T00DR value transferred to the latch matches the timer 00 counter value, the timer output becomes "H" and the counting continues until the T01DR value transferred to the latch matches the timer 01 counter value. When the T01DR value transferred to the latch of the 8-bit comparator matches the timer 01 counter value, the values of the T00DR register and the T01DR register 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 timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

- PWC timer function

  The 8/16-bit composite 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 composite timer 00/01 data register is read, the BF bit is set to "0". While the BF bit is "1", no data is transferred to the 8/16-bit composite timer 00/01 data register.

  There is an exception. With the F3 bit to F0 bit in the T00CR0/T01CR0 register having been set to "1001_B", even though the BF bit is set to "1", the "H" pulse measurement result is transferred to the 8/16-bit composite timer 00/01 data register, while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse.

  When reading the 8/16-bit composite timer 00/01 data register, ensure that the BF bit is not cleared accidentally.

  If new data is written to the 8/16-bit composite timer 00/01 data register, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, set the upper timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

- Input capture function

  The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is used to read input capture results. When an edge specified is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register.

  If new data is written to the 8/16-bit composite timer 00/01 data register, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, set the upper timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

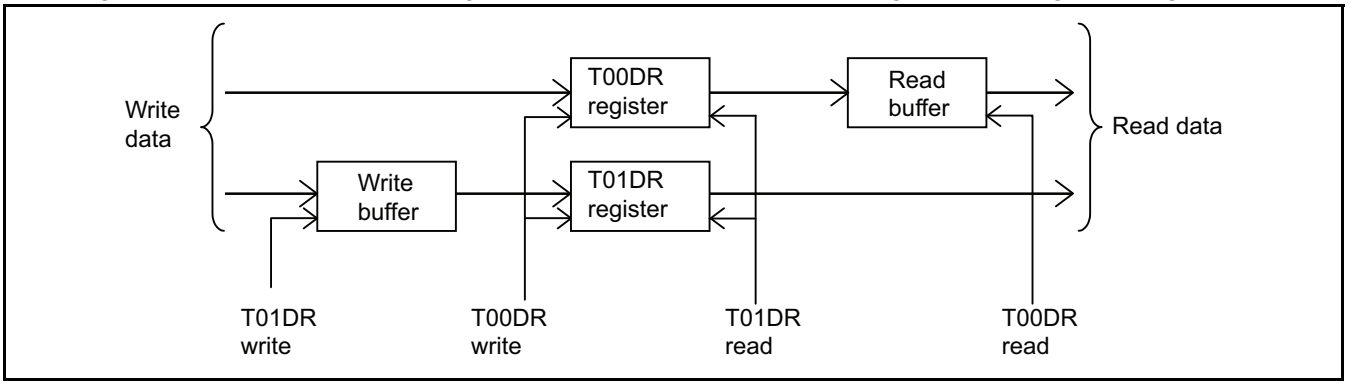
● Read and write operations

Read and write operations of T00DR and T01DR are performed in the following manner in 16-bit operation in which the PWM timer function (variable-cycle) is selected.

- Read from T01DR: In addition to the read access to T01DR, the value of T00DR is also stored in the internal read buffer at the same time.
- Read from T00DR: The internal read buffer is read.
- Write to T01DR: Data is written to the internal write buffer.
- Write to T00DR: In addition to the write access to T00DR, the value of the internal write buffer is stored in T01DR at the same time.

Figure 14.5-6 shows the T00DR and T01DR registers read from and written to during 16-bit operation.

Figure 14.5-6 Read and write operations of T00DR and T01DR registers during 16-bit operation



14.6 Interrupts of 8/16-bit Composite Timer

The 8/16-bit composite timer generates the following types of interrupts. An interrupt number and an interrupt vector are assigned to each type of interrupts.

- **Timer 00 interrupt**
- **Timer 01 interrupt**

■ Timer 00 Interrupt

Table 14.6-1 shows the timer 00 interrupt and its sources.

Table 14.6-1 Timer 00 Interrupt

Item	Description		
Interrupt generating condition	Comparison match in the interval timer operation or the PWM timer operation (variable-cycle mode)	Overflow in the PWC timer operation or the input capture operation	Completion of measurement in the PWC timer operation or edge detection in the input capture operation
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 14.6-2 shows the timer 01 interrupt and its sources.

Table 14.6-2 Timer 01 Interrupt

Item	Description		
Interrupt generating condition	Comparison match in the interval timer operation or the PWM timer operation (variable-cycle mode), except in 16-bit operation	Overflow in the PWC timer operation or the input capture operation, except in 16-bit operation	Completion of measurement in the PWC timer operation or edge detection in the input capture operation, except in 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 Table Addresses Related to Interrupts of 8/16-bit Composite Timer

Table 14.6-3 MB95200H Series Registers and Vector Table Addresses Related to Interrupts of 8/16-bit Composite Timer

Interrupt source	Interrupt request no.	Interrupt level setting 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	L022	$FFCE_H$	$FFCF_H$
Timer 11	IRQ14	ILR3	L014	$FFDE_H$	$FFDF_H$

Table 14.6-4 MB95210H Series Registers and Vector Table Addresses Related to Interrupts of 8/16-bit Composite Timer

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Timer 00	IRQ5	ILR1	L05	$FFF0_H$	$FFF1_H$
Timer 01	IRQ6	ILR1	L06	$FFEE_H$	$FFEF_H$

The interrupt request numbers and vector table addresses of all peripheral functions are listed in Appendix B "Interrupt Source Tables".

14.7 Operation of Interval Timer Function (One-shot Mode)

This section describes the operation of the interval timer function (one-shot mode) of the 8/16-bit composite timer.

■ Operation of Interval Timer Function (One-shot Mode)

The register settings shown in Figure 14.7-1 are required to use the interval timer function.

Figure 14.7-1 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 time (counter compare value)							

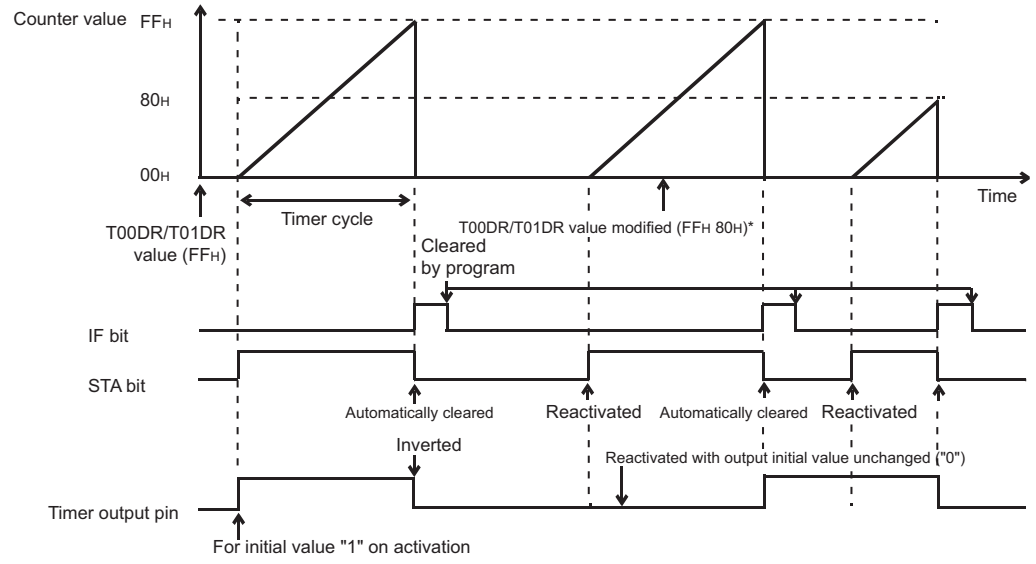
❍: Used bit
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the 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 composite 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", the start bit (T00CR0/T00CR1:STA) is set to "0", and the counter stops counting.

The value of the 8/16-bit composite 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. Do not write "00_H" to the 8/16-bit composite timer 00/01 data register.

Figure 14.7-2 shows the operation of the interval timer function in 8-bit operation.

Figure 14.7-2 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.

14.8 Operation of Interval Timer Function (Continuous Mode)

This section describes the interval timer function (continuous mode operation) of the 8/16-bit composite timer.

■ Operation of Interval Timer Function (Continuous Mode)

The register settings shown in Figure 14.8-1 are required to use interval timer function (continuous mode).

Figure 14.8-1 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)							

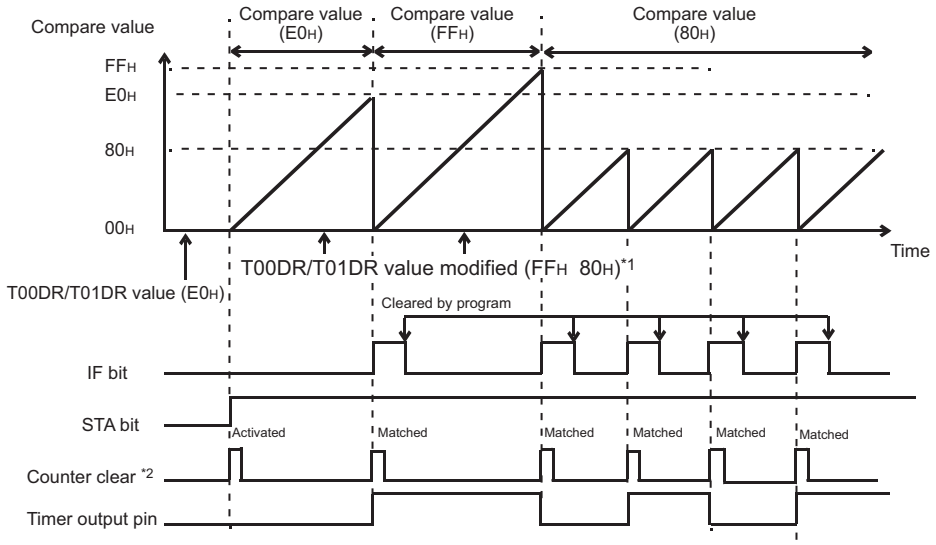
❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the 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 composite 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 returns to "00_H" and restarts counting. The timer outputs square wave as a result of this continuous operation.

The value of the 8/16-bit composite 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. Do not write "00_H" to the 8/16-bit composite timer 00/01 data register is disabled while the counter is counting.

When the timer stops operating, the timer output bit (TMCR0:TO0/TO1) holds the last value.

Figure 14.8-2 Operatiom 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 whenever a match is detected during operation.

14.9 Operation of Interval Timer Function (Free-run Mode)

This section describes the operation of the interval timer function (free-run mode) of the 8/16-bit composite timer.

■ Operation of Interval Timer Function (Free-run Mode)

The settings shown in Figure 14.9-1 are required to use the interval timer function (free-run mode).

Figure 14.9-1 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)							

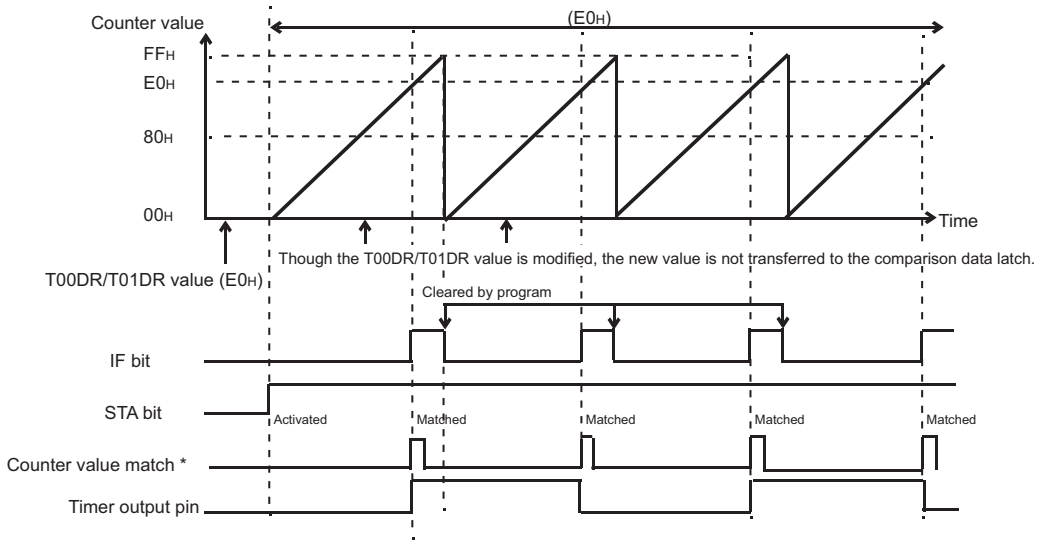
❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the 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 composite 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". If the counter continues to count with the above settings and then reaches "FF_H", it returns to "00_H" and restarts counting. The timer outputs square wave as a result of this continuous operation.

The value of the 8/16-bit composite 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. Do not write "00_H" to the 8/16-bit composite timer 00/01 data register.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

Figure 14.9-2 Operation Diagram of Interval Timer Function (Free-run Mode)



*: Even though a match is detected during operation, the counter is not cleared and the data register settings are not reloaded into the comparison data latch.

14.10 Operation of PWM Timer Function (Fixed-cycle mode)

This section describes the operation of the PWM timer function (fixed-cycle mode) of the 8/16-bit composite timer.

■ Operation of PWM Timer Function (Fixed-cycle Mode)

The settings shown in Figure 14.10-1 are required to use the PWM timer function (fixed-cycle mode).

Figure 14.10-1 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)							

❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the PWM timer function (fixed-cycle mode), PWM signal that has a fixed cycle and variable "H" pulse width is output from the timer output pin (TO00/TO01). The cycle is fixed at "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 composite timer 00/01 data register (T00DR/T01DR).

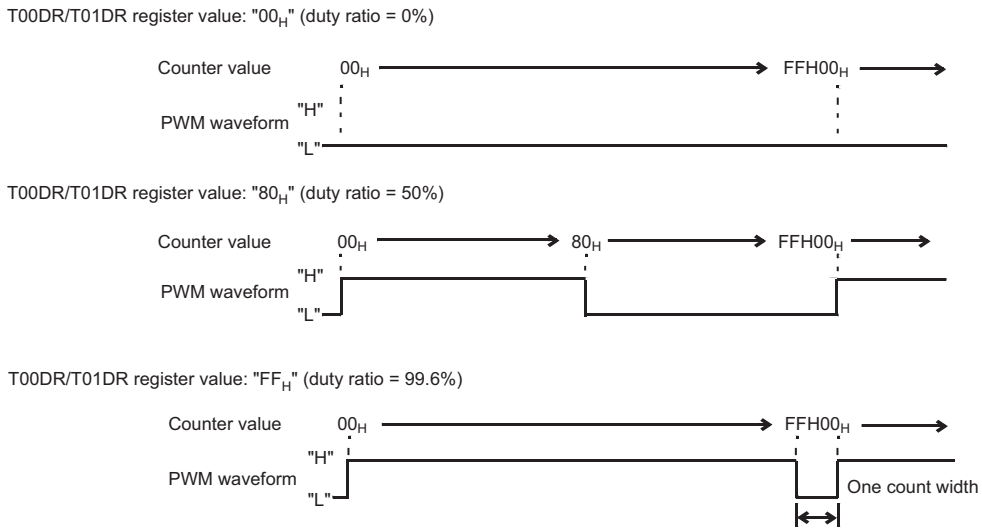
This function has no effect on the interrupt flag (T00CR1/T01CR1:IF). Since each cycle always starts with "H" pulse output, the timer output initial value setting bit (T00CR1/T01CR1:SO) has no effect on operation.

The value of the 8/16-bit composite 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), the "H" pulse is one count clock shorter than the value set in the T00DR/T01DR register.

Figure 14.10-2 Operation Diagram of PWM Timer Function (Fixed-cycle Mode)



Note: When the PWM function has been selected, the timer output pin holds the level at the point when the counter stops (T00CR0/T01CR0:STA = 0).

14.11 Operation of PWM Timer Function (Variable-cycle Mode)

This section describes the operation of the PWM timer function (variable-cycle mode) of the 8/16-bit composite timer.

■ Operation of PWM Timer Function (Variable-cycle Mode)

The settings shown in Figure 14.11-1 are required to use the PWM timer function (variable-cycle mode).

Figure 14.11-1 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)							

❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the PWM timer function (variable-cycle mode), both timers 00 and 01 are used. PWM signal of any cycle and of any duty is output from the timer output pin (TO00). The cycle is specified by the 8/16-bit composite timer 01 data register (T01DR), and the "L" pulse width is specified by the 8/16-bit composite timer 00 data register (T00DR).

Since both the 8-bit counters are used for this function, the composite timer cannot form a 16-bit counter.

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) has no effect on operation.

An interrupt flag (T00CR1/T01CR1:IF) is set when the 8-bit counter corresponding to that interrupt flag matches the value in its corresponding 8/16-bit composite timer 00/01 data register (T00DR/T01DR).

The 8/16-bit composite 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.

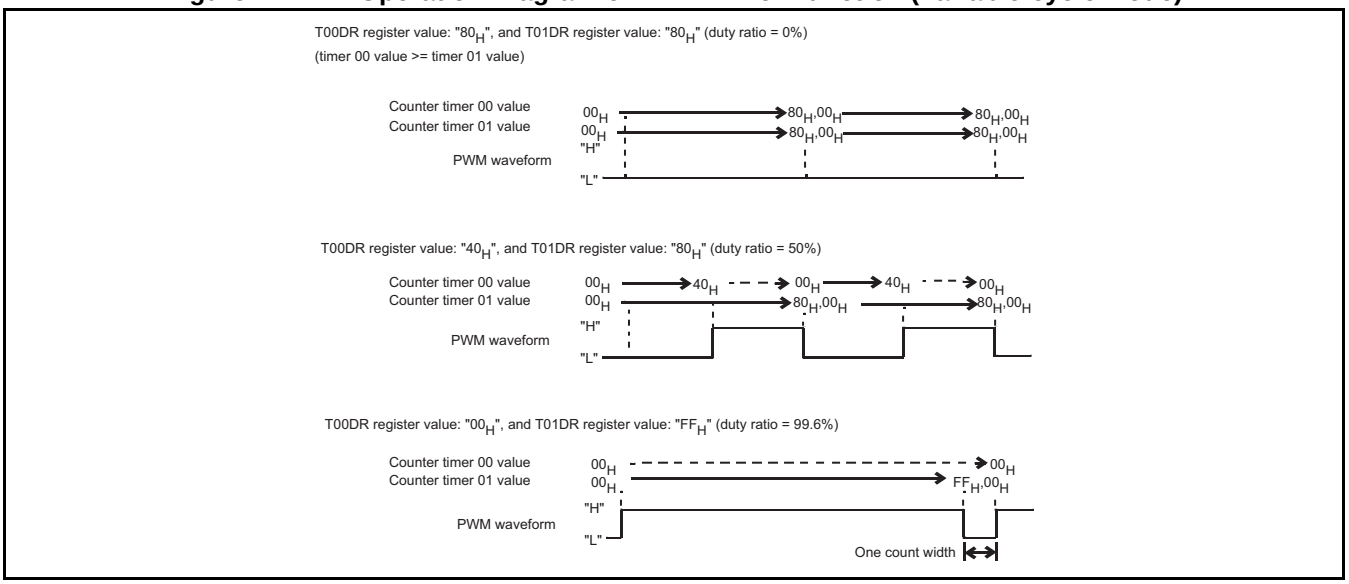
"H" is not output when the "L" pulse width setting value is greater than the cycle setting value.

The count clock must be selected for both of timers 00 and 01. Selecting different count clocks for the two timers is prohibited.

When the timer stops operating, the timer output bit (TMCR0:TO0) holds the last output value.

If the 8/16-bit composite timer 00/01 data register is modified during operation, the data written will become valid from the cycle immediately after the detection of a synchronous match.

Figure 14.11-2 Operation Diagram of PWM Timer Function (Variable-cycle Mode)



14.12 Operation of PWC Timer Function

This section describes the operation of the PWC timer function of the 8/16-bit composite timer.

■ Operation of PWC Timer Function

The settings shown in Figure 14.12-1 are required to use the PWC timer function.

Figure 14.12-1 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							

❍: Bit to be used
×: Unused bit
1: Set to "1"

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured. The edges at which counting starts and ends are selected by the timer operating mode select bits (T00CR0/T01CR0:F3, F2, F1, F0).

In the operation of this function, the counter starts counting from "00_H" immediately after a specified count start edge of an external input signal is detected. Upon the detection of a specified count end edge, the count value is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), and the interrupt flag (T00CR1/T01CR1:IR) and the buffer full flag (T00CR1/T01CR1:BF) are set to "1". The buffer full flag is set to "0" when the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is read.

If the buffer full flag is set to "1", the 8/16-bit composite timer 00/01 data register holds data. Even if the next edge is detected during that time, the next measurement result is lost since the count value has not been transferred to the 8/16-bit composite timer 00/01 data register.

There is an exception. With the F3 bit to F0 bit in the T00CR0/T01CR0 register having been set to "1001_B", even though the BF bit is set to "1", the "H" pulse measurement result is transferred to the 8/16-bit composite timer 00/01 data register, while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse.

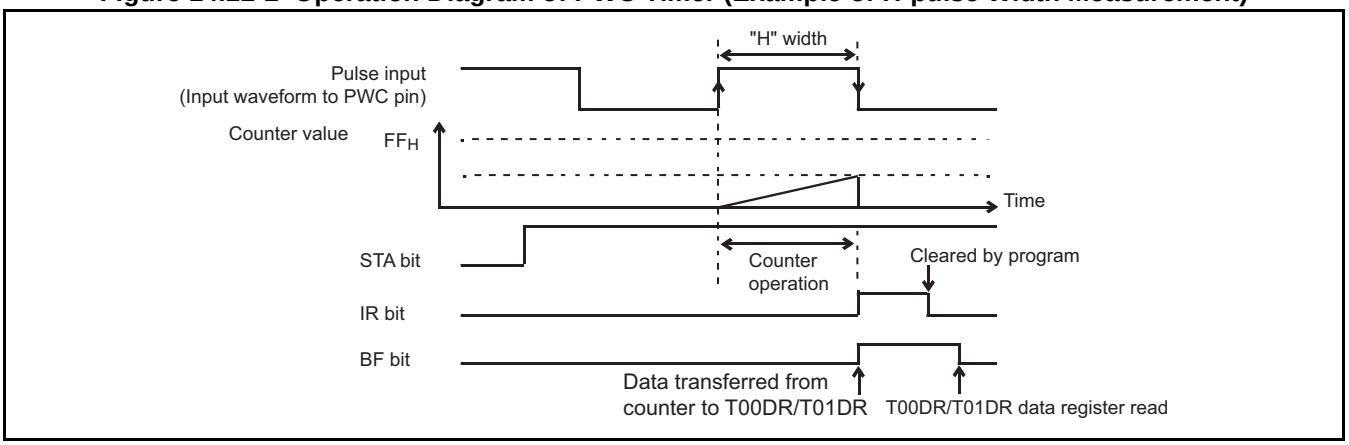
To measure the time exceeding the range of the counter, software can be used to count the number of counter overflows. When the counter overflows, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of overflows. In addition, the timer output is inverted due to the overflow. The timer output initial value can be set by the timer output initial

value bit (T00CR1/T01CR1:SO).

When the timer stops operating, the timer output bit (TMCR0:TO1/TO0) holds the last value.

If an interrupt occurs before the timer is activated (before "1" is written to the STA bit), invalidate the value of the 8/16-bit composite timer 00/01 data register (T00DR/T01DR)

Figure 14.12-2 Operation Diagram of PWC Timer (Example of H-pulse Width Measurement)



14.13 Operation of Input Capture Function

This section describes the operation of the input capture function of the 8/16-bit composite timer.

■ Operation of Input Capture Function

The settings shown in Figure 14.13-1 are required to use the input capture function.

Figure 14.13-1 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							

❍: Bit to be used
×: Unused bit
1: Set to "1"

When the input capture function is selected, the counter value is stored to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) immediately after an edge of the external signal input is detected. The target edge to be detected is selected by the timer operating mode select bits (T00CR0/T01CR0:F3, F2, F1, F0).

This function is available in free-run mode and clear mode, which can be selected by the timer operating mode select bits.

In clear mode, the counter starts counting from "00_H". When an edge is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), the interrupt flag (T00CR1/T01CR1:IR) is set to "1", and the counter returns to "00_H" and restarts counting.

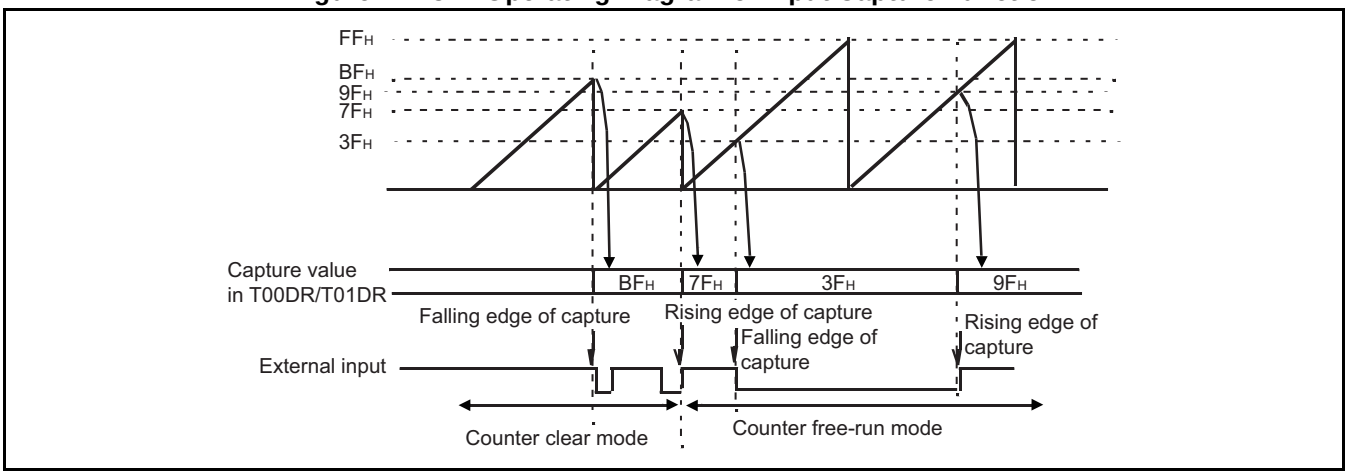
In free-run mode, when an edge is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) is set to "1". In this case, the counter continues to count without being cleared.

This function has no effect on the buffer full flag (T00CR1/T01CR1:BF).

To measure the time exceeding the range of the counter, software can be used to count the number of counter overflows. When the counter overflows, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of overflows. In addition, the timer output is inverted due to the overflow. The timer output initial value can be set by the timer output initial value bit (T00CR1/T01CR1:SO).

If an interrupt occurs before the timer is activated (before "1" is written to the STA bit), invalidate the value captured in the 8/16-bit composite timer 00/01 data register (T00DR/T01DR)

Figure 14.13-2 Operating Diagram of Input Capture Function

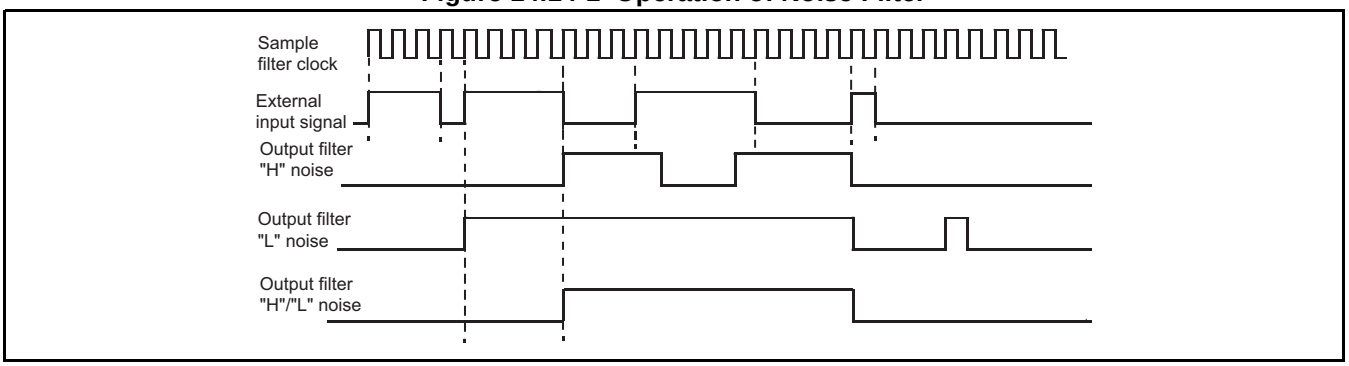


14.14 Operation of Noise Filter

This section describes the operation of the noise filter of the 8/16-bit composite timer.

When the input capture function or PWC timer function is 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 by setting the FE11, FE10, FE01 and FE00 bits in the TMCR0 register. The maximum pulse width that can be eliminated is three machine clock cycles. If the noise filter function is activated, the signal input will be delayed for four machine clock cycles.

Figure 14.14-1 Operation of Noise Filter



14.15 States in Each Mode during Operation

This section describes how the 8/16-bit composite timer behaves when the microcontroller transits to watch mode or stop mode or when a suspend (T00CR1/T01CR1:HO = 1) request is made during operation.

■ When Interval Timer, Input Capture, or PWC Function Is Selected

Figure 14.15-1 shows how the counter value changes when the microcontroller transits to watch mode or stop mode, or a suspend request is made during the operation of the 8/16-bit composite timer.

The counter stops operating while holding the value when the microcontroller transits to stop mode or watch mode. When the stop mode or watch mode is released by an interrupt, the counter resumes operating with the last value that it holds. Therefore, the first interval time or the initial external clock count value is incorrect. Always initialize the counter value after the microcontroller is released from stop mode or watch mode.

Figure 14.15-1 Operations of Counter in Standby Mode or in Pause (Not Serving as PWM Timer)

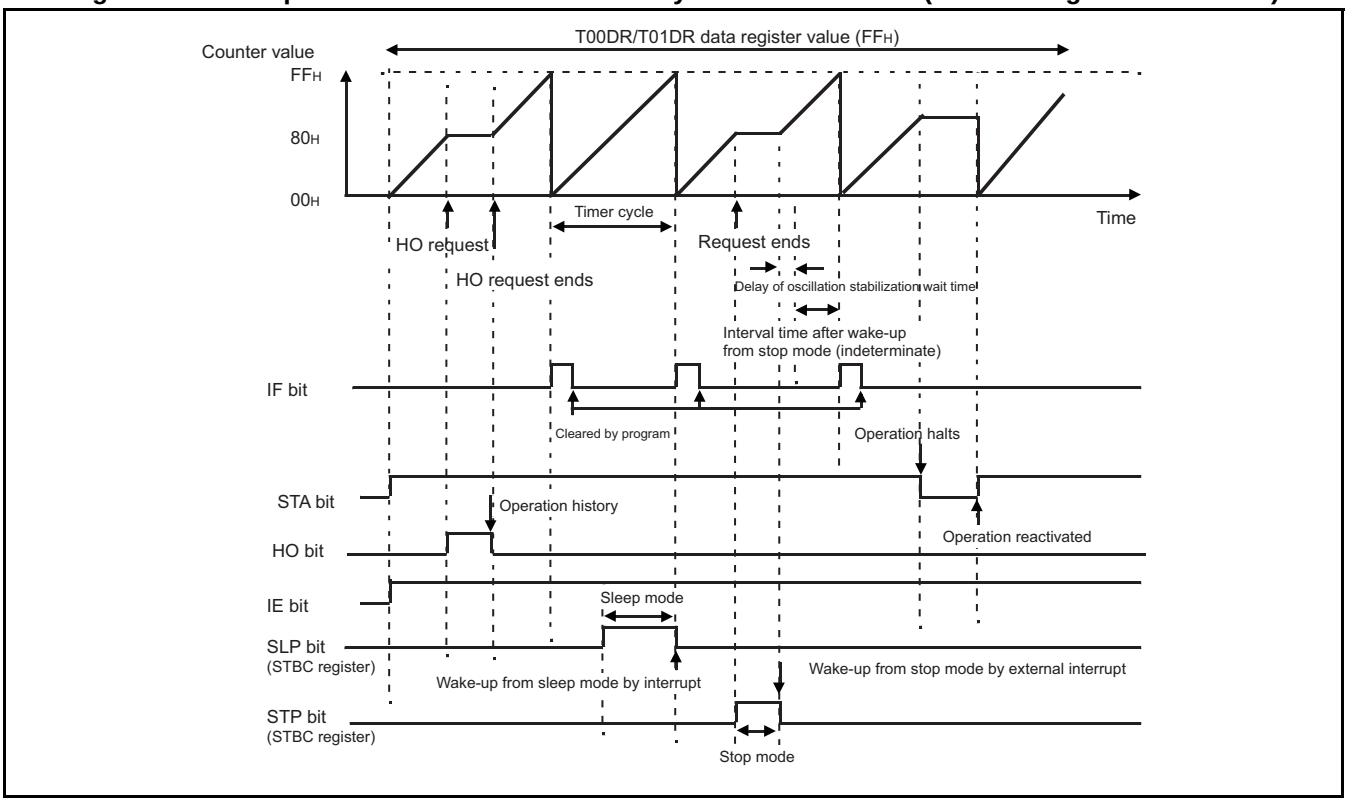
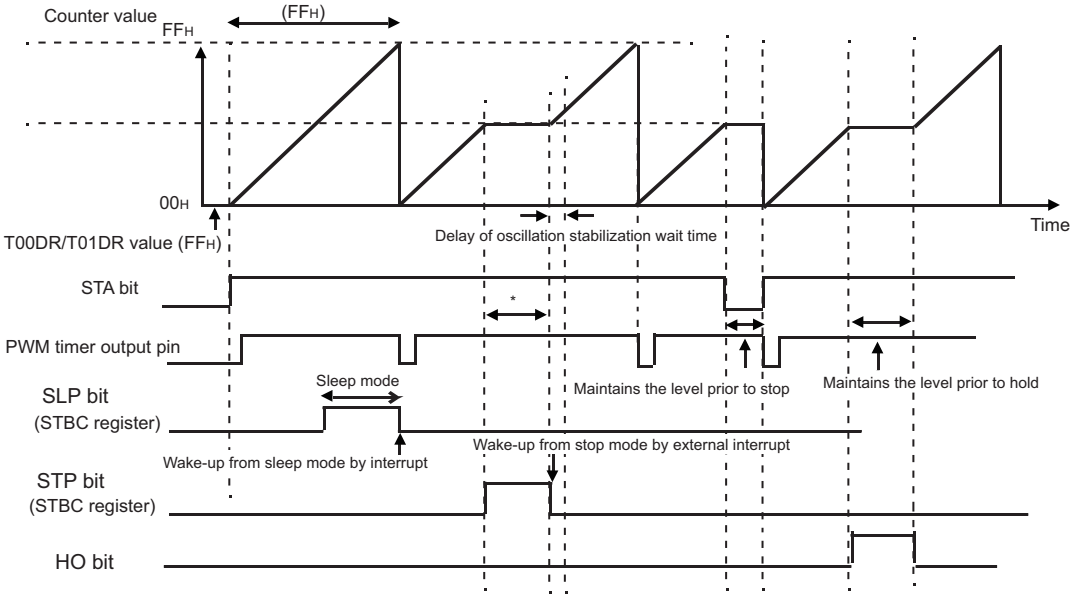


Figure 14.15-2 Operations of Counter in Standby Mode or in Pause (Serving as PWM Timer)



*: The PWM timer output maintains the value held before it enters the stop mode.

14.16 Notes on Using 8/16-bit Composite Timer

This section provides notes on using the 8/16-bit composite timer.

■ Notes on Using 8/16-bit Composite Timer

To switch the timer function with the timer operating mode select bits (T00CR0/T01CR0:F3, F2, F1, F0), stop the timer operation first (T00CR1/T01CR1:STA = 0), then clear the interrupt flag (T00CR1/T01CR1:IF, IR), the interrupt enable bits (T00CR1/T01CR1:IE, T00CR0/T01CR0:IFE) and the buffer full flag (T00CR1/T01CR1:BF).

In the operation of the PWC function or the input capture function, an interrupt may occur even before the timer is started (STA = 0). Therefore, invalidate the value of the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) existing before the start of the timer.

CHAPTER 15

EXTERNAL INTERRUPT CIRCUIT

This chapter describes the functions and operations of the external interrupt circuit.

15.1 Overview of External Interrupt Circuit

The external interrupt circuit detects edges on the signal that is input to the external interrupt pin, and outputs interrupt requests to the interrupt controller.

■ Function of External Interrupt Circuit

The function of the external interrupt circuit is to detect any edge of a signal that is input to an external interrupt pin and to generate an interrupt request to the interrupt controller. The interrupt generated according to this interrupt request can cause the device to wake up from standby mode and return to its normal operating state. Therefore, the operating mode of the device can be changed when a signal is input to the external interrupt pin.

15.2 Configuration of External Interrupt Circuit

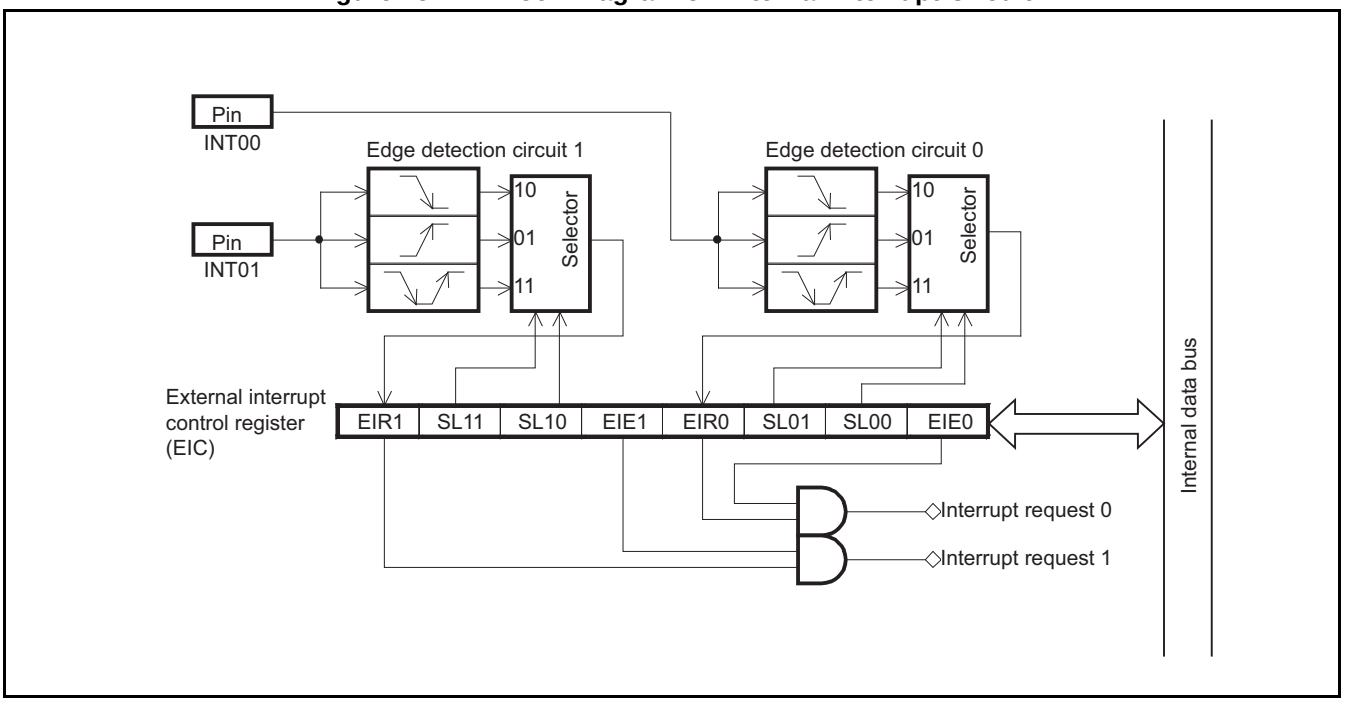
The external interrupt circuit consists of the following blocks:

- **Edge detection circuit**
- **External interrupt control register**

■ Block Diagram of External Interrupt Circuit

Figure 15.2-1 is the block diagram of the external interrupt circuit.

Figure 15.2-1 Block Diagram of External Interrupt Circuit



● Edge detection circuit

When the polarity of the edge detected on a signal input to an external interrupt circuit pin (INT) matches the polarity of the edge selected in the interrupt control register (EIC), a corresponding external interrupt request flag bit (EIR) is set to "1".

● External interrupt control register (EIC)

This register is used to select an edge, enable or disable interrupt requests, check for interrupt requests, etc.

15.3 Channels of External Interrupt Circuit

This section describes the channels of the external interrupt circuit.

■ Channels of External Interrupt Circuit

The MB95200H Series has three units of external interrupt circuit.

Table 15.3-1 shows the pins of the external interrupt circuit and Table 15.3-2 shows its registers.

Table 15.3-1 Pins of External Interrupt Circuit

Unit	Pin name	Pin function
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 15.3-2 Registers of External Interrupt Circuit

Unit	Register name	Corresponding register (Name in this manual)
1	EIC10	EIC: External Interrupt Control register
2	EIC20	
3	EIC30	

The MB95210H Series has two units of external interrupt circuit.

Table 15.3-3 shows the pins of the external interrupt circuit and Table 15.3-4 shows its registers.

Table 15.3-3 Pins of External Interrupt Circuit

Unit	Pin name	Pin function
2	INT04	External interrupt input ch.4
3	INT06	External interrupt input ch.6

Table 15.3-4 Registers of External Interrupt Circuit

Unit	Register name	Corresponding register (Name in this manual)
2	EIC20	EIC: External Interrupt Control register
3	EIC30	

In the following sections, only details of unit 1of the external interrupt circuit are provided.

Details of other units of the external interrupt circuit are the same as those of unit 1.

15.4 Pins of External Interrupt Circuit

This section provides details of the pins related to the external interrupt circuit and the block diagrams of such pins.

■ Pins Related to External Interrupt Circuit

In the MB95200H Series, the pins related to the external interrupt circuit are the INT02 to INT07 pins.

● INT02 to INT07 pins

These pins serve both as external interrupt input pins and as general-purpose I/O ports.

INT02 to INT07: If a pin of INT02 to INT07 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 (INT02 to INT07).

The state of a pin can always be read from the port data register (PDR) when that pin is set as an input port. However, the value of PDR is read when the read-modify-write (RMW) type of instruction is used.

■ Pins Related to External Interrupt Circuit

In the MB95210 Series, the pins related to the external interrupt circuit are the INT04 and INT06 pins.

● INT04 and INT06 pins

These pins serve both as external interrupt inputs and as general-purpose I/O ports.

INT04 and INT06: If INT04 or INT06 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 (INT04 or INT06).

The state of a pin can always be read from the port data register (PDR) when that pin is set as an input port. However, the value of PDR is read when the read-modify-write (RMW) type of instruction is used.

■ Block Diagrams of Pins Related to External Interrupt Circuit

Figure 15.4-1 Block Diagram of Pins INT02,INT03,INT05 (P02/INT02/AN02/SCK, P03/INT03/AN03/SOT, P05/INT05/AN05/TO00/HCLK2) Related to External Interrupt Circuit

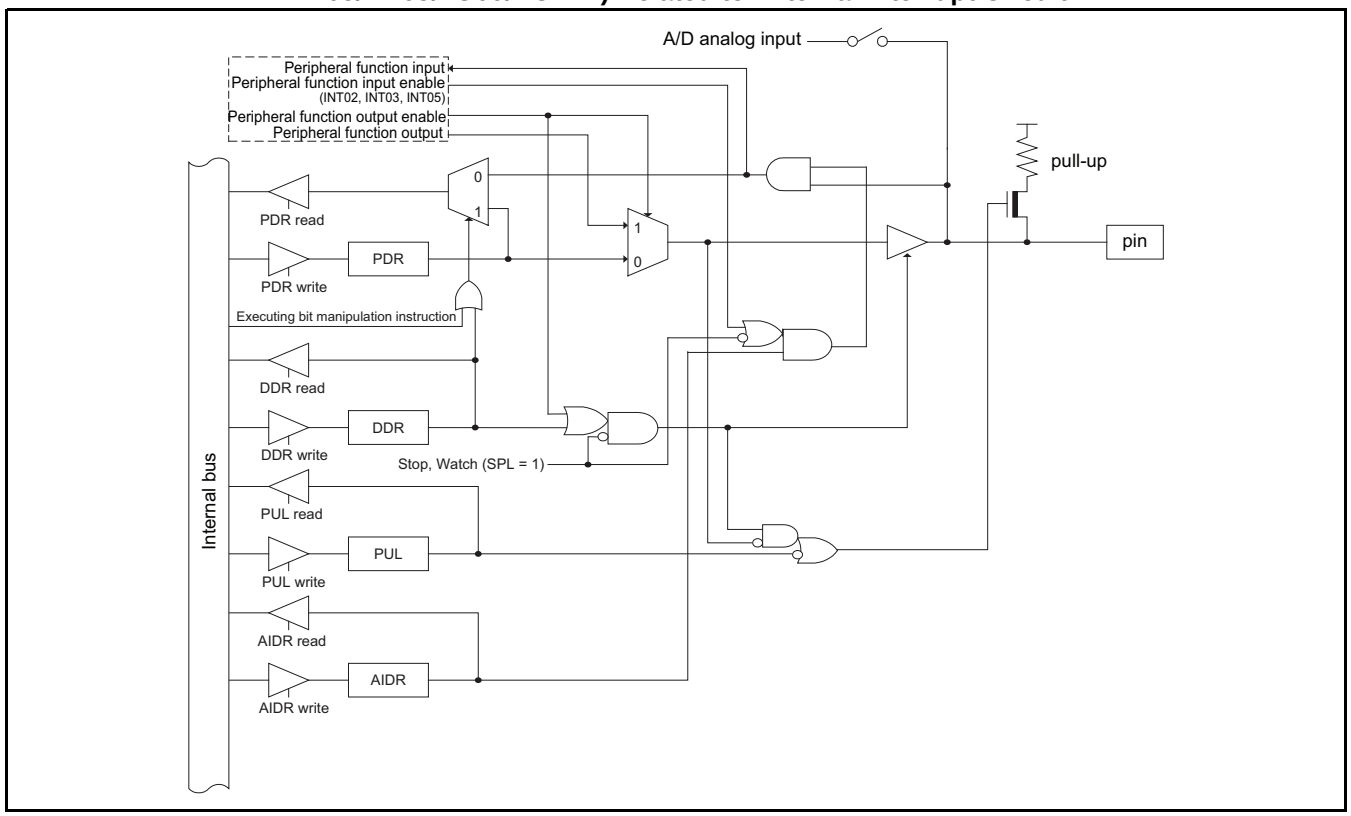


Figure 15.4-2 Block Diagram of Pin INT04 (P04/INT04/AN04/SIN /HCLK1/EC0) Related to External Interrupt Circuit

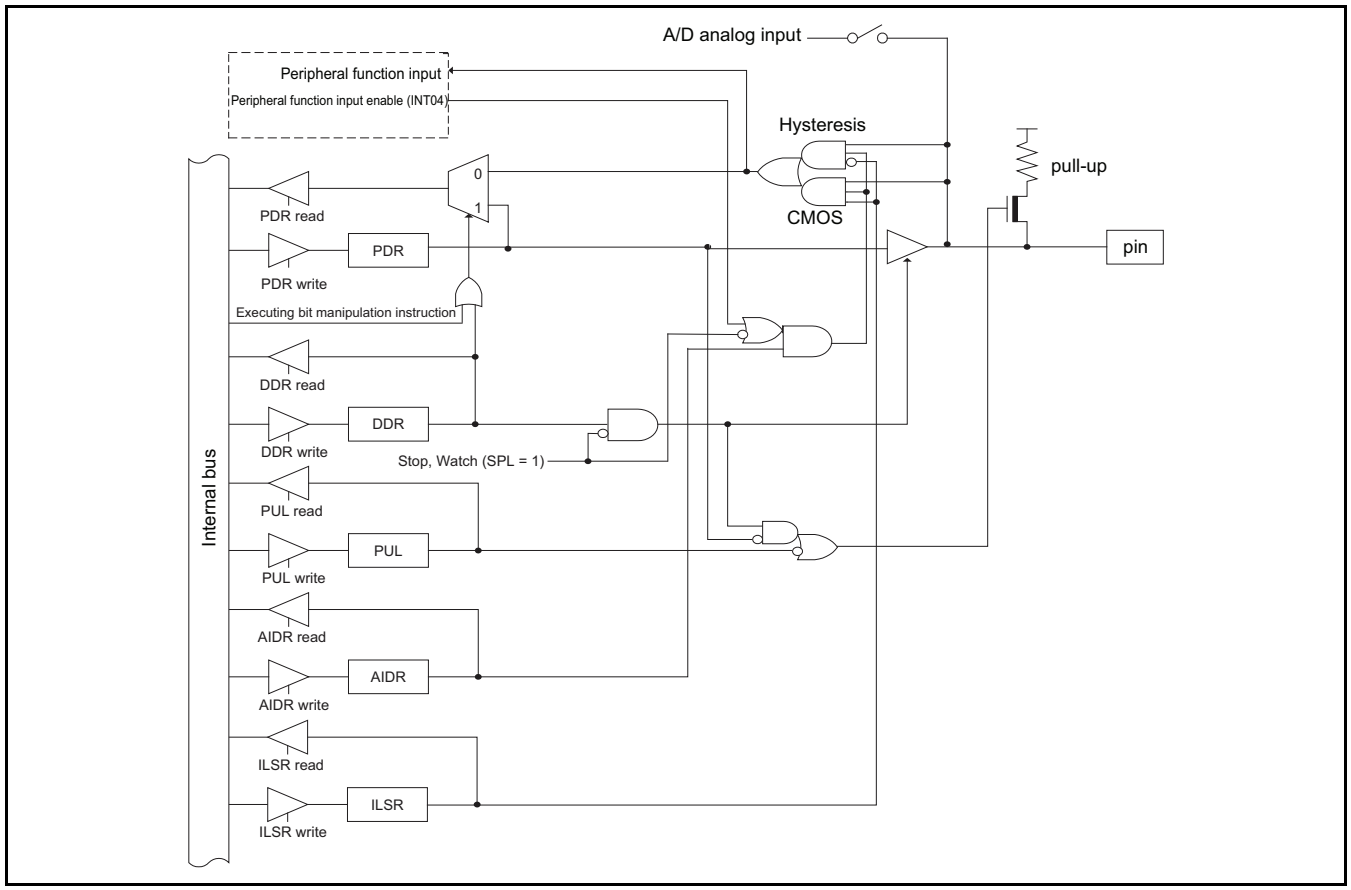


Figure 15.4-3 Block Diagram of Pin INT06 (P06/INT06/TO01) Related to External Interrupt Circuit

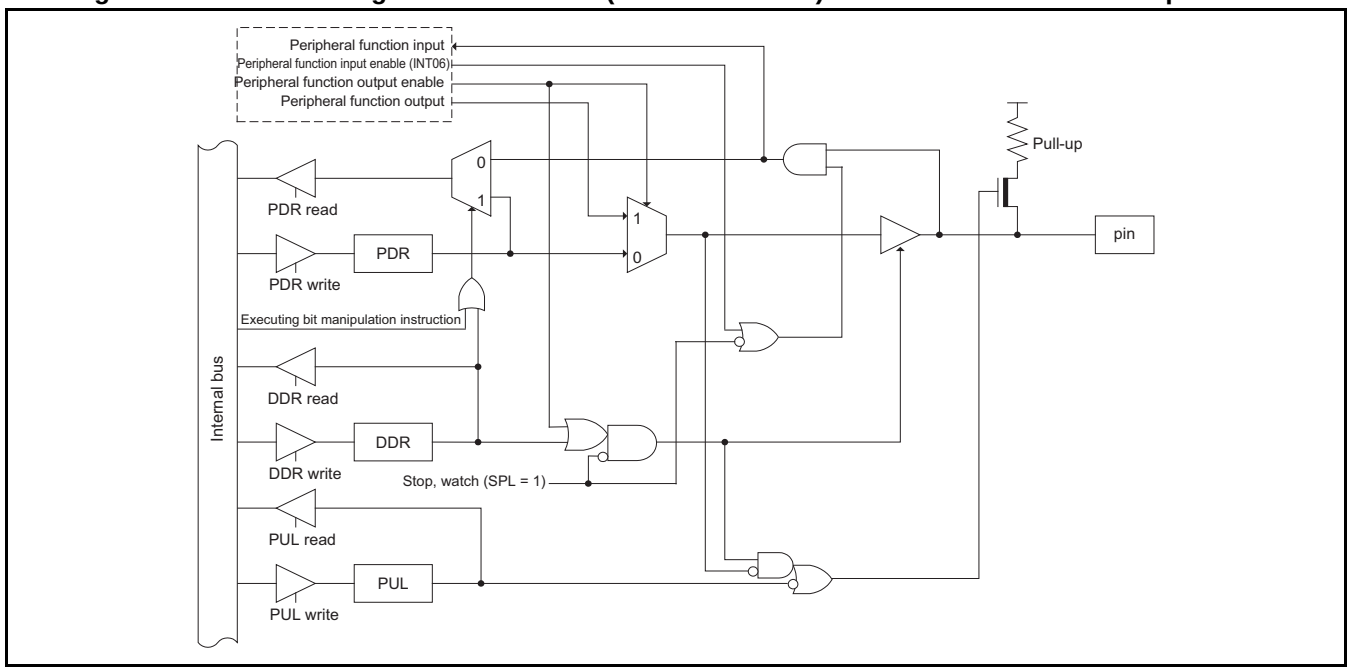
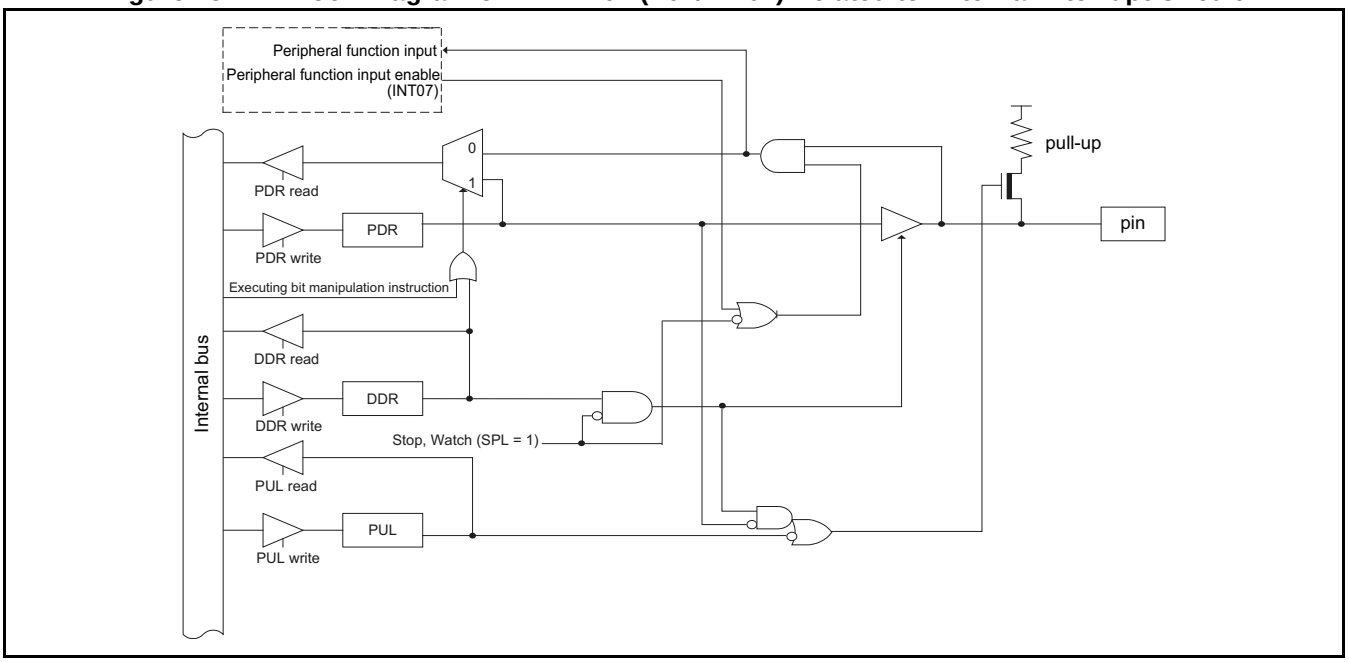


Figure 15.4-4 Block Diagram of Pin INT07 (P07/INT07) Related to External Interrupt Circuit



15.5 Registers of External Interrupt Circuit

This section describes the registers of the external interrupt circuit.

■ Registers of External Interrupt Circuit

Figure 15.5-1 shows the registers of the external interrupt circuit.

Figure 15.5-1 Registers of External Interrupt Circuit

External interrupt control register (EIC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0049_H EIC10	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
$004A_H$ EIC20	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
$004B_H$ EIC30	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 (The read value is the same as the write value.)
R(RM1), W: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)

15.5.1 External Interrupt Control Register (EIC10)

The external interrupt control register (EIC10) is used to select the edge polarity for the external interrupt input and control interrupts.

■ External Interrupt Control Register (EIC10)

Figure 15.5-2 External Interrupt Control Register (EIC10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0049H EIC10 004AH EIC20 004BH EIC30	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	

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 input	Clears this bit
1	Specified edge input	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 input	Clears this bit
1	Specified edge input	No change, no effect on others

R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
□ : Initial value

Table 15.5-1 Functions of Bits in External Interrupt Control Register (EIC10)

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 input to the external interrupt pin INT03. • When this bit and the interrupt request enable bit 1 (EIE1) are set to "1", an interrupt request is output. • Writing "0" clears the bit. Writing "1" has no effect on operation. • If this bit is read by the read-modify-write (RMW) type of instruction, it returns "1".
bit6, bit5	SL11, SL10: Edge polarity select bits 1	These bits select the polarity of an edge of the pulse input to the external interrupt pin INT03. The edge selected is to be the interrupt source. • If these bits are set to "00_B", edge detection is not performed and no interrupt request is made. • If these bits are set to "01_B", rising edges are to be detected; if "10_B", falling edges are to be detected; if "11_B", both edges are to be detected.
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 output. • 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 port. • 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 input to the external interrupt pin INT02. • When this bit and the interrupt request enable bit 0 (EIE0) are set to "1", an interrupt request is output. • Writing "0" clears the bit. Writing "1" has no effect on operation. • If this bit is read by the read-modify-write (RMW) type of instruction, it returns "1".
bit2, bit1	SL01, SL00: Edge polarity select bits 0	These bits select the polarity of an edge of the pulse input to the external interrupt pin INT02. The edge selected is to be the interrupt source. • If these bits are set to "00_B", edge detection is not performed and no interrupt request is made. • If these bits are set to "01_B", rising edges are to be detected; if "10_B", falling edges are to be detected; if "11_B", both edges are to be detected.
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 output 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 port. • 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.

15.6 Interrupts of External Interrupt Circuit

The interrupt sources for the external interrupt circuit include detection of the specified edge of the signal input 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, if the interrupt request enable bit (EIC: EIE0, EIE1 = 1) corresponding to that external interrupt request flag bit is enabled, an interrupt request is generated to the interrupt controller. In an interrupt service routine, write "0" to the external interrupt request flag bit corresponding to that interrupt request generated to clear the interrupt request.

■ Registers and Vector Table Addresses Related to Interrupts of External Interrupt Circuit

Table 15.6-1 MB95200H Registers and Vector Table Addresses 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.4	IRQ0	ILR0	L00	$FFFA_H$	$FFFB_H$
ch.5	IRQ1	ILR0	L01	$FFF8_H$	$FFF9_H$
ch.2	IRQ2	ILR0	L02	$FFF6_H$	$FFF7_H$
ch.6					
ch.3	IRQ3	ILR0	L03	$FFF4_H$	$FFF5_H$
ch.7					

Table 15.6-2 MB95210H Registers and Vector Table Addresses 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.4	IRQ0	ILR0	L00	$FFFA_H$	$FFFB_H$
ch.6	IRQ2	ILR0	L02	$FFF6_H$	$FFF7_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the interrupt request numbers and vector tables of all peripheral functions.

15.7 Operations of External Interrupt Circuit and Setting Procedure Example

This section describes the operations of the external interrupt circuit.

■ Operations of External Interrupt Circuit

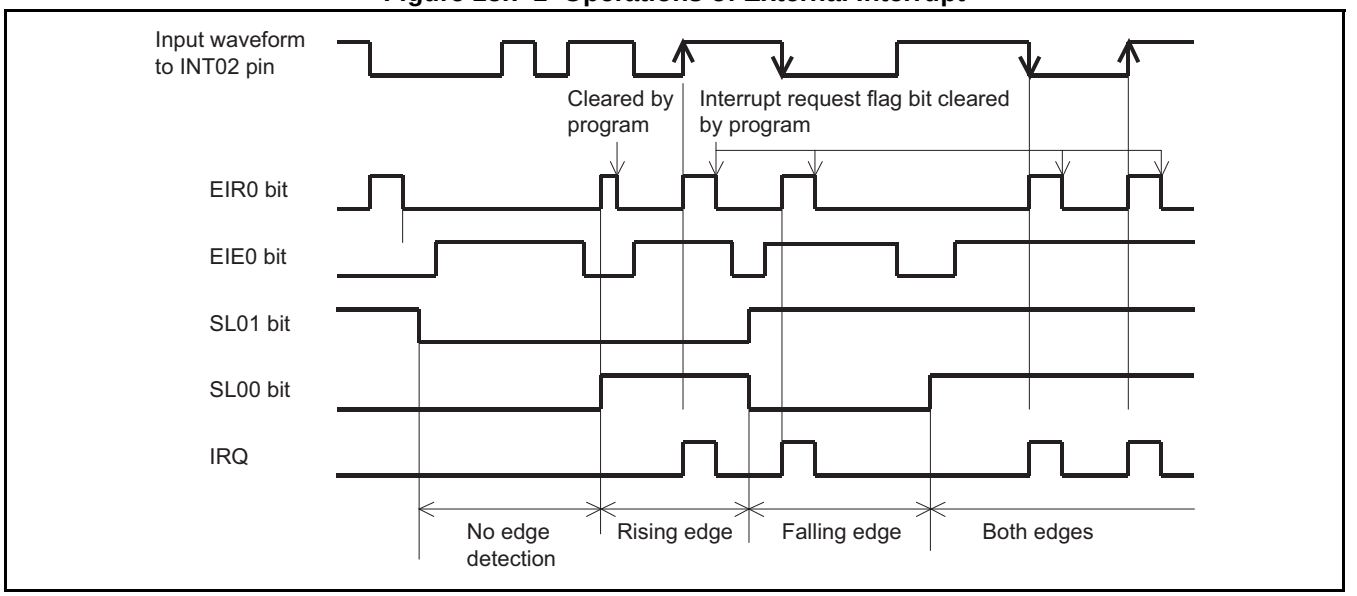
When the polarity of an edge of a signal input from one of the external interrupt pins (INT02, INT03) matches the polarity of the edge selected by the external interrupt control register (EIC: SL00 to SL11), the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1" and the interrupt request is generated.

Always set the interrupt request enable bit to "0" when not using an external interrupt to wake up the device from 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 15.7-1 shows the operations for setting the INT02 pin as an external interrupt input.

Figure 15.7-1 Operations of External Interrupt



■ Setting Procedure Example

Below is an example of procedure for setting the external interrupt circuit.

● Initial settings

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:

An external interrupt input port shares the same pin with an I/O port. Therefore, when using the pin as an external interrupt input port, set the bit in the port direction register (DDR) corresponding to that pin to "0" (input).

15.8 Notes on Using External Interrupt Circuit

This section provides notes on using the external interrupt circuit.

■ Notes on Using External Interrupt Circuit

- Prior to setting the edge polarity select bit (SL), set the interrupt request enable bit (EIE) to "0" (disabling interrupt requests) when setting the edge polarity select bit (SL). In addition, clear the external interrupt request flag bit (EIR) to "0" after setting the edge polarity.
- The external interrupt circuit cannot wake up from the interrupt service routine if the external interrupt request flag bit is "1" and the interrupt request enable bit is enabled. In the interrupt service routine, always clear the external interrupt request flag bit.

15.9 Sample Programs for External Interrupt Circuit

Fujitsu Microelectronics provides 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, see "■ Sample Programs" in "Preface".

■ Setting Methods Other than Sample Programs

● Detection levels and setting 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.

Operating mode	Detection level bits (SL01,SL00)
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 a corresponding bit in the data direction register (DDR0) to "0".

Operation	Direction bit (P02 to P07)	Setting
Using INT02 pin for external interrupt	DDR0: P02	Set to "0"
Using INT03 pin for external interrupt	DDR0: P03	Set to "0"
Using INT04 pin for external interrupt	DDR0: P04	Set to "0"
Using INT05 pin for external interrupt	DDR0: P05	Set to "0"
Using INT06 pin for external interrupt	DDR0: P06	Set to "0"
Using INT07 pin for external interrupt	DDR0: P07	Set to "0"

● Interrupt-related registers

The interrupt level is set by the interrupt level setting registers shown in the following table.

Channel	Interrupt level setting register	Interrupt vector
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 interrupt requests

Interrupts requests are enabled/disabled by the interrupt request enable bit (EIC00: EIE0 or EIE1).

Operation	Interrupt request enable bit (EIE0 or EIE1)
To disable an interrupt requests	Set to "0"
To enable an interrupt request	Set to "1"

Interrupt requests are cleared by the interrupt request bit (EIC00: EIR0 or EIR1).

Operation	Interrupt request bit (EIR0 or EIR1)
To clear an interrupt request	Set to "0"

CHAPTER 16

LIN-UART

This chapter describes the functions and operations of the LIN-UART (The LIN-UART is available in the MB95200H Series).

16.1 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 special functions with the LIN bus.

■ Functions of LIN-UART

The LIN-UART is a general-purpose serial data communication interface for exchanging serial data with other CPUs and peripheral devices. Table 16.1-1 lists the functions of the LIN-UART.

Table 16.1-1 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 of sampling values (only asynchronous mode).
Transfer mode	• Clock-synchronous (Select start/stop synchronization, or start/stop bits) • 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. It can be adjusted by the reload counter.
Data length	• 7 bits (not in synchronous or LIN mode) • 8 bits
Signal type	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 operating mode 1)
Interrupt request	• Receive interrupts (reception completed, reception error detected, LIN synch break detected) • Transmit interrupts (transmit 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 (supports both the master and slave system)
Synchronous mode	Transmit 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 composite timer
Synchronous serial clock	Continuous output to the SCK pin enabled for synchronous communication using the start/stop bits
Clock delay option	Special synchronous clock mode for delaying the clock (used in Special Peripheral Interface (SPI))

The LIN-UART operates in four different modes. The operating mode is selected by the MD0 and MD1 bits in the LIN-UART serial mode register (SMR). Operating mode 0 and operating mode 2 are used for bi-directional serial communication; mode 1 for master/slave communication; and mode 3 for LIN master/slave communication.

Table 16.1-2 LIN-UART Operating Modes

Operating mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multiprocessor mode	7 bits or 8 bits +1*	-	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	
3	LIN mode	8 bits	-	Asynchronous	1 bit	LSB first

- : Unavailable
* : "+1" is the address/data select 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 operating modes.

Table 16.1-3 LIN-UART Operating 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.
- The communication format of Mode 3 is fixed: 8-bit data, no parity, stop bit 1, LSB-first.

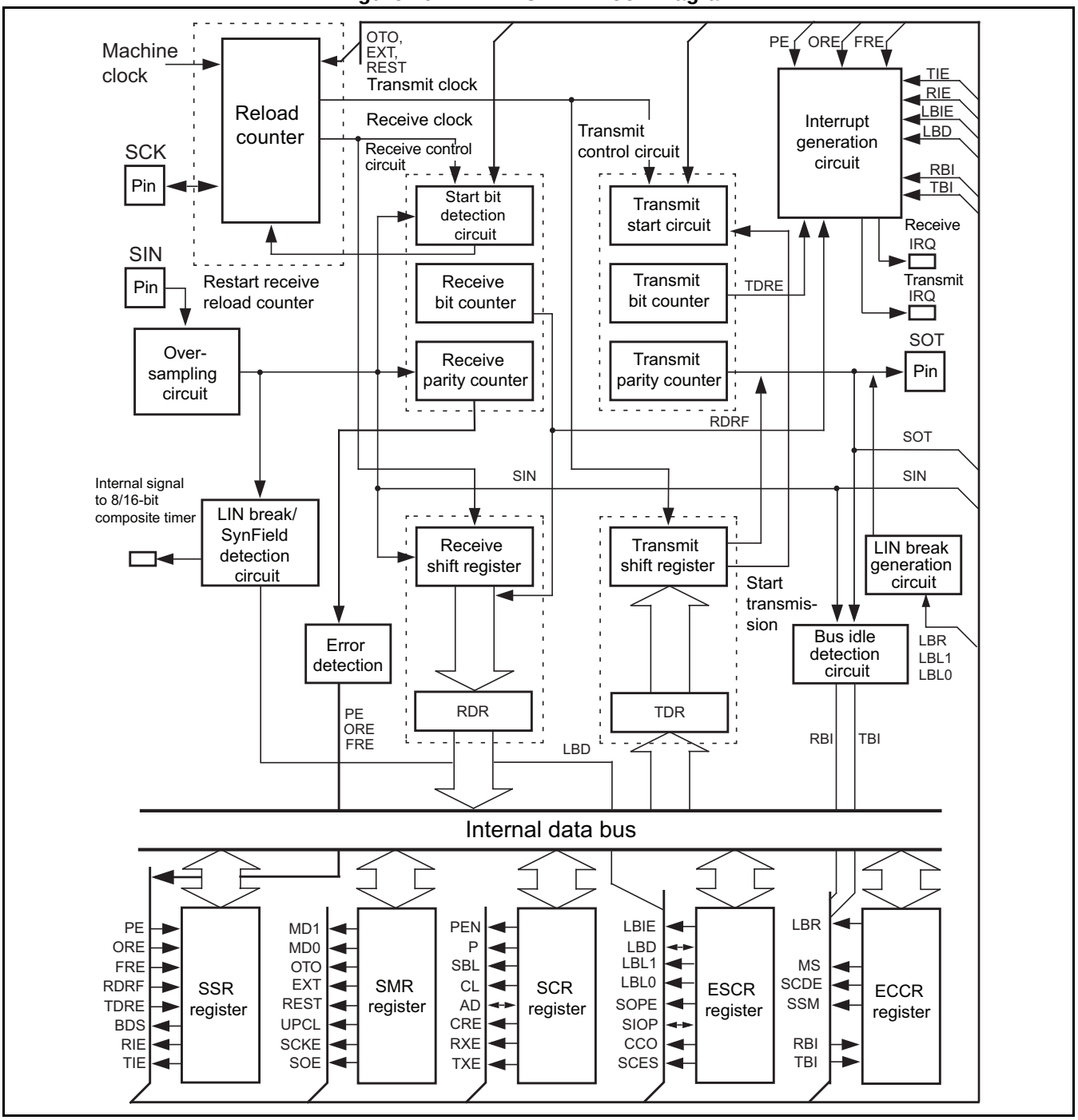
16.2 Configuration of LIN-UART

LIN-UART is made up of the following blocks.

- **Reload counter**
- **Receive control circuit**
- **Receive shift register**
- **LIN-UART receive 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 and data register (SSR)**
- **LIN-UART extended status control register (ESCR)**
- **LIN-UART extended communication control register (ECCR)**

■ LIN-UART Block Diagram

Figure 16.2-1 LIN-UART Block Diagram



● Reload counter

This block is a 15-bit reload counter functioning as a dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/receive clock from the external or internal clock. The count value in the transmit reload counter is read from the baud rate generator1, 0 (BGR 1 and BGR 0).

- Receive control circuit

  This block consists of a receive bit counter, a start bit detection circuit, and a receive parity counter. The receive bit counter counts the receive data bits and sets a flag in the LIN-UART receive data register when the reception of one data is completed according to the specified data length. If the receive interrupt has been enabled, a receive interrupt request is made. 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 receive parity counter calculates the parity of the received data.

- Receive shift register

  The circuit captures received data from the SIN pin while performing bit shifting of received data. The receive shift register transfers received data to the RDR register.

- LIN-UART receive data register (RDR)

  This register retains the received data. Serial input data is converted and stored in the LIN-UART receive data register.

- Transmit control circuit

  This block consists of a transmit bit counter, a transmit 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 the transmission of one data is completed according to the specified data length. If the transmit interrupt has been enabled, a transmit interrupt request is made. 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 has a parity.

- Transmit shift register

  Data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register, and then the transmit shift register outputs the data to the SOT pin while performing bit shifting of the data.

- LIN-UART transmit data register (TDR)

  This register sets the transmit data. Data written to this register is converted to serial data and then output.

- Error detection circuit

  This circuit detects errors occurring at the end of reception. If an error occurs, a corresponding error flag is set.

- Oversampling circuit

  In asynchronous mode, the oversampling circuit oversamples received data for five times to determine the received value by majority of sampling values. The circuit stops operating in synchronous mode.

- Interrupt generation circuit

  This circuit controls all interrupt sources. An interrupt is generated immediately provided that 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 composite timer in order to detect the first and the 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 a length set.

- Bus idle detection circuit

  If this circuit detects that no transmission or reception is in progress, it sets the TBI flag bit or the RBI flag bit to "1" respectively.

- LIN-UART serial control register (SCR)

  Its operating functions are as follows:

  - Setting the use of the parity bit
  - Parity bit select
  - Setting stop bit length
  - Setting data length
  - Selecting the frame data format in mode 1
  - Clearing the error flag
  - Enabling/disabling transmission
  - Enabling/disabling reception

- LIN-UART serial mode register (SMR)

  Its operating functions are as follows:

  - Selecting the LIN-UART operating mode
  - Selecting the clock input source
  - Selecting between one-to-one connection to the external clock and connection to the reload counter
  - Resetting the dedicated reload timer
  - LIN-UART software reset (maintaining register settings)
  - Enabling/disabling output to the serial data pin
  - Enabling/disabling output to the clock pin

- LIN-UART serial status register (SSR)

  Its operating functions are as follows:

  - Checking transmission/reception or error status
  - Selecting the transfer direction (LSB-first or MSB-first)
  - Enabling/disabling receive interrupts
  - Enabling/disabling transmit interrupts

- Extended status control register (ESCR)

  Its operating functions are as follows:

  - Enabling/disabling LIN synch break interrupts
  - LIN synch break detection
  - Selecting LIN synch break length
  - Direct access to SIN pin and SOT pin
  - Setting continuous clock output in LIN-UART synchronous clock mode
  - Sampling clock edge selection

- LIN-UART extended communication control register (ECCR)

  Its operating functions are as follows:

  - Bus idle detection
  - Synchronous clock setting
  - LIN synch break generation

■ Input Clock

The LIN-UART uses a machine clock or an input signal from the SCK pin as an input clock.

The input clock is used as the transmission/reception clock source of the LIN-UART.

16.3 LIN-UART Pins

This section describes LIN-UART pins.

■ LIN-UART Pins

The LIN-UART pins are also used as general-purpose ports. Table 16.3-1 lists the LIN-UART pin functions and settings for using them.

Table 16.3-1 LIN-UART Pins

Pin name	Pin function	Settings required for using pin
SIN	Serial data input	Set to the input port (DDR: corresponding bit = 0)
SOT	Serial data output	Enable output. (SMR:SOE = 1)
SCK	Serial clock input/output	Set to the input port when this pin is used for clock input. (DDR: corresponding bit = 0)
		Enable output when this pin is used as an clock output pin. (SMR:SCKE = 1)

■ Block Diagrams of LIN-UART Pins

Figure 16.3-1 Block Diagram of Pins SCK, SOT(P02/INT02/AN02/SCK, P03/INT03/AN03/SOT) Related to LIN-UART

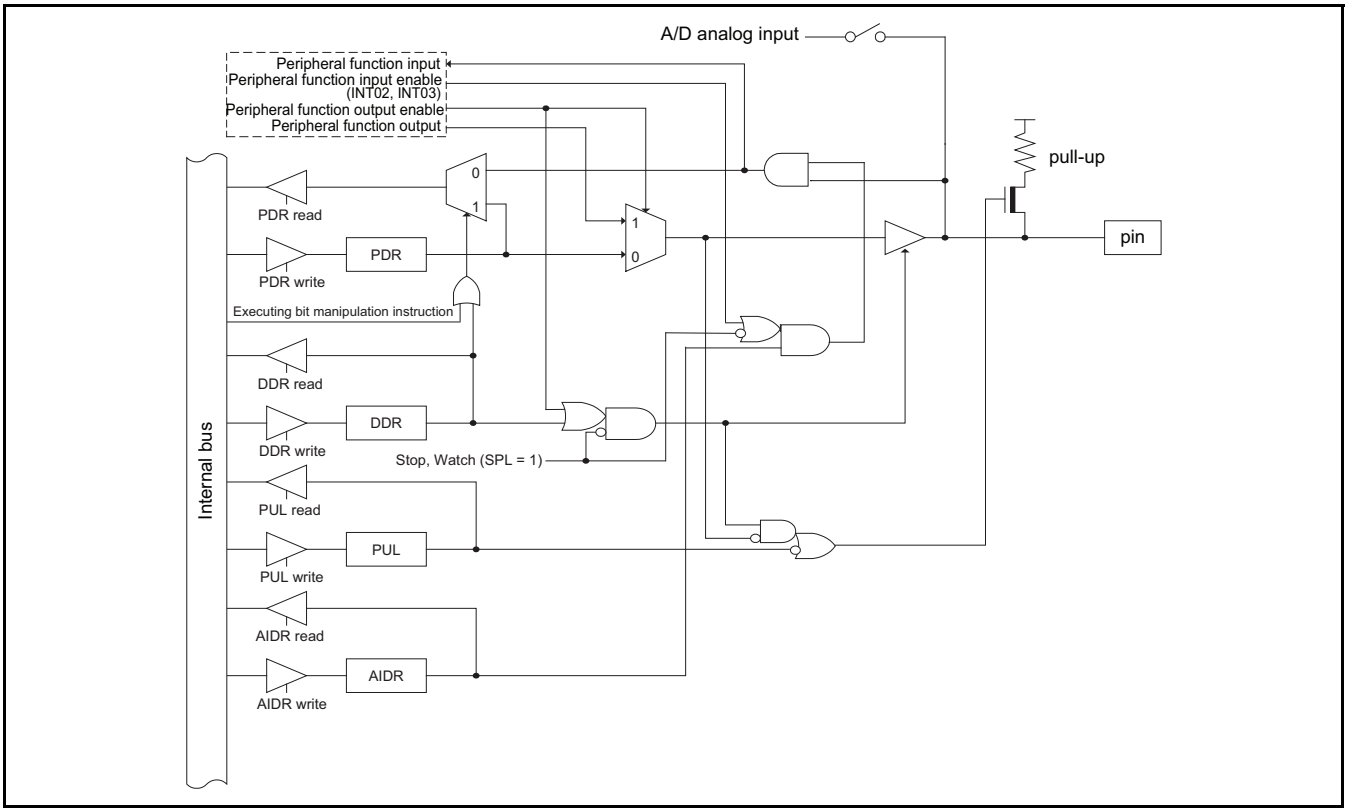
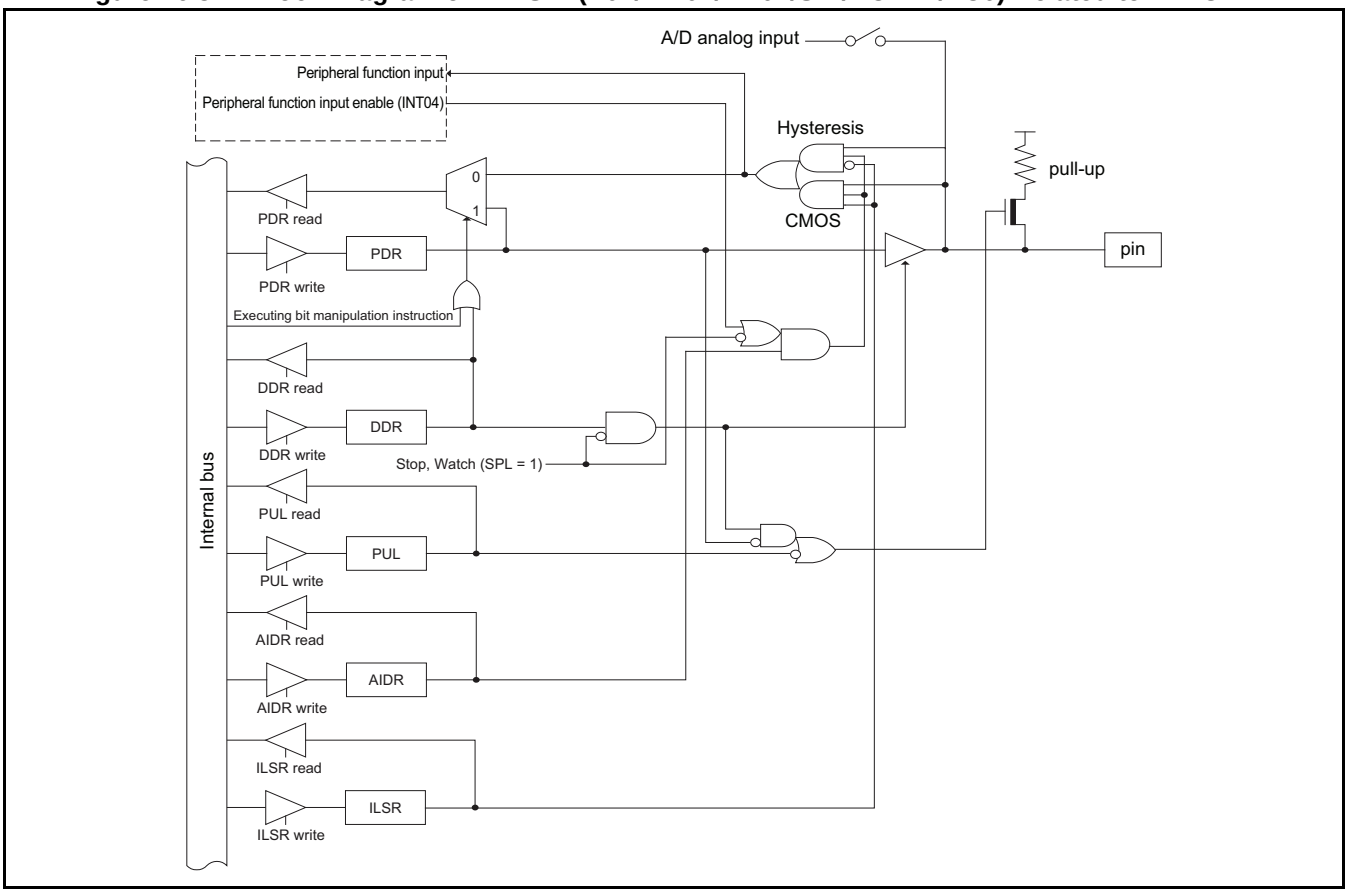


Figure 16.3-2 Block Diagram of Pin SIN (P04/INT04/AN04/SIN/HCLK1/EC0) Related to LIN-UART



16.4 Registers of LIN-UART

This section lists the registers of the LIN-UART.

■ Registers of LIN-UART

Figure 16.4-1 Registers 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 receive data register/transmit data register (RDR/TDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053_H									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	$000000XX_B$
	R0/W	R0,W	R/W	R/W	R/W	RX,W0	R/WX	R/WX	

LIN-UART baud rate generator register 1 (BGR 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBC_H$	-								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 (BGR 0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBD_H$									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (The read value is the same as the write value.)
R/WX: Read only (Readable. Writing a value to it has no effect on operation.)
R(RM1), W: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R0, W: Write only (Writable. "0" is read.)
R0, WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
RX,W0: Reserved bit (The read value is undefined; the write value is "0".)

16.4.1 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 receive error flag, and enable/disable transmission/reception.

■ LIN-UART Serial Control Register (SCR)

Figure 16.4-2 LIN-UART Serial Control Register (SCR)

Address 0050H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PEN	P	SBL	CL	AD	CRE	RXE	TXE
R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W

Initial value: 00000000B

TXE	Transmit operation enable bit
0	Disable transmission
1	Enable transmission

RXE	Receive operation enable bit
0	Disable reception
1	Enable reception

CRE	Receive error flag clear bit	
	Write	Read
0	No effect	"0" is always read.
1	Clear receive error flag (PE, FRE, ORE)	

AD	Address/data format select bit
0	Data frame
1	Address frame

CL	Data length select bit
0	7-bit
1	8-bit

SBL	Stop bit length select bit
0	1-bit
1	2-bit

P	Parity select bit
0	Even parity
1	Odd parity

PEN	Parity enable bit
0	No parity
1	With parity

R/W : Readable/writable
(The read value is the same as the write value.)
R0,W : Write only (Writable. "0" is read.)
: Initial value

Table 16.4-1 Functions of Bits in LIN-UART Serial Control Register (SCR)

Bit name		Function
bit7	PEN: Parity enable bit	This bit specifies whether or not to add (at transmission) and detect (at reception) a parity bit. Note: The parity bit is added only in operating mode 0, or in operating mode 2 in which the start/stop bits are to be added to the synchronous data format (ECCR:SSM = 1). This bit is fixed at "0" in operating mode 3 (LIN).
bit6	P: Parity select bit	With the parity bit having been enabled (SCR:PEN = 1), setting this bit to "1" selects the odd parity and setting this bit to "0" selects the even parity.
bit5	SBL: Stop bit length select bit	This bits sets the bit length of the stop bit (frame end mark in transmit data) in operating mode 0, 1 (asynchronous) or in operating mode 2 (synchronous) in which the start/stop bits are to be added to the synchronous data format (ECCR:SSM = 1). This bit is fixed at "0" in operating mode 3 (LIN). Note: At reception, only the first bit of the stop bit is always detected.
bit4	CL: Data length select bit	This bit specifies the data length to be transmitted and received. This bit is fixed at "1" in operating mode 2 and operating mode 3.
bit3	AD: Address/data format select bit	This bit specifies the data format for the frame to be transmitted and received in multiprocessor mode (mode 1). Write a value to this bit in master mode; read this bit in slave. The operation in master mode is as follows. **Writing "0":** the data frame is used as the data format. **Writing "1":** the address data frame is used as the data format. The value for the last received data format is read. Note: See "16.8 Notes on Using LIN-UART" for the usage of this bit.
bit2	CRE: Receive error flag clear bit	This bit clears the FRE, ORE, and PE flags in serial status register (SSR). **Writing "0":** has no effect on operation. **Writing "1":** clears the error flag. When this bit is read, it always returns "0". Note: Clear the receive error flag after reception is disabled (RXE=0) If the receive error flag is cleared without reception being disabled, the reception is immediately suspended, but will be resumed after a certain period of time. Therefore, after reception is resumed, the data that the LIN-UART receives may not be correct.
bit1	RXE: Receive operation enable bit	This bits enables or disables the reception of the LIN-UART. **Writing "0":** disables data frame reception. **Writing "1":** enables data frame reception. The LIN synch break detection in operating mode 3 is not affected by the setting of this bit. Note: If data frame reception is disabled (RXE = 0) while it is in progress, the reception halts immediately. In this case, the integrity of data is not guaranteed.
bit0	TXE: Transmit operation enable bit	This bits enables or disables the transmission of the LIN-UART. **Writing "0":** disables data frame transmission. **Writing "1":** enables data frame transmission. Note: When data frame transmission is disabled (TXE = 0) while it is in progress, the transmission halts immediately. In this case, the integrity of data is not guaranteed.

16.4.2 LIN-UART Serial Mode Register (SMR)

The LIN-UART serial mode register (SMR) is used to select the operating mode, specify the baud rate clock, and enable/disable output to the serial data and clock pins.

■ LIN-UART Serial Mode Register (SMR)

Figure 16.4-3 LIN-UART Serial Mode Register (SMR)

Address 0051H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W

Initial value: 00000000B

SOE	LIN-UART serial data output enable bit
0	General-purpose I/O port
1	LIN-UART serial data output pin

SCKE	LIN-UART serial clock output enable bit
0	General-purpose I/O port or LIN-UART clock input pin
1	LIN-UART serial clock output pin

UPCL	LIN-UART programmable clear bit	
	Write	Read
0	No effect	"0" is always read.
1	LIN-UART reset	

REST	Reload counter restart bit	
	Write	Read
0	No effect	"0" is always read.
1	Restart the reload counter	

EXT	External serial clock source select bit
0	Use the baud rate generator (reload counter).
1	Use the external serial clock source.

OTO	One-to-one external clock input enable bit
0	Use the baud rate generator (reload counter).
1	Use the external clock directly.

MD1	MD0	Operating mode select bits
0	0	Mode 0: Asynchronous (Normal mode)
0	1	Mode 1: Asynchronous (Multiprocessor mode)
1	0	Mode 2: Synchronous (Normal mode)
1	1	Mode 3: Asynchronous (LIN mode)

R/W : Readable/writable
(The read value is the same as the write value.)
R0,W : Write only (Writable. "0" is read.)
: Initial value

Table 16.4-2 Functions of Bits 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:
Operating mode select bits</td><td>These bits sets the operating mode.
Note: If the mode is changed during communication, exchanging on the LIN-UART is suspended and the LIN-UART waits for the start of 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>Writing "1": enables the external clock to be used directly as the LIN-UART serial clock.
In operating mode 2 (asynchronous), the external clock is used when the reception side of the serial clock is selected (ECCR:MS = 1).
When EXT = 0, the OTO bit is fixed at "0".</td></tr>
<tr><td>bit4</td><td>EXT:
External serial clock source select bit</td><td>This bit selects a clock input.
Writing "0": selects the clock of the internal baud rate generator (reload counter).
Writing "1": selects the external serial clock source.</td></tr>
<tr><td>bit3</td><td>REST:
Reload counter restart bit</td><td>This bits restarts the reload counter.
Writing "0": no effect.
Writing "1": restarts the reload counter.
When this bit is read, it always returns "0".</td></tr>
<tr><td>bit2</td><td>UPCL:
LIN-UART programmable clear bit (LIN-UART software reset)</td><td>This bit resets the LIN-UART.
Writing "0": no effect.
Writing "1": resets the LIN-UART immediately (LIN-UART software reset). However, the register settings are maintained. At that time, transmission and reception are suspended. All of the transmit/receive interrupt sources (TDRE, RDRF, LBD, PE, ORE, FRE) are cleared.
Reset the LIN-UART after disabling the interrupt and transmission.
In addition, after the LIN-UART is reset, the receive data register is cleared (RDR = 00_H), and the reload counter is restarted.
When this bit is read, it always returns "0".</td></tr>
<tr><td>bit1</td><td>SCKE:
LIN-UART serial clock output enable bit</td><td>This bit controls the serial clock I/O port.
Writing "0": the SCK pin functions as a general-purpose I/O port or a serial clock input pin.
Writing "1": the SCK pin functions as a serial clock output pin, and outputs the clock in operating mode 2 (synchronous).
Note: To use the SCK pin as a serial clock input pin (SCKE = 0), enable the use of the input port by setting the bit in the DDR register corresponding to the general-purpose I/O port sharing the same pin with SCK. In addition, select the external clock (EXT = 1) using the external serial clock source select bit.
When set as a serial clock output pin (SCKE = 1), the SCK pin functions as a serial clock output pin regardless of the state of the general-purpose I/O port sharing the same pin with SCK.</td></tr>
<tr><td>bit0</td><td>SOE:
LIN-UART serial data output enable bit</td><td>This bit enables or disables output of serial data.
Writing "0": the SOT pin becomes a general-purpose I/O port.
Writing "1": the SOT pin becomes a serial data output pin (SOT).
When set as a serial data output (SOE = 1), the SOT pin functions as a serial data output pin (SOT) regardless of the state of the general-purpose I/O port sharing the same pin with SOT.</td></tr>
</table>

16.4.3 LIN-UART Serial Status Register (SSR)

The LIN-UART serial status register (SSR) is used to check the status of transmission, reception and error, and to enable and disable interrupts.

■ LIN-UART Serial Status Register (SSR)

Figure 16.4-4 LIN-UART Serial Status Register (SSR)

Address 0052H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE
R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W

Initial value: 00001000B

TIE	Transmit interrupt request enable bit
0	Disable transmit interrupts.
1	Enable transmit interrupts.

RIE	Receive interrupt request enable bit
0	Disable receive interrupts.
1	Enable receive 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	Receive data full flag bit
0	Receive data register (RDR) is empty.
1	Receive data register (RDR) has data.

FRE	Framing error flag bit
0	No framing error
1	Framing error occurs

ORE	Overrun error flag bit
0	No overrun error
1	Overrun error occurs

PE	Parity error flag bit
0	No parity error
1	Parity error occurs

R/W : Readable/writable (The read value is the same as the write value.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
: Initial value

Table 16.4-3 Functions of Bits in Serial Status Register (SSR)

Bit name		Function
bit7	PE: Parity error flag bit	This bit detects the parity error in received data. • This bit is set to "1" when a parity error occurs during reception with PE = 1, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • When both the PE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the receive data register (RDR) is invalid.
bit6	ORE: Overrun error flag bit	This bit detects the 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). • When both the ORE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the receive data register (RDR) is invalid.
bit5	FRE: Framing error flag bit	This bit detects the 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). • When both the FRE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the LIN-UART receive data register (RDR) is invalid.
bit4	RDRF: Receive data full flag bit	This flag shows the status of the LIN-UART receive data register (RDR). • This bit is set to "1" when received data is loaded into RDR, and cleared to "0" by reading the receive data register (RDR). • When both the RDRF bit and the RIE bit are "1", a receive interrupt request is output.
bit3	TDRE: Transmit data empty flag bit	This flag shows the status of the LIN-UART 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. When data is loaded into the transmit shift register and data transfer starts, this bit is set to "1", indicating that the TDR does not have valid data. • When both the TDRE bit and the TIE bit are "1", a transmit interrupt request is output. • When the TDRE bit is "1", setting the LBR bit in the LIN-UART extended communication control register (ECCR) to "1" changes the TDRE bit to "0". After the LIN synch break is generated, the TDRE bit returns to "1". Note: The initial value of TDRE is "1".
bit2	BDS: Transfer direction select bit	This bit specifies whether the transfer of serial data starts from the least significant bit (LSB-first, BDS = 0) or from the most significant bit (MSB-first, BDS = 1). Note: When data is written to or read from the serial data register, the data on the upper side and that on the lower side are swapped. Therefore, if the BDS bit is modified after data is written to the RDR register, the data in the RDR register becomes invalid. In operating mode 3 (LIN), the BDS bit is fixed at "0".
bit1	RIE: Receive interrupt request enable bit	This bit enables or disables the receive interrupt request output to the interrupt controller. When both the RIE bit and the receive data flag bit (RDRF) are "1", or when one or more error flag bits (PE, ORE, FRE) is "1", a receive interrupt request is output.
bit0	TIE: Transmit interrupt request enable bit	This bit enables or disables the transmit interrupt request output to the interrupt controller. When both the TIE bit and the TDRE bit are "1", a transmit interrupt request is output.

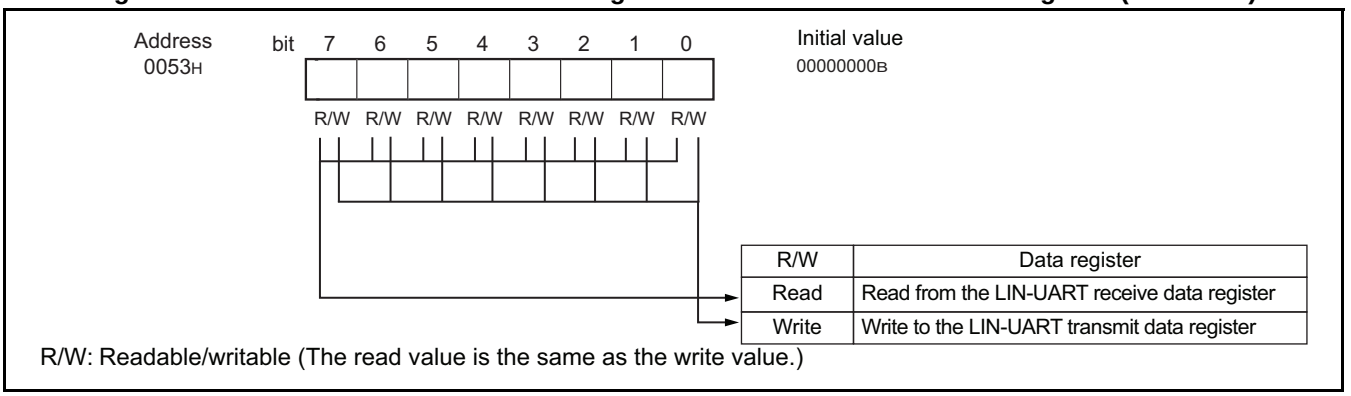
16.4.4 LIN-UART Receive Data Register/LIN-UART Transmit Data Register (RDR/TDR)

The LIN-UART receive data register and the LIN-UART transmit data register are located at the same address. If read, they function as the receive data register; if written, they function as the transmit data register.

■ LIN-UART Receive Data Register (RDR)

Figure 16.4-5 shows the bit configuration of LIN-UART receive data register/LIN-UART transmit data register.

Figure 16.4-5 LIN-UART Receive Data Register/LIN-UART Transmit Data Register (RDR/TDR)



R/W	Data register
Read	Read from the LIN-UART receive data register
Write	Write to the LIN-UART transmit data register

The LIN-UART receive data register (RDR) is the data buffer register for serial data reception.

Serial input data signals transmitted to the serial input pin (SIN pin) are converted by the shift register, and the converted data is stored in the LIN-UART receive data register (RDR).

If the data length is 7 bits, the MSB (RDR:D7) is "0".

The receive data full flag bit (SSR:RDRF) is set to "1" when received data is stored in the LIN-UART receive data register (RDR). If the receive interrupt has been enabled (SSR:RIE = 1), a receive interrupt request is made.

Read the LIN-UART receive data register (RDR) with the receive data full flag bit (SSR:RDRF) being "1". The receive data full flag bit (SSR:RDRF) is automatically cleared to "0" if the LIN-UART receive data register (RDR) is read. In addition, the receive interrupt is cleared when the receive interrupt has been enabled and no errors occur.

When a reception error occurs (any of SSR:PE, ORE, or FRE is "1"), the data in the LIN-UART receive data register (RDR) becomes invalid.

■ LIN-UART Transmit Data Register (TDR)

The LIN-UART transmit data register (TDR) is the data buffer register for serial data transmission.

If the data to be transmitted is written to the LIN-UART transmit data register (TDR) when transmission has been enabled (SCR:TXE = 1), the transmit data is transferred to the transmit shift register to convert to serial data, and the serial data is output from the serial data output pin (SOT pin).

If the data length is 7 bits, the data in the MSB (TDR:D7) is invalid.

The transmit data empty flag (SSR:TDRE) is cleared to "0" when 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 transmit shift register and data transmission starts.

If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written to TDR. If the transmit interrupt has been enabled, a transmit interrupt is generated. Write the next transmit data to TDR written after a 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 receive data register is a read-only register. Since both registers are located at the same address, the write value and the read value are different. Thus, the read-modify-write (RMW) type of instruction, such as the INC instruction and the DEC instruction, cannot be used.

16.4.5 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, continuous clock output in LIN-UART synchronous clock mode and sampling clock edge.

■ Bit Configuration of LIN-UART Extended Status Control Register (ESCR)

Figure 16.4-6 shows the bit configuration of the LIN-UART extended status control register (ESCR). Table 16.4-4 lists the function of each bit.

Figure 16.4-6 Bit Configuration of LIN-UART Extended Status Control Register (ESCR)

Address 0054H — Initial value 00000100B

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES
R/W	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W

SCES	Sampling clock edge select bit (mode 2)
0	Sampling with rising clock edge (normal)
1	Sampling with falling clock edge (inverted clock)

CCO	Continuous clock output enable bit (mode 2)
0	Disable continuous clock output
1	Enable continuous clock output

SIOP	Serial I/O pin direct access bit	
	Write (SOPE = 1)	Read
0	Fix SOT pin at "0"	Read the value of SIN pin
1	Fix SOT pin at "1"	

SOPE	Serial output pin direct access enable bit
0	Disable serial output pin direct access
1	Enable serial output pin direct access

LBL0	LBL1	LIN synch break length select bits
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

LBD	LIN synch break detection flag bit	
	Write	Read
0	LIN synch break detection flag clear	No LIN synch break detection
1	No effect	With LIN synch break detection

LBIE	LIN synch break detection interrupt enable bit
0	Disable LIN synch break detection interrupt
1	Enable LIN synch break detection interrupt

R/W: Readable/writable
(The read value is the same as the write value.)

R(RM1), W: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)

: Initial value

Table 16.4-4 Functions of Bits 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 at "0" in operating mode 1 and operating mode 2.
bit6	LBD: LIN synch break detection flag bit	This bit detects the LIN synch break. This bit is set to "1" when a LIN synch break is detected in operating mode 3 (the serial input is "0" when bit width is 11 bits or more). If "0" is written to the LBD bit, the LBD bit and the interrupt are cleared. Although the bit always returns "1" if read by the read-modify-write (RMW) type of instruction, this does not indicate that a LIN synch break has been 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 select 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*	This bit enables or disables direct writing to the SOT pin. Setting this bit to "1" when serial data output has been enabled (SMR:SOE = 1) enables direct writing to the SOT pin.*
bit2	SIOP: Serial I/O pin direct access bit*	This bit controls direct access to the serial I/O pin. The SIOP bit always returns the value of the SIN pin if read by a normal read instruction. If direct access to the serial output pin is enabled (SOPE = 1), the value written to this bit is reflected in the SOT pin.* Note: When the bit manipulation instruction is used, the SIOP bit returns the bit value of the SOT pin in the read cycle.
bit1	CCO: Continuous clock output enable bit	This bit enables or disables continuous serial clock output from the SCK pin. In operating mode 2 (synchronous) in which the serial clock transmission side is selected, setting the CCO bit to "1" enables the continuous serial clock output from the SCK pin when the SCK pin is used as an clock output pin. Note: When the CCO bit is "1", set the SSM bit in the ECCR register to "1".
bit0	SCES: Sampling clock edge select bit	This bit selects a sampling edge. In operating mode 2 (synchronous) in which the serial clock reception side is selected, setting the SCES bit to "1" switches the sampling edge from the rising edge to the falling edge. In operating mode 2 (synchronous) in which the serial clock transmission side is selected (ECCR:MS = 0), when the SCK pin is used as an clock output pin, the internal serial clock signal and the output clock signal are inverted. In operating mode 0/1/3, set this bit to "0".

***: 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"	

16.4.6 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 16.4-7 shows the bit configuration of the LIN-UART extended communication control register (ECCR). Table 16.4-5 lists the function of each bit.

Figure 16.4-7 Bit Configuration of LIN-UART Extended Communication Control Register (ECCR)

Address 0055H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
RX,W0	R0,W	R/W	R/W	R/W	RX,W0	R/WX	R/WX

Initial value: 000000XXB

TBI*	Transmit bus idle detection flag bit
0	Transmission in progress
1	No transmission

RBI*	Receive bus idle detection flag bit
0	Reception in progress
1	No reception

Reserved bit
The read value is undefiend. Always set this bit to "0".

SSM	Start/stops mode enable bit (mode 2)
0	No start/stop bits
1	Start/stop bits available

SCDE	Serial clock delay enable bit (mode 2)
0	Disables clock delay
1	Enables clock delay

MS	Serial clock transmission/reception side select bit (mode 2)
0	Transmission side (serial clock generation)
1	Reception side (external serial clock reception)

LBR	LIN synch break generation bit (mode 3)	
	Write	Read
0	No effect	"0" is always read.
1	LIN synch break generation	

Reserved bit
The read value is undefined. Always set this bit to "0".

R/W : Readable/writable (The read value is the same as the write value.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R0,W : Write only (Writable. "0" is read,)
RX,W0 : Reserved bit (The write value is "0"; the read value is "1".)
X : Indeterminate
: Initial value

*: Unused when SSM = 0 in operating mode 2

Table 16.4-5 Functions of Bits in LIN-UART Extended Communication Control Register (ECCR)

Bit name		Function
bit7	Reserved bit	The read value is undefined. Always set this bit to "0".
bit6	LBR: LIN synch break generation bit	In operating mode 3, if this bit is set to "1", a LIN synch break whose length is specified in the LBL0/LBL1 bit in the ESCR register is generated. In operating mode 0/1/2, set this bit to "0".
bit5	MS: Transmission side/ reception side of serial clock select bit	This bit selects the transmission side/reception side of the serial clock in operating mode 2. If the transmission side (MS = 0) is selected, the LIN-UART generates a synchronous clock. If the reception side (MS = 1) is selected, the LIN-UART receives an external serial clock. In mode 0/1/3, this bit is fixed at "0". Modify this bit only when the SCR:TXE bit is "0". Note: When the reception side is selected, the external clock must be selected as 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	In operating mode 2 in which the serial clock transmission side is selected, if the SCDE bit is set to "1", a delayed serial clock as shown in Figure 16.7-5 is output. The function of outputting delayed serial clock can be used in the Serial Peripheral Interface (SPI). This bit is fixed at "0" in operating mode 0/1/3.
bit3	SSM: start/stop bits mode enable bit	In operating mode 2, if this bit is set to "1", the start/stop bits are added to the synchronous data format. In operating mode 0/1/3, this bit is fixed at "0".
bit2	Reserved bit	The read value is undefined. Always set this bit to "0".
bit1	RBI: Receive bus idle detection flag bit	If the SIN pin is at "H" level and no reception is performed, this bit is "1". Do not use this bit when SSM = 0 in operating mode 2.
bit0	TBI: Transmit bus idle detection flag bit	If there is no transmission on the SOT pin, this bit is "1". Do not use this bit when SSM = 0 in operating mode 2.

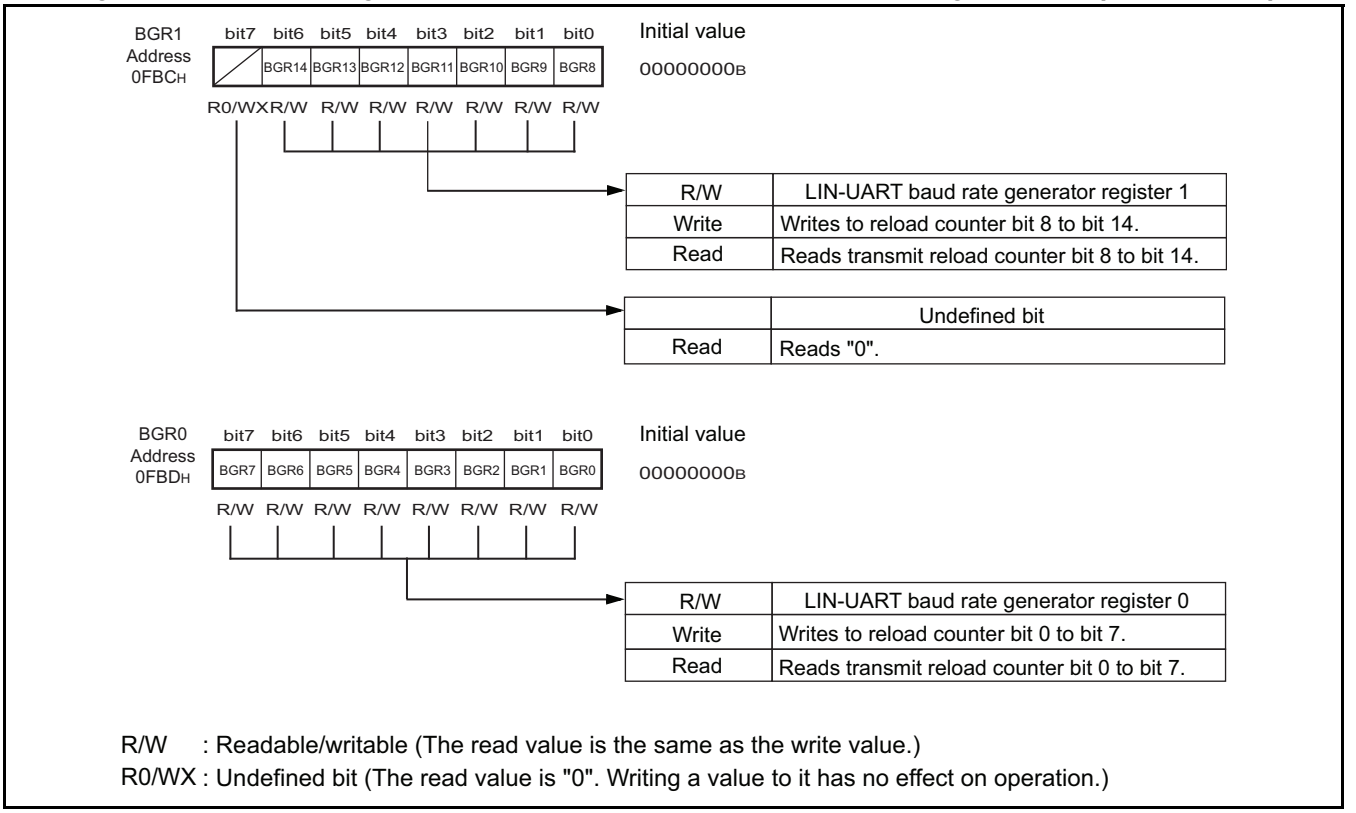
16.4.7 LIN-UART Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) set the division ratio of the serial clock. Also, the count value in the transmit reload counter is read from this generator.

■ LIN-UART Bit Configuration of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 16.4-8 shows the bit configuration of LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 16.4-8 Bit Configuration of LIN-UART Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



The LIN-UART baud rate generator registers set the division ratio of the serial clock.

BGR1 corresponds to the upper bits and BGR0 to the lower bits. The reload value of the counter can be written to and the transmit reload counter value can be read from BGR1 and BRG0. In addition, BGR1 and BGR0 can be accessed by byte access and word access.

Writing a reload value to the LIN-UART baud rate generator registers causes the reload counter to start counting.

Note:

Write to this register only when the LIN-UART stops.

16.5 LIN-UART Interrupts

The LIN-UART has receive interrupts and transmit interrupts, which are generated by the following sources. An interrupt number and an interrupt vector are assigned to each interrupt. In addition, it has a LIN synch field edge detection interrupt function using the 8/16-bit composite timer interrupt.

- **Receive interrupt**
  A receive interrupt occurs when received data is set in the LIN-UART receive data register (RDR), or when a receive error occurs, or when a LIN synch break is detected.
- **Transmit interrupt**
  A transmit interrupt occurs when transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register, and data transmission starts.

■ Receive Interrupt

Table 16.5-1 shows the control bits and interrupt sources of receive interrupts.

Table 16.5-1 Interrupt Control Bits and Interrupt Sources of Receive Interrupts

Interrupt request flag bit	Flag register	Operating mode				Interrupt source	Interrupt source enable bit	Interrupt request flag clear
		0	1	2	3			
RDRF	SSR	❍	❍	❍	❍	Writing received data to RDR	SSR:RIE	Read received data
ORE	SSR	❍	❍	❍	❍	Overrun error		Write "1" to receive 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

❍ : Bit to be used
× : Unused bit
△ : Usable only when ECCR:SSM = 1

● Receive interrupts

If one of the following operations occurs in reception mode, the bit in the LIN-UART serial status register (SSR) corresponding to that operation is set to "1".

Data reception completed

Received data is transferred from the LIN-UART serial input shift register to the LIN-UART receive data register (RDR) (RDRF = 1).

Overrun error

With RDRF = 1, the next serial data is received while the CPU has not read the RDR register. (ORE = 1).

Framing error

A stop bit reception error occurs (FRE = 1).

Parity error

A parity detection error occurs (PE = 1).

A receive interrupt request is made if the receive interrupt has been enabled (SSR:RIE = 1) when one of the above flag bits is "1".

RDRF flag is automatically cleared to "0" if the LIN-UART receive data register (RDR) is read. All of the error flags are cleared to "0" if "1" is written to the receive error flag clear bit (CRE) in the LIN-UART serial control register (SCR).

Note:

The CRE flag is write-only, and keeps "1" for one clock cycle after "1" is written to the bit.

● LIN synch break interrupts

In operating mode 3, the LIN synch break interrupt functions when the LIN-UART performs LIN slave operation.

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 composite timer interrupt is generated within the LIN synch field.

To detect a LIN synch break, the reception must be disabled (SCR:RXE = 0).

■ Transmit Interrupts

Table 16.5-2 shows the control bit and interrupt source of the transmit interrupt.

Table 16.5-2 Interrupt Control Bit and Interrupt Source of Transmit Interrupt

Interrupt request flag bit	Flag register	Operating mode				Interrupt source	Interrupt source enable bit	Interrupt request flag clear
		0	1	2	3			
TDRE	SSR	❍	❍	❍	❍	Transmit register is empty	SSR:TIE	Write transmit data

❍: Bit to be used

● Transmit interrupts

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 transmit shift register, and data transmission starts. In this case, if the transmit interrupt has been enabled (SSR:TIE = 1), a transmit interrupt request is made.

Note:

Since the initial value of TDRE is "1" after a hardware reset/software reset, if the TIE bit is set to "1" after a hardware reset/software reset, an interrupt is generated immediately. 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 Composite Timer Interrupt)

Table 16.5-3 shows the control bits and interrupt sources of the LIN synch field edge detection interrupt.

Table 16.5-3 Interrupt Control Bits and Interrupt Sources of LIN Synch Field Edge Detection Interrupt

Interrupt request flag bit	Flag register	Operating mode				Interrupt source	Interrupt source enable bit	Interrupt request flag clear
		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		

❍ : Bit to be used
× : Unused bit

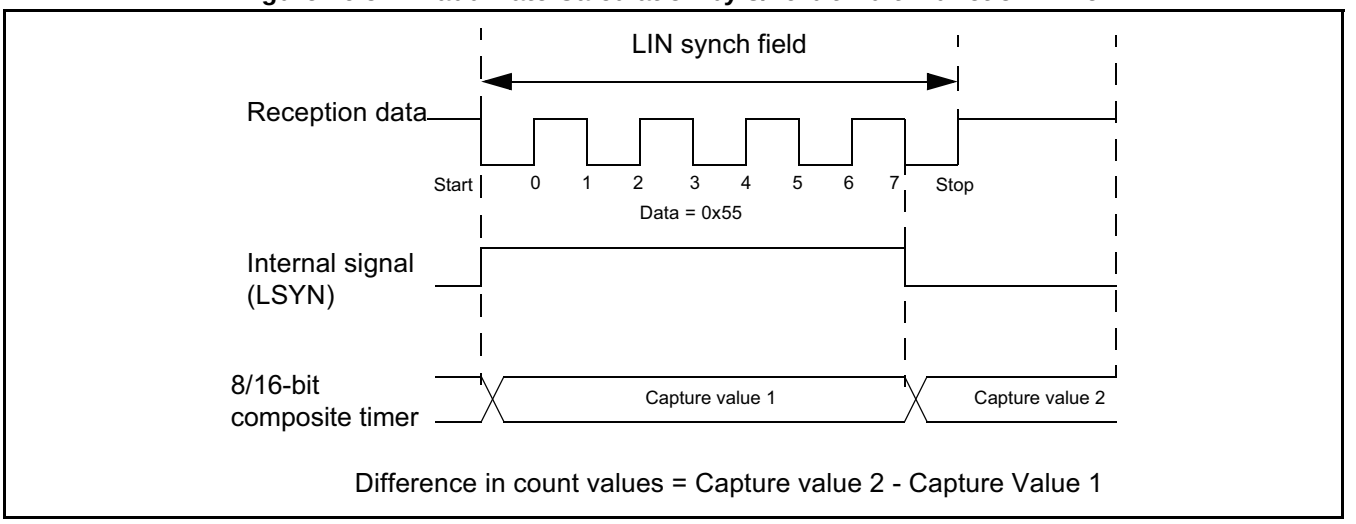
● LIN synch field edge detection interrupt (8/16-bit composite timer interrupt)

In operating mode 3, the LIN synch field edge detection interrupt functions when the LIN-UART performs LIN slave operation.

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. Between both falling edges, a 8/16-bit composite timer interrupt is generated, provided that the 8/16-bit composite timer has been configured to receive internal signals and detect rising edges and falling edges and the 8/16-bit composite interrupt has been enabled.

The difference in the count values detected by the 8/16-bit composite timer (see Figure 16.5-1) is equivalent to eight bits of the master serial clock. A new baud rate can be calculated from this value.
After being set, a new baud rate becomes effective from the falling edge detected at the next start bit set.

Figure 16.5-1 Baud Rate Calculation by 8/16-bit Multi-Function Timer



■ Registers and Vector Table Addresses Related to LIN-UART Interrupts

Table 16.5-4 Registers and Vector Table Addresses Related to LIN-UART Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Reception	IRQ7	ILR1	L07	$FFFC_H$	$FFFD_H$
Transmission	IRQ8	ILR2	L08	$FFEA_H$	$FFEB_H$

16.5.1 Timing of Receive Interrupt Generation and Flag Set

A receive interrupt is generated when reception is completed (SSR:RDRF) or when a reception error occurs (SSR:PE, ORE, FRE).

■ Timing of Receive Interrupt Generation and Flag Set

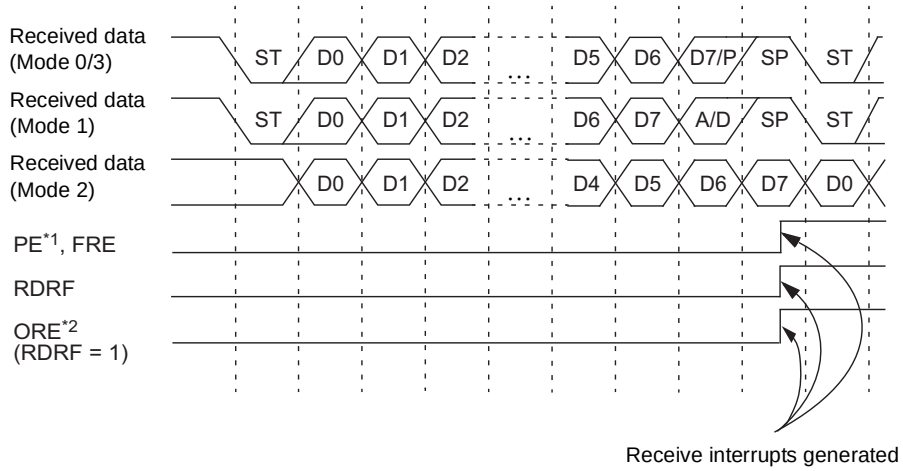
Received data is stored in the LIN-UART receive data register (RDR) when the first stop bit is detected in operating mode 0/1/2(SSM =1)/3, or when the last data bit is detected in operating mode 2 (SSM = 0). When reception is completed (SSR:RDRF = 1), or when a reception error occurs (SSR:PE, ORE, FRE = 1), an error flag corresponding to one of the events mentioned above is set. If the receive interrupt has been enabled (SSR:RIE = 1) when an error flag is set, a receive interrupt is generated.

Note:

In all operating modes, when a receive error occurs, data in the LIN-UART receive data register (RDR) becomes invalid.

Figure 16.5-2 shows the timing of reception and flag set.

Figure 16.5-2 Timing of Reception and Flag Set

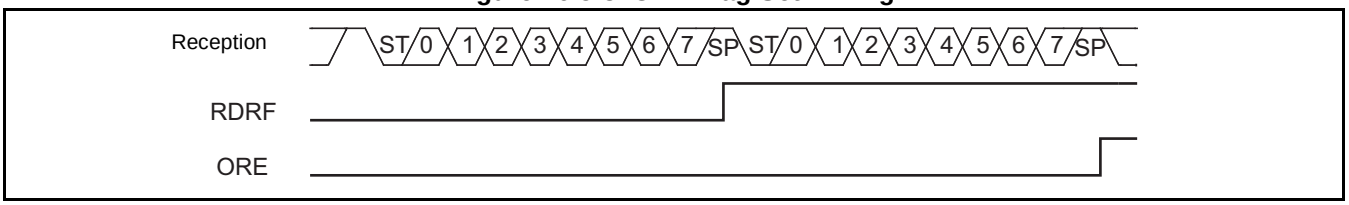


* 1: The PE flag is always "0" in operating mode 1 and operating mode 3.
* 2: An overrun error occurs if the next data is transferred before received data is read (RDRF = 1).
ST: Start bit, SP: Stop bit, A/D: Mode 1 (multiprocessor) address data select bit

Note:

Figure 16.5-2 does not show all reception operations in mode 0. It only shows two examples of reception operations using different communication formats. One reception operation uses 7-bit data, a parity bit (parity bit = "even parity" or "odd parity") and one stop bit. The other uses 8-bit data, no parity bit and one stop bit.

Figure 16.5-3 ORE Flag Set Timing



16.5.2 Timing of Transmit Interrupt Generation and Flag Set

A transmit interrupt is generated when transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register and then data transmission starts.

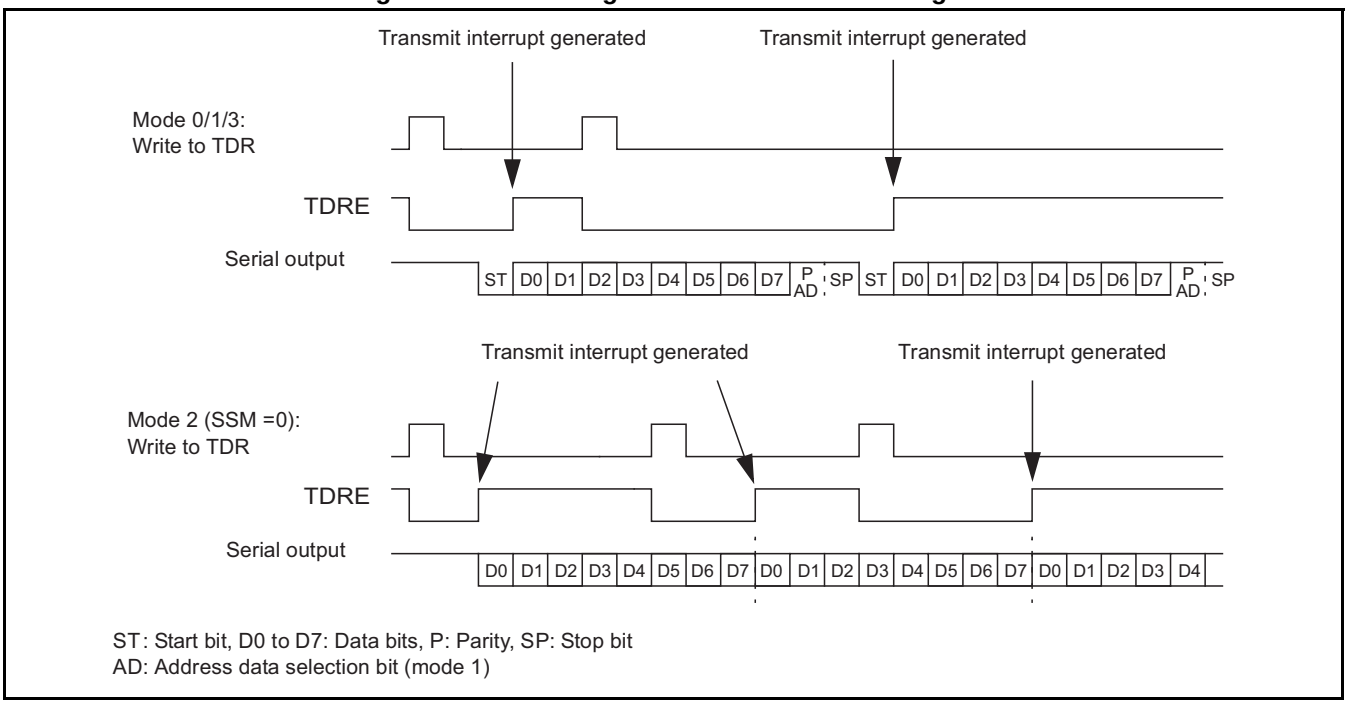
■ Timing of Transmit Interrupt Generation and Flag Set

When the data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register and the transmission of that data starts, the next data can be written to the TDR register (SSR:TDRE = 1). At the start of the data transmission, if the transmit interrupt has been enabled (SSR:TIE = 1), a transmit interrupt is generated.

The TDRE bit is a read-only bit, and is cleared to "0" only when data is written to the LIN-UART transmit data register (TDR).

Figure 16.5-4 shows the timing of transmission and flag set.

Figure 16.5-4 Timing of Transmission and Flag Set



Note:

Figure 16.5-4 does not show all transmission operations in mode 0. It only shows an example of a transmission operation using 8-bit data, a parity bit ("even parity" or "odd parity") and one stop bit.

No parity bit is transmitted in mode 3, or in mode 2 with SSM = 0.

■ Transmit Interrupt Request Generation Timing

With the transmit interrupt having been enabled (SSR:TIE = 1), if the TDRE flag is set to "1", a transmit interrupt is generated.

Note:

Since the initial value of the TDRE bit is "1", a transmit interrupt is generated immediately after the transmit interrupt is enabled (SSR:TIE = 1). When deciding the timing of enabling the transmit interrupt, take into consideration that the TDRE bit can be cleared only by writing new data to the LIN-UART transmit data register (TDR).

See "APPENDIX B Table of Interrupt Sources" for interrupt request numbers and vector table addresses of respective peripheral functions.

16.6 LIN-UART Baud Rate

The input clock (transmit/receive clock source) of the LIN-UART can be selected from one of the following:

- **Input a machine clock to a baud rate generator (reload counter).**
- **Input an external clock to a baud rate generator (reload counter).**
- **Use an external clock (SCK pin input clock) directly.**

■ LIN-UART Baud Rate Selection

The baud rate can be selected from one of following three types. Figure 16.6-1 shows the baud rate selection circuit.

● Baud rate derived from the internal clock divided by the dedicated baud rate generator (reload counter)

There are two internal reload counters, corresponding to the transmit serial clock and the receive serial clock respectively. The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock by the value set in BGR1 and BGR0.

The baud rate is used in asynchronous mode and in synchronous mode (transmit side of the serial clock).

As for clock source settings, select the internal clock and use the baud generator clock (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 registers 1, 0 (BGR0, BGR1).

The reload counter divides the external clock by the value set in BGR1 and BGR0.

The baud rate is used in asynchronous mode.

As for clock source settings, select the external clock and use the baud generator clock (SMR:EXT = 1, OTO = 0).

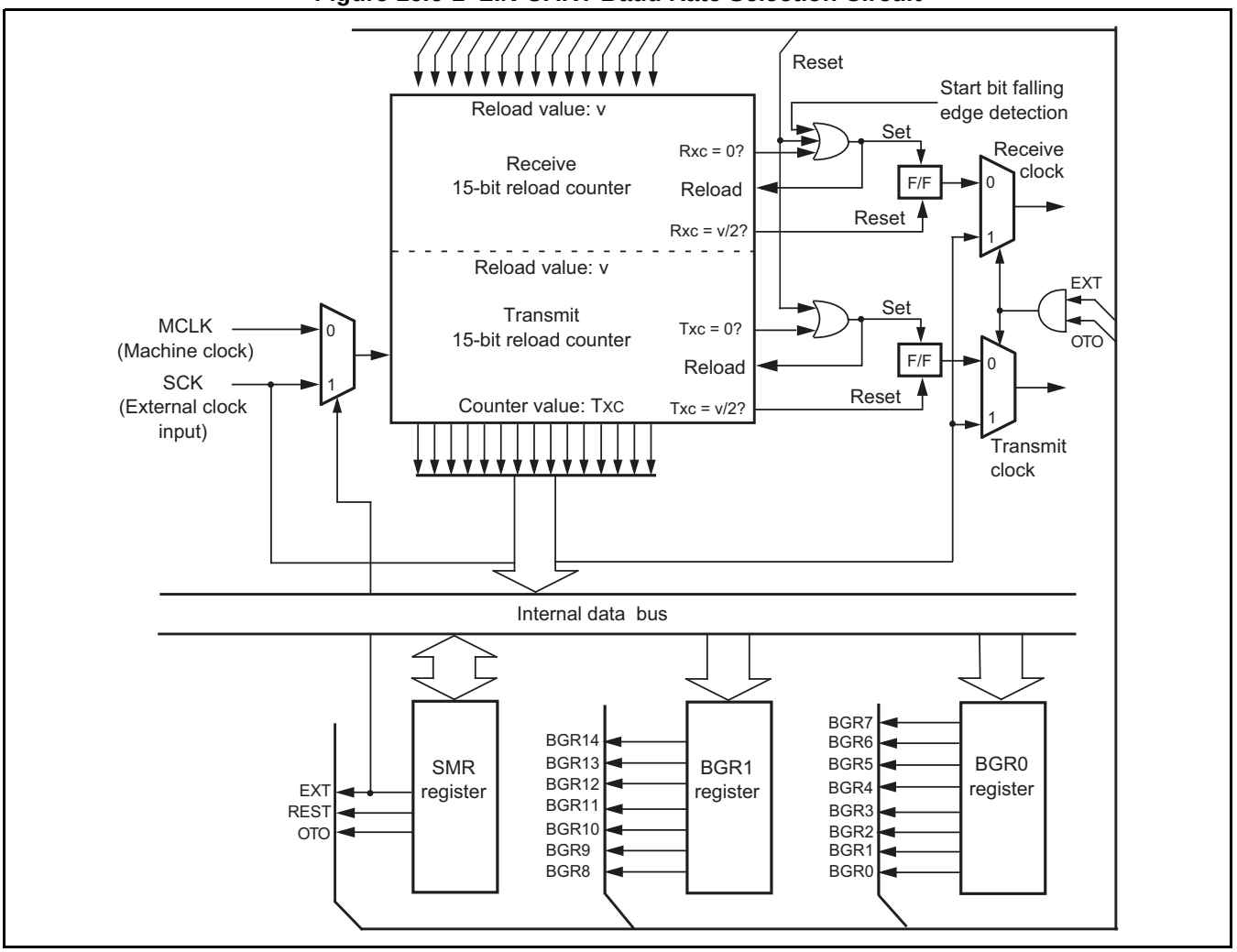
● Baud rate by the external clock (one-to-one mode)

The clock input from the clock input pin (SCK) of the LIN-UART is used as the baud rate (slave operation in operating mode 2 (synchronous) (ECCR:MS = 1)).

It is used in synchronous mode (serial clock reception side).

To set the clock source, select the external clock and its direct use (SMR:EXT = 1, OTO = 1).

Figure 16.6-1 LIN-UART Baud Rate Selection Circuit



16.6.1 Baud Rate Setting

This section shows baud rate settings and the result of calculating the serial clock frequency.

■ Baud Rate Calculation

The two 15-bit reload counters are set by the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0).

The equation for the baud is shown below.

Reload value:

$$v = (\frac{MCLK}{b}) - 1$$

v: Reload value, b: Baud rate, MCLK: Machine clock, or external clock frequency

Calculation example

Assuming that the machine clock is 10 MHz, the internal clock is used, and the baud rate is set to 19200 bps:

Reload value:

$$v = \left(\frac{10 \times 10^6}{19200} \right) - 1 = 519.83... \approx 520$$

Thus, the actual baud rate can be calculated as shown below.

$$b = \frac{MCLK}{(v + 1)} = \frac{10 \times 10^6}{521} = 19193.8579$$

Note:

The reload counter stops if the reload value is set to "0". Therefore, set the smallest reload value to "1".

For transmission/reception in asynchronous mode, since five times of oversampling have to be done before the reception value is determined, the reload value must be set to at least "4".

■ Reload Value and Baud Rate of Each Clock Speed

Table 16.6-1 shows the reload value and baud rate of each clock speed.

Table 16.6-1 Reload Value and Baud Rate

Baud rate	8 MHz (MCLK)		10 MHz (MCLK)		16 MHz (MCLK)		16.25 MHz (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 represents 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 operating mode 2 (synchronous), 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.

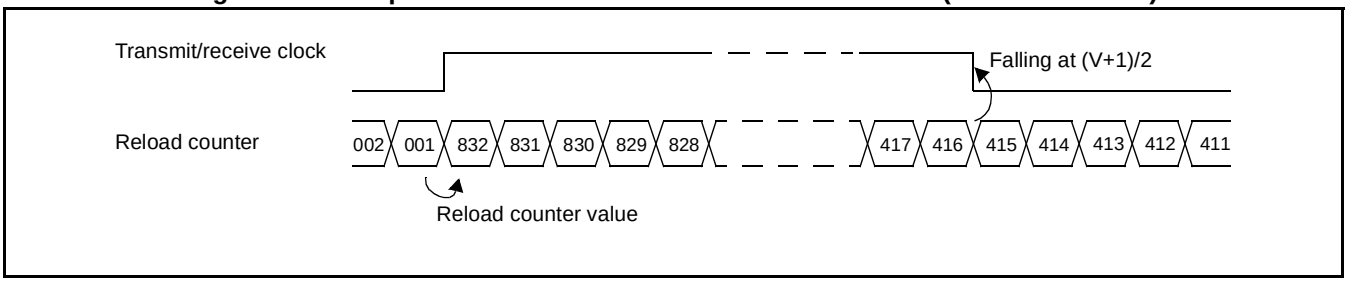
Note:

The external clock signal is synchronized with the internal clock (MCLK: machine clock) in the LIN-UART. Therefore, if the external clock becomes not divisible because its cycle is faster than half the cycle of the internal clock, the external clock signal becomes unstable.
For the value of the SCK clock, refer to the data sheet of this microcontroller.

■ Operation of Dedicated Baud Rate Generator (Reload Counter)

Figure 16.6-2 shows the operation of two reload counters using a reload value "832" as an example.

Figure 16.6-2 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.

16.6.2 Reload Counter

This block is a 15-bit reload counter functioning as a dedicated baud rate generator. It generates the transmit/receive clock from the external clock or internal clock.
The count value in the transmit reload counter can be read from the LIN-UART baud rate generator registers 1, 0 (BGR 1 and BGR 0).

■ Functions of Reload Counter

There are two types of reload counter, the transmit reload counter and the receive reload counter. The reload counter functions as a dedicated baud rate generator. It consists of a 15-bit register for a reload value and generates the transmit/receive clock from the external clock or internal clock. The count value in the transmit reload counter can be read from the LIN-UART baud rate generator registers 1, 0 (BGR 1 and BGR 0).

● Start of counting

Writing a reload value to the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0) causes the reload counter to start counting.

● Restart

The reload counter restarts under the following conditions.

For both transmit/receive reload counters

- LIN-UART programmable reset (SMR:UPCL bit)
- Programmable restart (SMR:REST bit)

For the receive reload counter

- Detection of a start bit falling edge in asynchronous mode

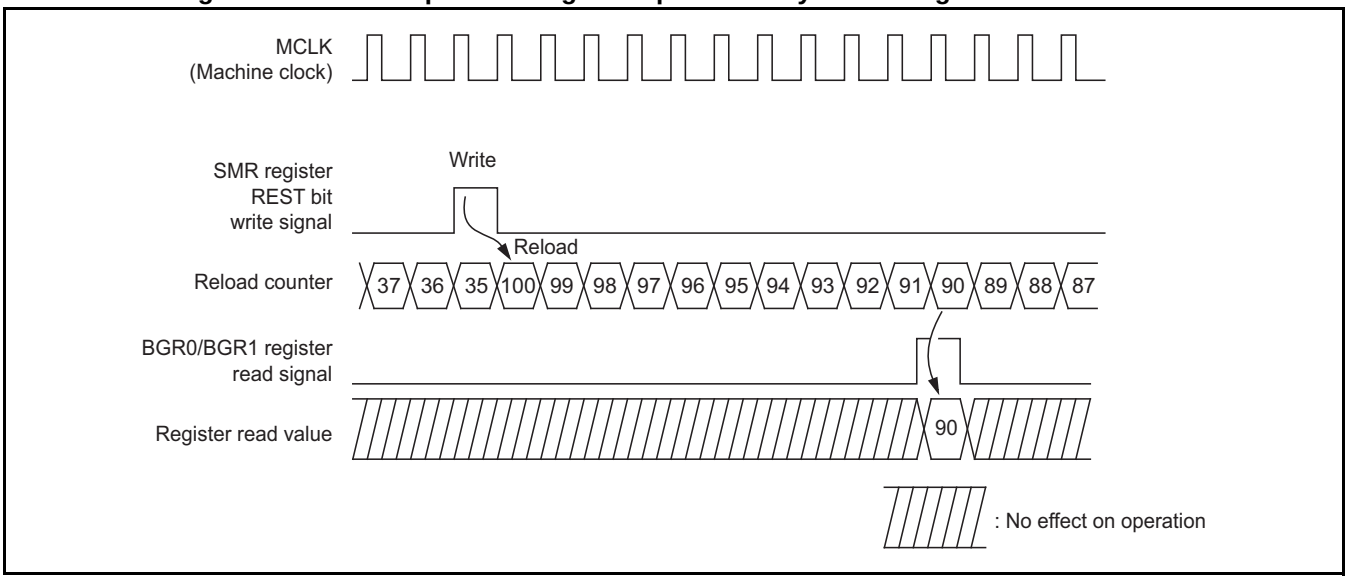
● Simple timer function

If the LIN-UART serial mode register (SMR) is set to "1", the two reload counters restart at the next clock cycle.

This function enables the transmit reload counter to be used as a simple timer.

Figure 16.6-3 shows an example of using this function (when the reload value is 100).

Figure 16.6-3 Example of Using a Simple Timer by Restarting the Reload Timer



The number of machine clock cycles "cyc" after the restart in this example is obtained by the following equation.

cyc = v - c + 1 = 100 - 90 + 1 = 11

v: Reload value, c: Reload counter value

Note:

The transmit reload counter restarts also when the LIN-UART is reset by writing "1" to the SMR:UPCL bit.

Automatic restart (receive reload counter only)

The receive reload counter restarts when the start bit falling edge is detected in asynchronous mode. This automatic restart function is to synchronize the receive shift register with the received data.

● Clear counter

When a reset occurs, the reload values in the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0) and the reload counter are cleared to "00_H", and the reload counter stops.

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 kept.

If the restart setting is used (writing "1" to SMR:REST), the reload counter restarts without the counter value being cleared to "00_H".

16.7 Operations of LIN-UART and LIN-UART Setting Procedure Example

The LIN-UART performs bi-directional serial communication in operating mode 0/2, master/slave communication in operating mode 1, LIN master/slave communication in operating mode 3.

■ Operations of LIN-UART

● Operating mode

The LIN-UART has four operating modes (0 to 3), providing different connection methods between CPUs and different data transfer methods as shown in Table 16.7-1.

Table 16.7-1 LIN-UART Operating Modes

Operating mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multiprocessor mode	7 bits or 8 bits +1*	-	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	
3	LIN mode	8 bits	-	Asynchronous	1 bit	LSB first

- : Unavailable
* : "+1" is the address/data select bit (A/D) 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 operating modes.

Table 16.7-2 LIN-UART Operating 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:

- In operating mode 1, a system connecting to a master/slave supports both master operations and slave operations.
- In operating mode 3, the communication format is fixed at "8-bit data, no parity bit, one stop bit, LSB-first".
- If the operating mode is changed, all transmission operations and reception operations are canceled, and the LIN-UART waits for the next transmission/reception.

■ Inter-CPU Connection Method

The external clock one-to-one connection (normal mode) and the master/slave connection (multiprocessor mode) can be selected as an inter-CPU connection method. In either method, CPUs must use the same data length, parity setting, synchronization type, etc. Select their operating modes as follows.

- One-to-one connection: Both CPUs must use the either operating mode 0 or operating mode 2. Select the operating mode 0 for asynchronous method or the operating mode 2 for synchronous method. In addition, in operating mode 2, set one CPU as the transmission side of serial clock and the other as the reception side of serial clock.
- Master/slave connection: Select operating mode 1. Use the CPU as a master/slave system.

■ Asynchronous/Synchronous Method

As for the asynchronous method, the receive clock is synchronized with the receive start bit falling edge. As for the synchronous method, the receive clock can be synchronized with the clock signal of the serial clock transmission side, or with the clock signal of the LIN-UART operating as the transmission side.

■ 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. Execute the following operations to disable transmission or reception.

- To disable reception while it is in progress: wait until reception ends, read the receive data register (RDR), then disable reception.
- To disable transmission while it is in progress: wait until transmission ends, then disable transmission.

■ Setting Procedure Example

Below is an example of procedure for setting the LIN-UART.

● Initial settings

1) Set the port input (DDR0).
2) Set the interrupt level (ILR1, ILR2).
3) Set the data format and enable transmission/reception (SCR).
4) Select the operating mode and the baud rate, and enable pin output (SMR).
5) Set the baud rate generators 1, 0 (BGR1,BGR0).

16.7.1 Operations in Asynchronous Mode (Operating Mode 0, 1)

When the LIN-UART is used in operating mode 0 (normal mode) or operating mode 1 (multiprocessor mode), the transfer method is asynchronous transfer.

■ Operations in Asynchronous Mode

● Transmit/receive data format

Transmit/receive data always begins with a start bit ("L" level), followed by a specified data bits length, and ends 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 the parity bit is used, it is always placed between the last data bit and the first stop bit.

In operating mode 0, the data length can be 7 bits or 8 bits. The use of the parity can be selected. The stop bit length can also be selected from one and two.

In operating mode 1, the data length can be 7 bits or 8 bits. No parity is added while an address/data bit is added. The stop bit length can be selected from one and two.

Below is the equation for the bit length of a transmit/receive frame.

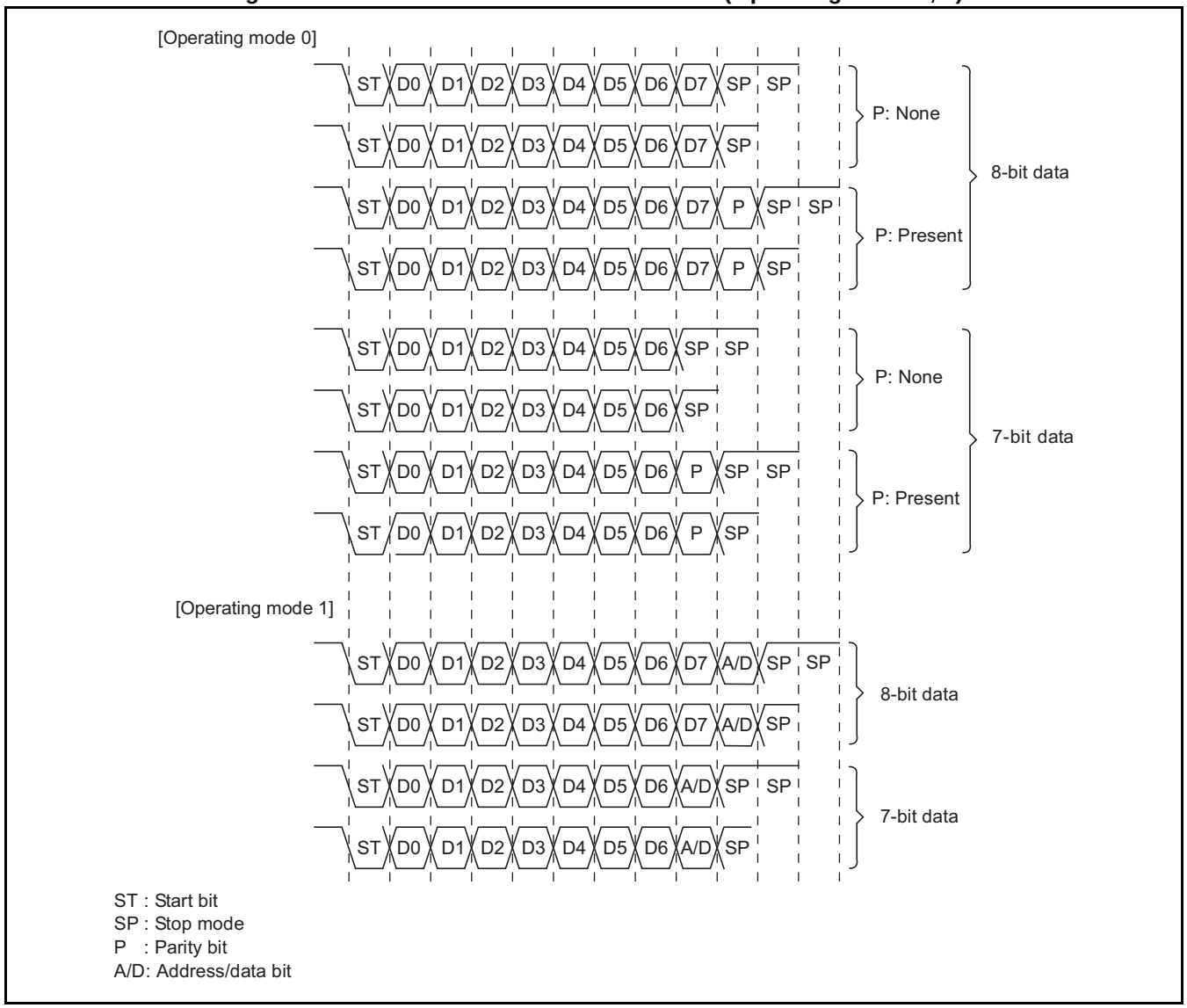
Length = 1 + d + p + s

(d = Number of data bits [7 or 8], p = parity [0 or 1],

s = Number of stop bits [1 or 2])

Figure 16.7-1 shows the transmit/receive data format in asynchronous mode (operating mode 0, 1).

Figure 16.7-1 Transmit/Receive Data Format (Operating 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 following order: 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 to the LIN-UART transmit data register (TDR). Writing data sets the TDRE flag to "0". If transmission has been enabled (SCR:TXE = 1) when the TDRE flag is set to "0", the data written to TDR is written to the transmit shift register, and, in the next serial clock cycle, the transmission of the data is started from the start bit.

With the transmit interrupt having been enabled (TIE = 1), if transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register, the TDRE flag is set to "1" and an interrupt is generated.

When the data length is set to 7 bits (CL = 0), bit 7 in the TDR register becomes an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Since the initial value of the transmit data empty flag bit (SSR:TDRE) is "1", an interrupt is generated immediately when the transmit interrupt is enabled (SSR:TIE =1).

● Reception

The reception is performed when reception is enabled (SCR:RXE =1). When a 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, an error flag (SSR:PE, ORE, FRE) is set. After the reception of one frame data ends, the received data is transferred from the receive shift register to the LIN-UART receive data register (RDR), and the receive data register full flag bit (SSR:RDRF) is set to "1". If the reception interrupt request has already been enabled (SSR:RIE = 1) at that time, a reception interrupt request is output.

To read the received data, first check the error flag status to ensure that reception has been executed normally, then read the data from the LIN-UART receive data register (RDR) if the reception is normal. If a reception error has occurred, perform error processing.

When the received data is read, the receive data register full flag bit (SSR:RDRF) is cleared.

When the data length is set to 7 bits (CL = 0), bit 7 in the TDR register becomes an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Data in the LIN-UART receive data register (RDR) becomes valid, provided that the receive data register full flag bit (SSR:RDRF) is set to "1" and no error has occurred (SSR:PE, ORE, FRE=0).

● Input clock

Use the internal clock or the external clock. For the baud rate, select the baud rate generator (SMR:EXT = 0 or 1, OTO = 0).

● Stop bit and reception bus idle flag

For transmission, the number of stop bits can be selected from one and two. If two stop bits are selected, both stop bits are detected during reception.

When the first stop bit is detected, the receive data register full flag (SSR:RDRF) is set to "1". When no start bit is detected afterward, the receive bus idle flag (ECCR:RBI) is set to "1", indicating that no reception is executed.

● Error detection

In operating mode 0, the parity error, the overrun error and the frame error can be detected.

In operating mode 1, the overrun error and the frame error can be detected. However, the parity error cannot be detected.

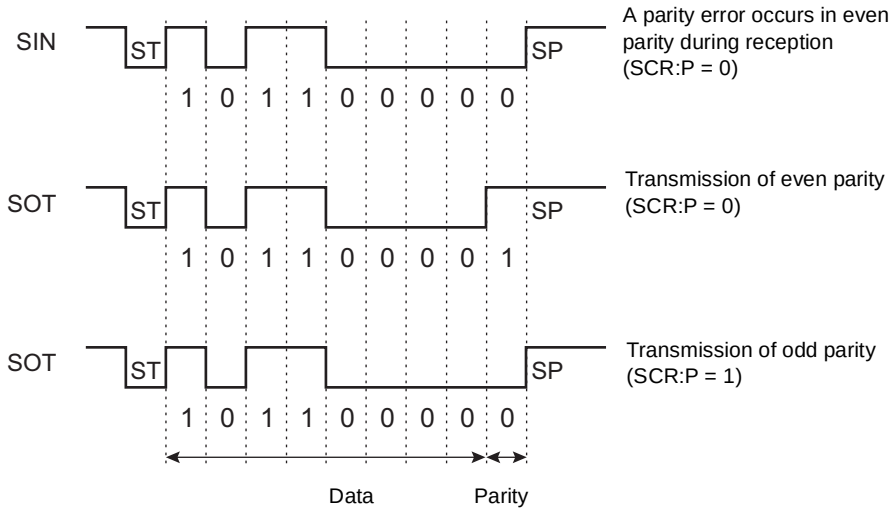
● Parity

The addition (at transmission) of and the detection (during reception) of a parity bit can be set.

The parity enable bit (SCR:PEN) is used to select whether or not to use a parity; the parity select bit (SCR:P) is used to select the odd/even parity.

In operating mode 1, the parity cannot be used.

Figure 16.7-2 Transmission Data when Parity is Enabled



ST: Start bit, SP: Stop bit, Parity used (PEN = 1)
Note: In operating mode 1, the parity cannot be used.

● Data signaling

NRZ data format.

● Data bit transfer method

The data bit transfer method can be LSB-first transfer or MSB-first transfer.

16.7.2 Operations in Synchronous Mode (Operating Mode 2)

When the LIN-UART is used in operating mode 2 (normal mode), the transfer method is clock-synchronous transfer.

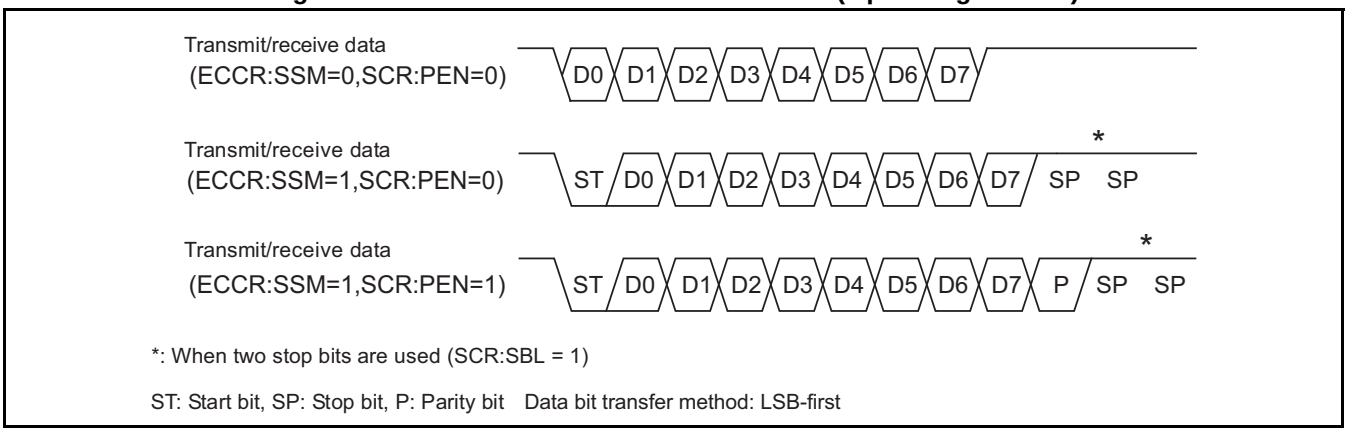
■ Operations in Synchronous Mode (Operating Mode 2)

● Transmit/receive data format

In synchronous mode, 8-bit data is transmitted and received; the addition of the start bit and of the stop bit can be selected (ECCR:SSM). When the start/stop bits are added to the data format (ECCR:SSM = 1), the addition of the parity bit can also be selected (SCR:PEN).

Figure 16.7-3 shows the data format in synchronous mode (operating mode 2).

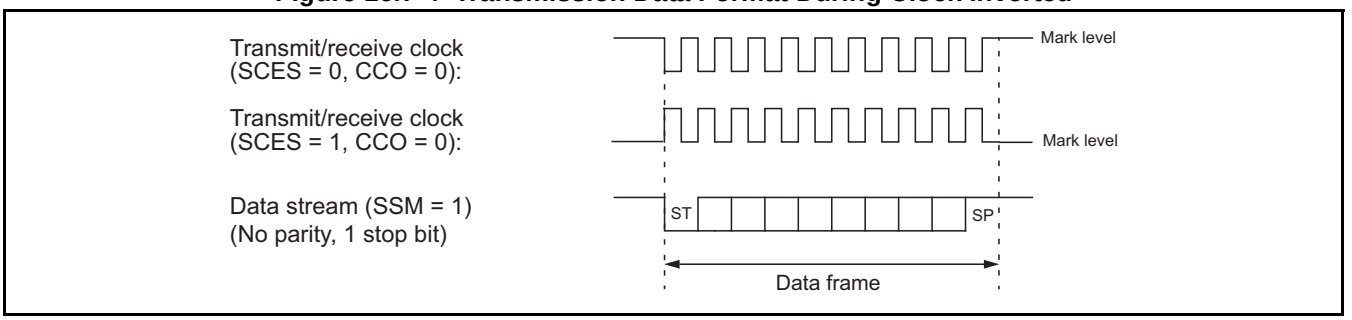
Figure 16.7-3 Transmit/Receive Data Format (Operating Mode 2)



● Clock inversion function

When the SCES bit in the LIN-UART extended status control register (ESCR) is "1", the serial clock is inverted. In the case of serial clock reception side is selected, the LIN-UART samples data at the falling edge of the received serial clock. In the case of serial clock transmission side is selected, the mark level is set to "0" when the SCES bit is "1".

Figure 16.7-4 Transmission Data Format During Clock Inverted



● Start/stop bits

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 they are in asynchronous mode.

● Clock supply

In clock synchronous mode (normal), the number of transmit/receive data bits must be equal to the number of clock cycles. When the start/stop bits are enabled, the number of clock cycles must be equal to the sum of the transmit/receive data bits and the added start/stop bits.

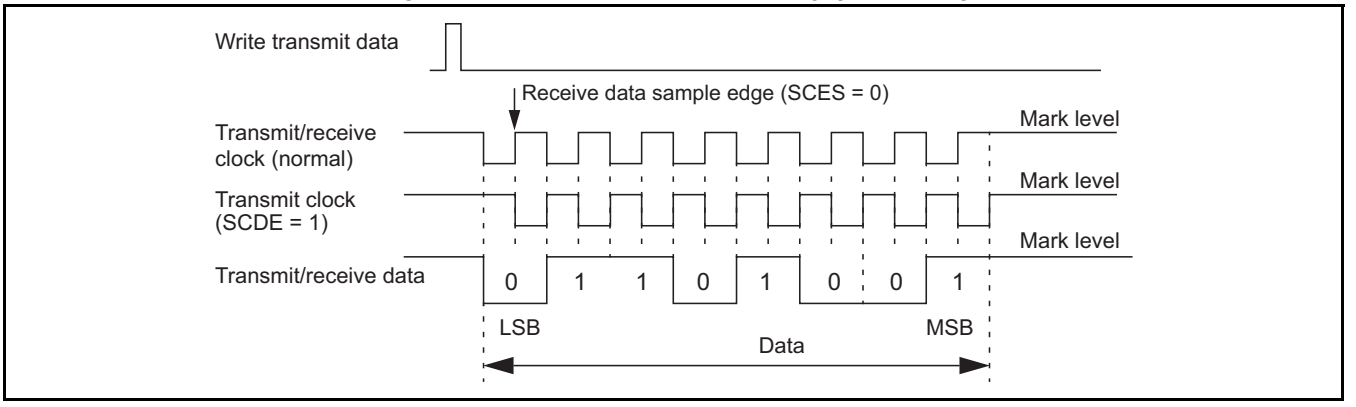
With the serial clock transmission side having been selected (ECCR:MS = 0), when the serial clock output is enabled (SMR:SCKE = 1), a synchronous clock is automatically output during transmission/reception. When the serial clock reception side (ECCR:MS = 1) is selected or the serial clock output is disabled (SMR:SCKE = 0), clock cycles equal to the number of transmit/receive data bits must be supplied from an external clock pin.

The clock signal must be kept at the mark level ("H") if serial data is not related to transmission/reception.

● Clock delay

When the SCDE bit in the ECCR is set to "1", a delayed transmit clock is output as shown in Figure 16.7-5. This function is required when the device on the reception side samples data at the rising edge or falling edge of the serial clock.

Figure 16.7-5 Transmit Clock Delay (SCDE = 1)



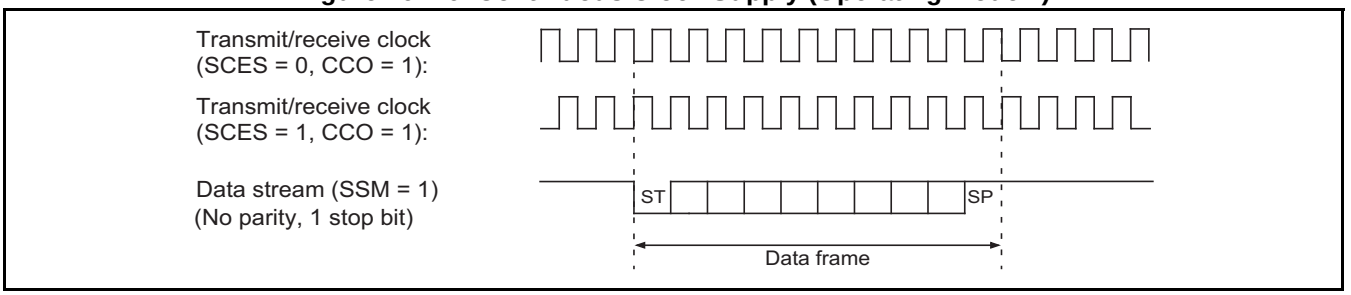
● Clock inversion

When the SCES bit in the LIN-UART extended status register (ESCR) is "1", the LIN-UART clock is inverted, and receive data is sampled at the falling edge of the LIN-UART clock. At that time, the value of the serial data must become valid at the edge of the LIN-UART clock.

● Continuous clock supply

When the CCO bit in the ESCR register is "1", the serial clock output from the SCK pin is continuously supplied on the serial clock transmission side. In this case, add the start bit and the stop bit to the data format (SSM = 1) in order to identify the beginning and end of the data frame. Figure 16.7-6 shows the operation of continuous clock supply (operating mode 2).

Figure 16.7-6 Continuous Clock Supply (Operating Mode 2)



● Error detection

When the start bit and the stop bit are disabled (ECCR:SSM = 0), only overrun errors are to be detected.

● Communication settings for synchronous mode

To perform communications in synchronous mode, the following settings are required.

- LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0)
  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"– Uses the dedicated baud rate reload counter
  : "0"– Inputs an external clock
  SOE : "1"– Enables transmission/reception
  : "0"– Enables only reception
- LIN-UART serial control register (SCR)
  RXE, TXE: Set either bit to "1".
  AD : Since the address/data format selection function is not used, the value of this bit has no effect on operation.
  CL : Since the bit length is automatically set to 8 bits, the value of this bit has no effect on operation.
  CRE : "1": Since the error flag is cleared, transmission/reception is suspended.
  - For SSM = 0:

    PEN, P, SBL: Since neither the parity bit nor the stop bit is used, the values of these three bits have no effect on operation.
  - For SSM = 1:

    PEN : "1": Adds/detects parity bit,"0": Not use parity bit
    P : "1": Even parity, "0": Odd 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"– Enables receive interrupts, "0"– Disables receive interrupts

  TIE : "1"– Enables transmit interrupts, "0"– Disables transmit interrupts
- LIN-UART extended communication control register (ECCR)

  SSM : "0"– Not use start/stop bits (normal),

  "1"– Uses start/stop bits (extended function),

  MS : "0"– Serial clock transmission side of serial clock (serial clock output),

  "1"– Serial clock reception side (inputs serial clock from the device on the serial clock transmission side)

Note:

To start communication, write data to the LIN-UART transmit data register (TDR).

To receive data only, disable the serial output (SMR:SOE = 0), and then write dummy data to the TDR register.

Enabling continuous clock output and the start/stop bits enables bi-directional communication as that in asynchronous mode.

16.7.3 Operations of LIN function (Operating Mode 3)

In operating mode 3, the LIN-UART works as the LIN master and the LIN slave. In operating mode 3, the communication format is set to 8-bit data, no parity, stop bit 1, 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 with the master.

Writing "1" to the LBR bit in the LIN-UART extended communication control register (ECCR) outputs 13 bits to 16 bits at the "L" level from the SOT pin. These bits are the LIN synch break indicating the beginning of a LIN message.

The TDRE flag bit in the LIN-UART serial status register (SSR) is then set to "0". After the LIN synch break, the TDRE flag bit 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 LBL 0/1 bits in ESCR as shown in the following table.

Table 16.7-3 LIN Synch Break Length

LBL0	LBL1	Synch break length
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

A LIN synch field is transmitted as byte data 0x55 following a LIN synch break. To prevent the generation of a transmit interrupt, 0x55 can be written to the TDR after the LBR bit in ECCR is set to "1" even if the TDRE flag bit is "0".

● Operation as LIN slave

In LIN slave mode, the LIN-UART must synchronize with the baud rate of the master. The LIN-UART generates a receive interrupt when LIN break interrupt is enabled (LBIE = 1) even though reception has been disabled (RXE = 0). The LBD bit in ESCR is set to "1" as a receive interrupt is generated.

Writing "0" to the LBD bit clears the receive interrupt request flag.

The calculation of baud rate is illustrated below using the operation of the LIN-UART as an example. When the LIN-UART detects the first falling edge of the synch field, set the internal signal to be input to the 8/16-bit composite timer to "H", and then start the 8/16-bit composite timer. The internal signal becomes "L" at the fifth falling edge. The 8/16-bit composite timer must be set to the input capture mode. In addition, the 8/16-bit composite timer interrupt must be enabled and the 8/16-bit composite timer must be set to detect both edges. The time at which the input signal input to the 8/16-bit composite timer is eight times the baud rate.

The baud rate setting can be found by the following equations.

When the counter of the 8/16-bit composite timer does not overflow
: BGR value = (b - a) / 8 - 1

When the counter of the 8/16-bit composite timer has overflowed
: 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:

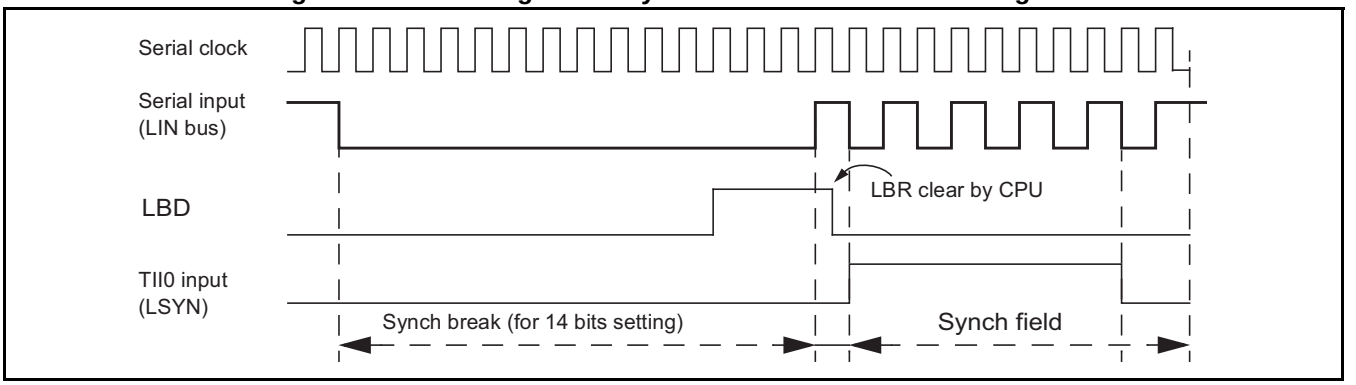
If the BGR value newly calculated based on the synch field in LIN slave mode as explained above has an error of ±15% or more, do not set the baud rate.

For the operations of the input capture function of the 8/16-bit composite timer, see "14.13 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 16.7-7 Timing of LIN Synch Break Detection and Flag Set



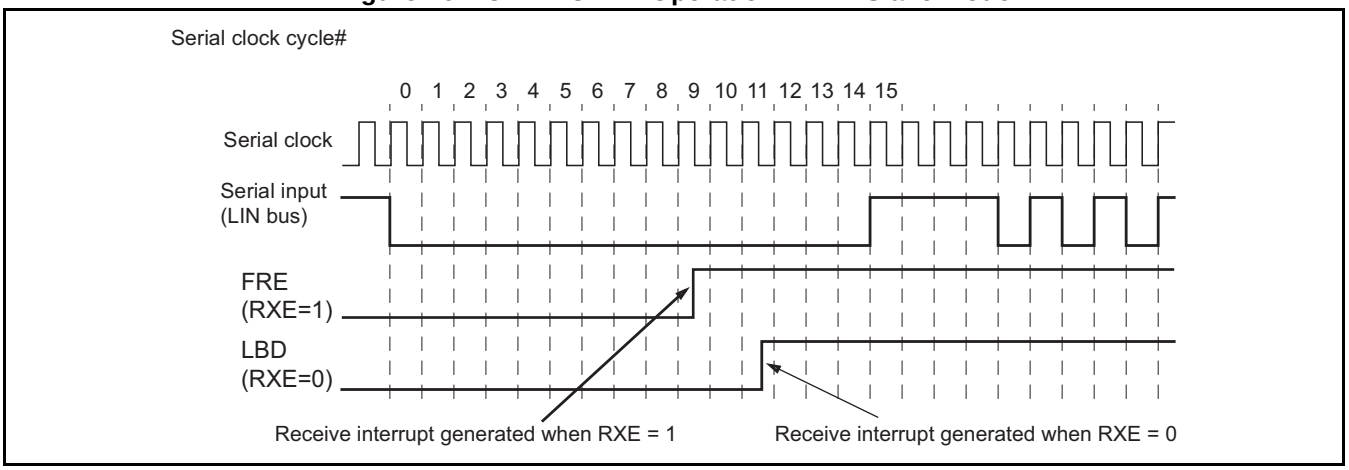
The above diagram shows the timing of the LIN synch break detection and flag.

Since the data framing error (FRE) flag bit in SSR generates a receive interrupt two bits earlier than a LIN break interrupt (if the following communication format is used: 8-bit data, no parity, one stop bit.), set the RXE to "0" when using the LIN break.

The LIN synch break detection functions only in operating mode 3.

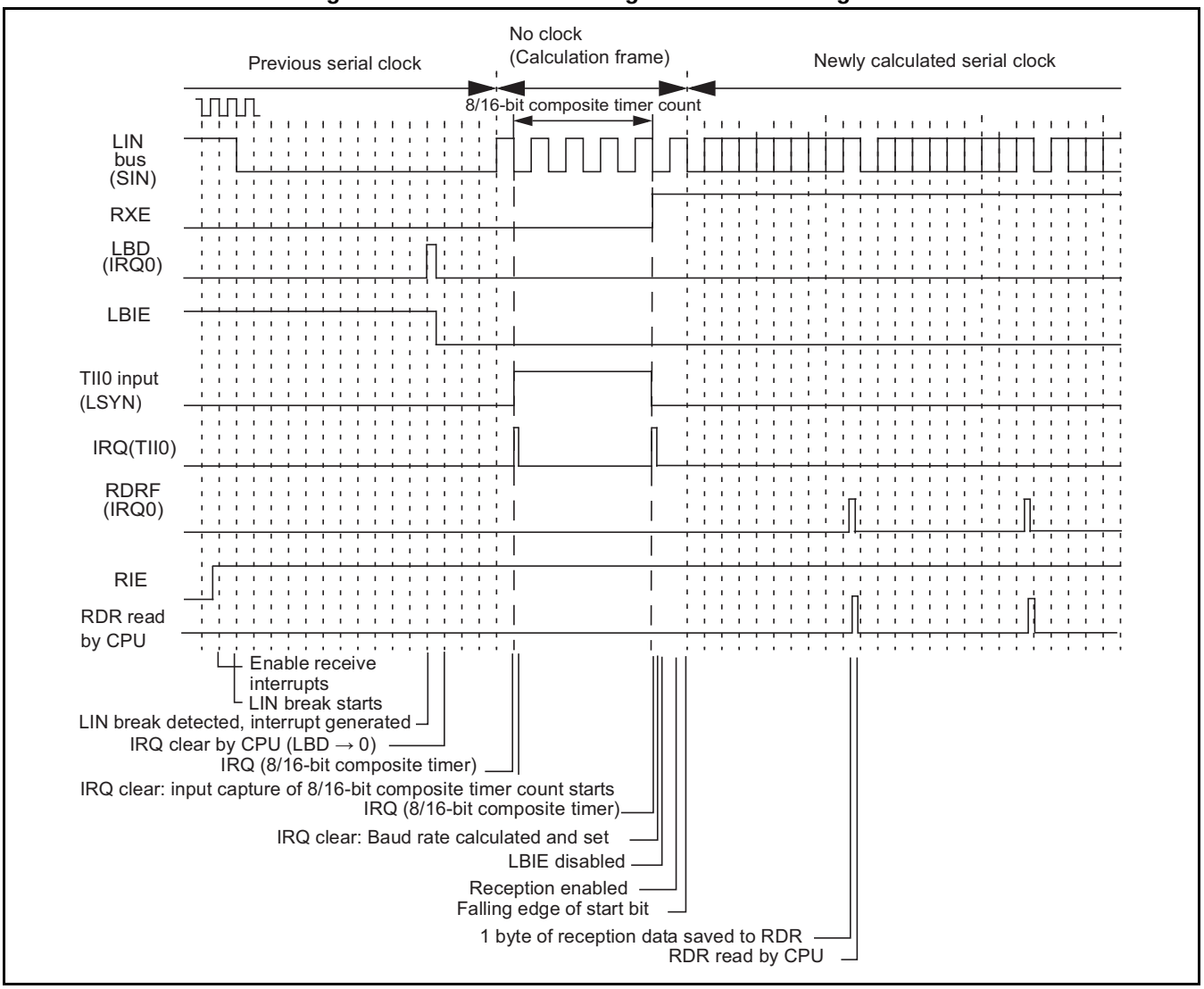
Figure 16.7-8 shows the LIN-UART operation in LIN slave mode.

Figure 16.7-8 LIN-UART Operation in LIN Slave Mode



● LIN bus timing

Figure 16.7-9 LIN Bus Timing and LIN-UART Signals



16.7.4 Serial Pin Direct Access

The transmit pin (SOT) and the receive 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).

To freely set the value of the serial output pin (SOT), enable the direct write access to the serial output pin (SOT) (ESCR:SOPE = 1), write "0" or "1" to the serial I/O pin direct access bit (ESCR:SIOP), and then enable serial output (SMR:SOE = 1).

In LIN mode, this feature is used for reading transmitted data and for error handling when there is a physical LIN bus line signal error.

Note:

Direct access is allowed only when transmission is not in progress (the transmit 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 from the SIOP bit by the read-modify-write (RMW) type of instruction.

16.7.5 Bidirectional Communication Function (Normal Mode)

Normal serial bidirectional communication can be performed in operating mode 0 or 2. Asynchronous mode can be selected in operating mode 0 and synchronous mode in operating mode 2.

■ Bidirectional Communication Function

To operate the LIN-UART in normal mode (operating mode 0 or 2), the settings shown in Figure 16.7-10 are required.

Figure 16.7-10 Settings of LIN-UART Operating 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	A/D	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	◎	◎

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7 to bit0
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)
Mode 0 →	◎	◎	◎	◎	◎	◎	◎	◎	
Mode 2 →	□	◎	□	◎	◎	◎	◎	◎	

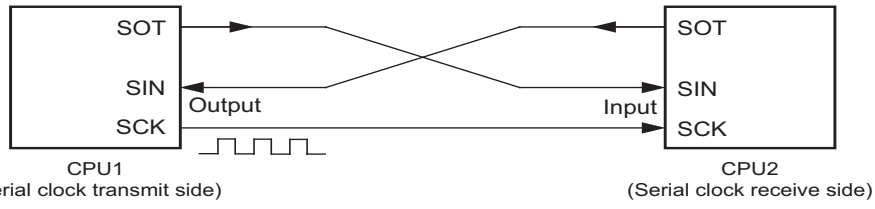
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 0 →	×	×	×	×	◎	◎	0	0	0	0	×	×	×	0	◎	◎
Mode 2 →	×	×	×	×	◎	◎	□	◎	0	×	◎	◎	◎	0	□	□

◎ : Bit to be used
× : 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

When using bidirectional communication, connect two CPUs as shown in Figure 16.7-11.

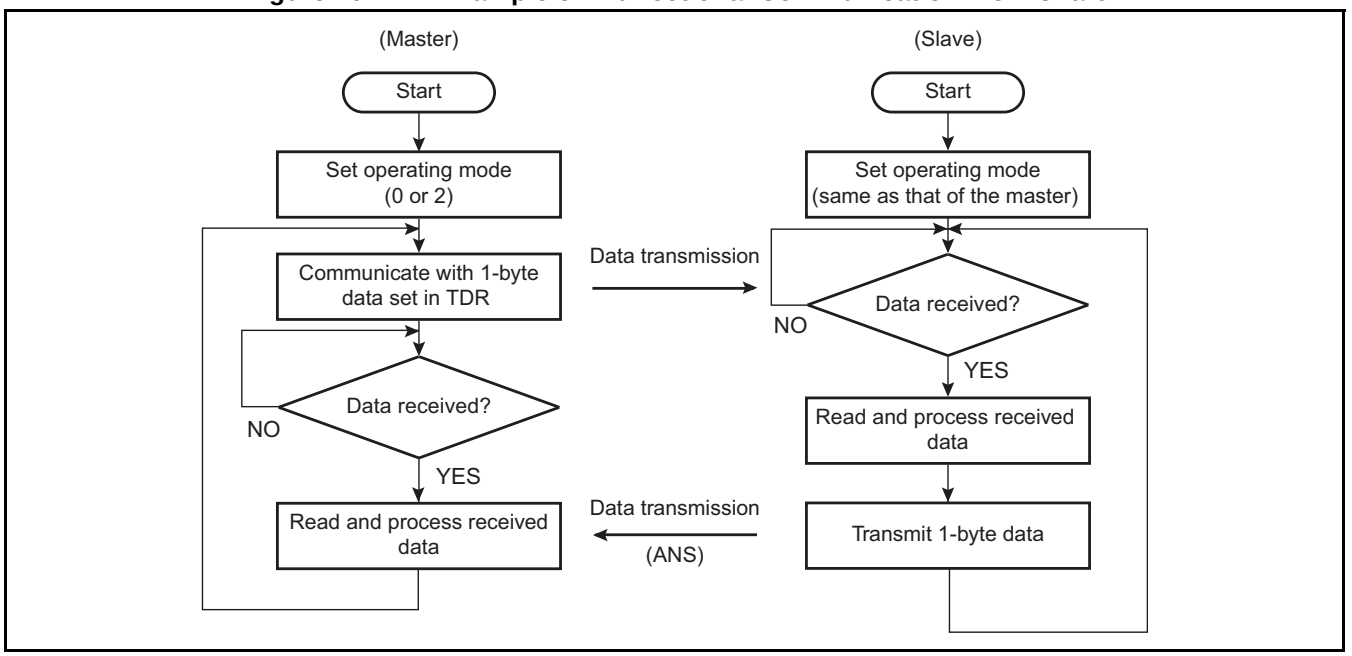
Figure 16.7-11 Example of Connection for Bidirectional Communication in LIN-UART Mode 2



● Communication procedure example

The communication starts from the transmit side at any time after transmit data is ready. The receive side returns ANS (per one byte in this example) regularly after receiving transmit data. Figure 16.7-12 is an example of bidirectional communication flow chart.

Figure 16.7-12 Example of Bidirectional Communication Flow Chart



16.7.6 Master/Slave Mode Communication Function (Multiprocessor Mode)

Operating mode 1 allows communication among multiple CPUs connected in master/slave mode. The LIN-UART can be used as a master or a slave.

■ Master/Slave Mode Communication Function

To operate the LIN-UART in multiprocessor mode (operating mode 1), the settings shown in Figure 16.7-13 are required.

Figure 16.7-13 Settings of LIN-UART Operating 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, RDR1/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set compare data (during writing) Retain receive data (during reading)
Mode 1 →	×	◎	◎	◎	◎	◎	◎	◎	

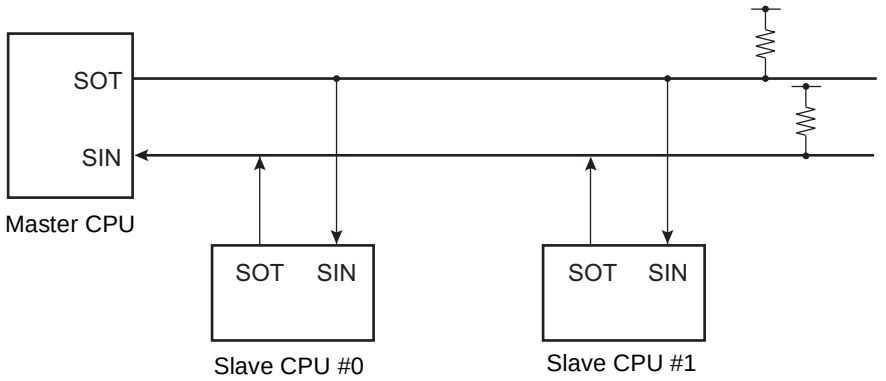
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 1 →	×	×	×	×	◎	◎	0	0	0	×	×	×	×	0	◎	◎

◎ : Bit to be used
× : 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 consists of two common communication lines connecting between one master CPU and multiple slave CPUs as shown in Figure 16.7-14. The LIN-UART can be used as a master or a slave.

Figure 16.7-14 Connection Example of LIN-UART Master/Slave Mode Communication



● Function selection

In master/slave mode communication, select the operating mode and the data transfer method as shown in Figure 16.7-14.

Table 16.7-4 Selection of Master/Slave Mode Communication Functions

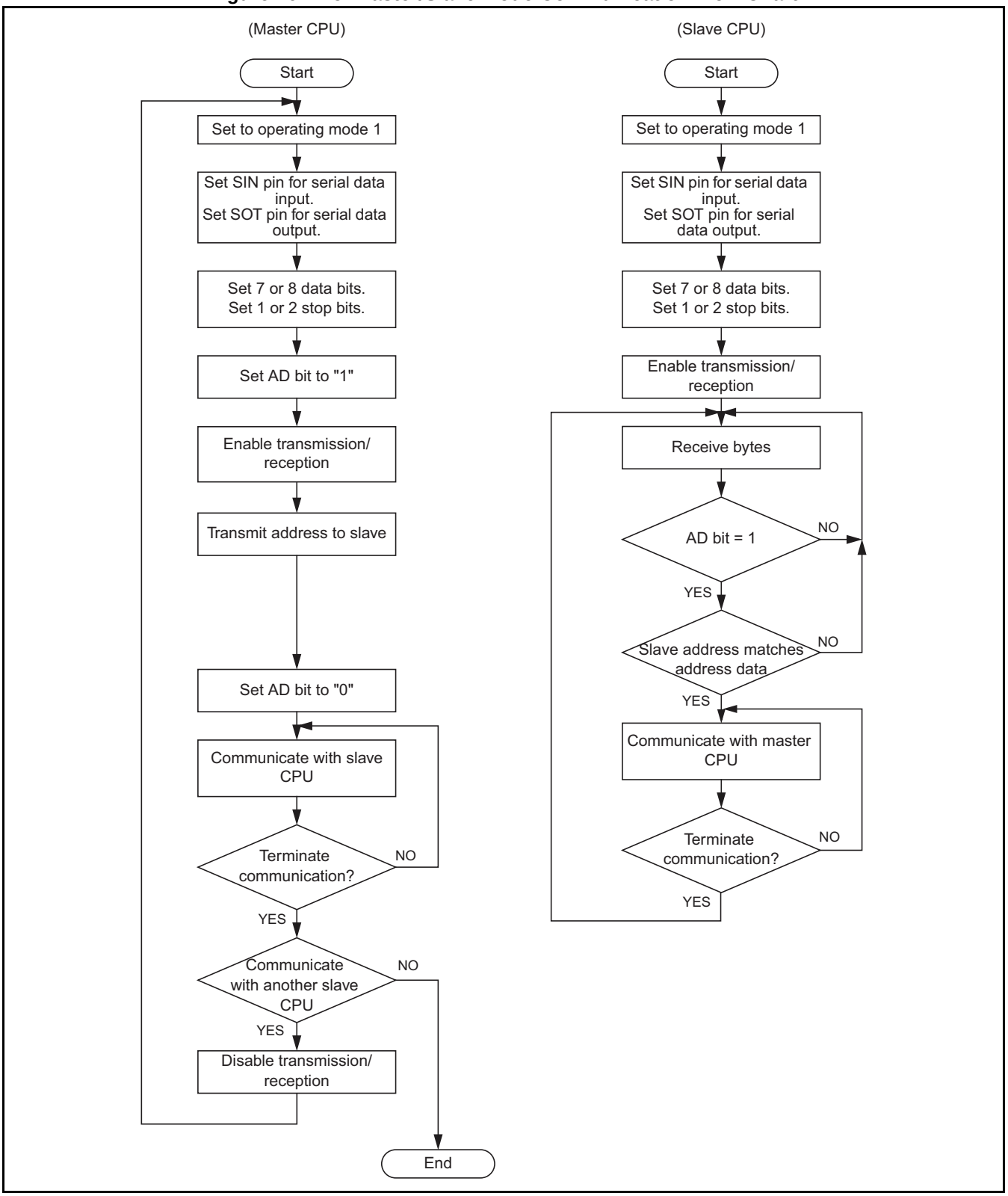
	Operating mode		Data	Parity	Synchronous method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission/ reception	Mode 1 (Transmit/ receive A/D bit)	Mode 1 (Transmit/ receive A/D bit)	AD = 1 + 7-bit or 8-bit address	None	Asynchronous	1 bit or 2 bits	LSB first or MSB first
Data transmission/ reception			AD = 0 + 7-bit or 8-bit data				

● Communication procedure

Master/slave mode communication starts as the master CPU transmits address data. The address data, which is the data chosen when the AD bit is set to "1", determines the slave CPU that is to be the destination of the communication. A slave CPU uses a program to check address data, and communicates with the master CPU when the address data matches the address assigned to that slave CPU.

Figure 16.7-15 is a flow chart showing master/slave mode communication (multiprocessor mode).

Figure 16.7-15 Master/Slave Mode Communication Flow Chart



16.7.7 LIN Communication Function

In LIN-UART communication, a LIN device can be used in a LIN master system or a LIN slave system.

■ LIN Master/Slave Mode Communication Function

Figure 16.7-16 shows the required settings for the LIN communication mode (operating mode 3) of the LIN-UART.

Figure 16.7-16 Settings of LIN-UART Operating 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	◎	◎

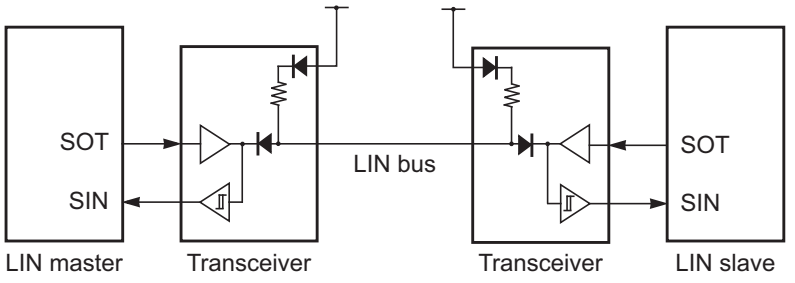
◎ : Bit to be used
× : Unused bit
1 : Set to "1"
0 : Set to "0"
+ : Bit correctly set automatically

● LIN device connection

Figure 16.7-17 shows an example of communication in a LIN bus system.

The LIN-UART can operate as a LIN master or a LIN slave.

Figure 16.7-17 Example of LIN Bus System Communication

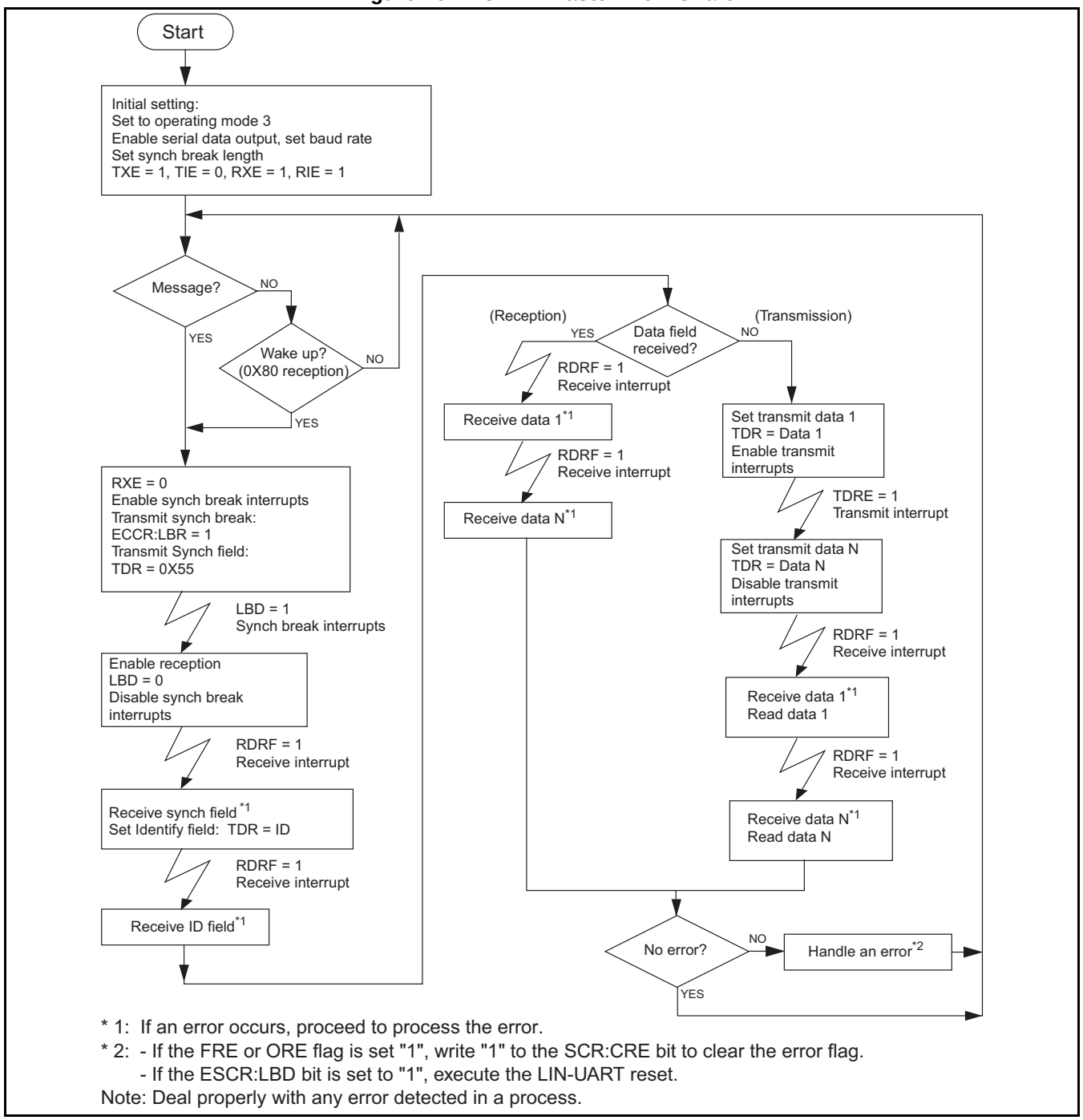


16.7.8 Examples of LIN-UART LIN Communication Flow Chart (Operating Mode 3)

This section shows examples of LIN-UART LIN communication flow charts.

■ LIN Master Device

Figure 16.7-18 LIN Master Flow Chart

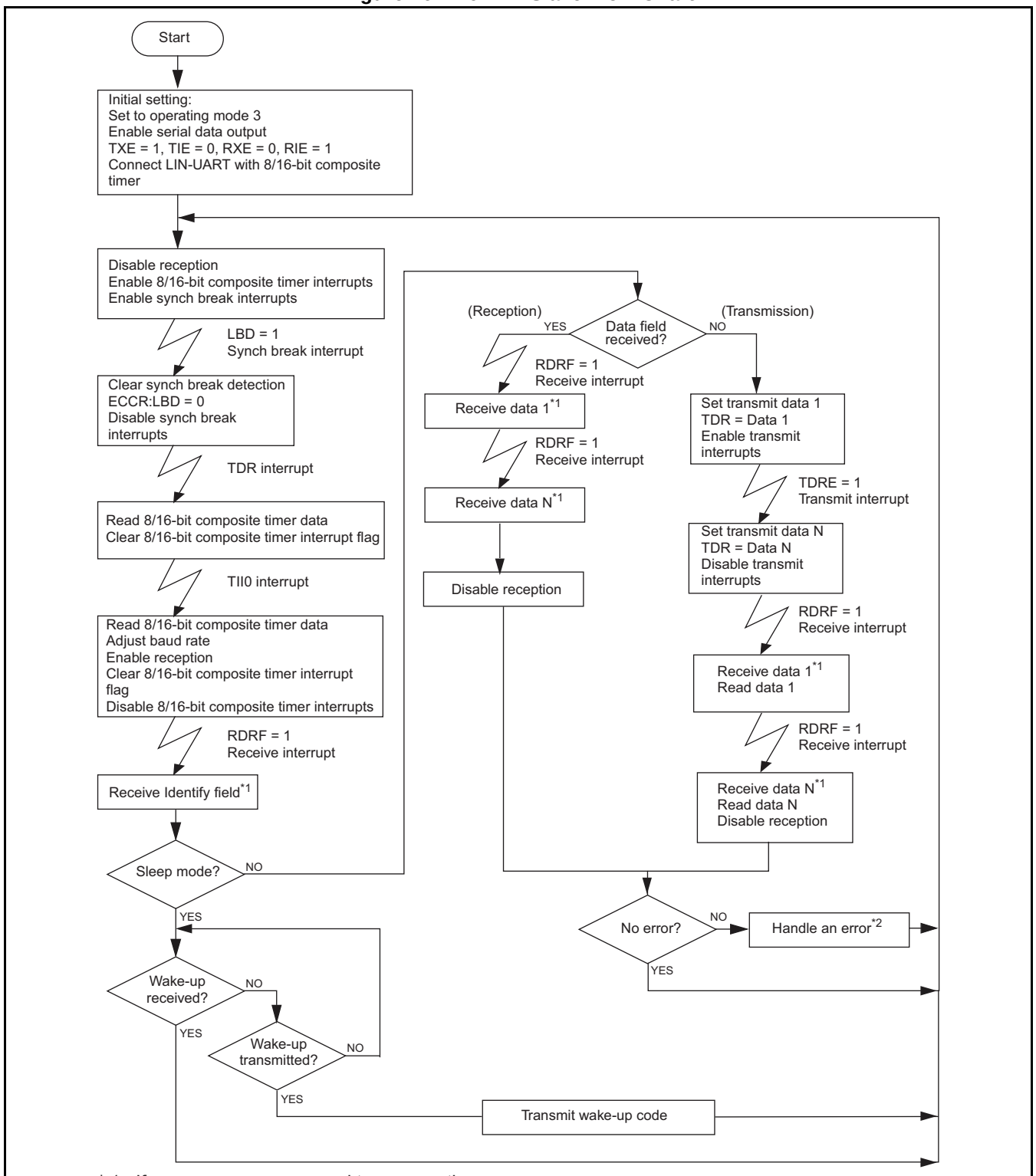


* 1: If an error occurs, proceed to process the error.
* 2: - If the FRE or ORE flag is set "1", write "1" to the SCR:CRE bit to clear the error flag.
- If the ESCR:LBD bit is set to "1", execute the LIN-UART reset.

Note: Deal properly with any error detected in a process.

■ LIN Slave Device

Figure 16.7-19 LIN Slave Flow Chart



* 1: If an error occurs, proceed to process the error.
* 2: - If the FRE or ORE flag is set "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: Deal properly with any error detected in a process.

16.8 Notes on Using LIN-UART

This section provides notes on using the LIN-UART.

■ Notes on Using LIN-UART

● Enabling operation

The LIN-UART has the TXE bit and the RXE bit in the LIN-UART serial control register (SCR) to enable transmission and reception respectively. Since both transmission and reception are disabled by default (initial values), they must be enabled before the transfer starts. Transmission and reception can be disabled to stop transfer if necessary.

● Setting communication mode

The communication mode should be set while the LIN-UART stops operating. If the communication mode is set while transmission or reception is in progress, the integrity of data being transmitted or received at the setting of the mode 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 made immediately after the transmit interrupt request is enabled (SSR:TIE = 1). To prevent any transmit interrupt request from being made, always set the TIE flag bit to "1" after setting transmit data.

● Modifying operation settings

After modifying operation settings such as the addition of start/stop and changing the data format, reset the LIN-UART.

Even though the setting of the LIN-UART serial mode register (SMR) and the resetting of the LIN-UART (SMR:UPCL = 1) are executed simultaneously, that does not ensure that the operation settings are correct. Therefore, after setting the LIN-UART serial mode register (SMR), reset the LIN-UART again.

● Using LIN functions

The LIN functions are available in operating mode 3. In the same mode, the communication format is predefined (8-bit data, no parity, one stop bit, LSB first).

While the length of the LIN synch break transmit bit is variable, in detection, the bit length is fixed at 11 bits.

● LIN slave settings

Before the LIN-UART starts operating as a slave, the baud rate must be set before the first LIN synch break is received to ensure that a LIN synch break whose length is a minimum of 13 bits is successfully detected.

● Bus idle function

The bus idle function is not available in synchronous mode (operating mode 2).

- AD bit (LIN-UART serial control register (SCR): Address/data format select bit)

  Pay attention to the following issues when using the AD bit.

  The AD bit is used to select the address/data for transmission by writing a value to it. When the AD bit is read, it returns the value of the AD bit received last. Inside the microcontroller, the AD bit value received and the one transmitted are saved in separate registers.

  The AD bit value transmitted is read when the read-modify-write (RMW) type of instruction is used. Therefore, if another bit in the SCR register is accessed by bit access, an incorrect value may be written to the AD bit.

  For the above reason, the AD bit must be set by the last access to the SCR register before transmission. The above problem can also be prevented by always using byte access to write values to the SCR register.

- 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 operating mode 3 (LIN mode), when serial input is 11 bits or more in width and becomes "L", the LBD bit in the extended status control register (ESCR) is set to "1" (synch break detected) and the LIN-UART waits for the synch field. Therefore, when serial input has more than 11 bits of "0" not at the time of a synch break, the LIN-UART recognizes that a synch break has been input (LBD = 1) and then waits for the synch field.

  In this case, execute the LIN-UART reset (SMR: UPCL = 1).

16.9 Sample Programs for LIN-UART

Fujitsu Microelectronics provides sample programs for operating the LIN-UART.

■ Sample Programs for LIN-UART

For the sample programs for the LIN-UART, see "■ Sample Programs" in "PREFACE".

■ Setting Methods Other than Sample Programs

● Method of selecting an operating mode

Use the operating mode select bits (SMR:MD[1:0]).

Operating mode		Operating mode select bits (MD[1:0]).
Mode 0	Asynchronous (Normal mode)	Set the bits to "00_B".
Mode 1	Asynchronous (Multiprocessor mode)	Set the bits to "01_B".
Mode 2	Synchronous (Normal mode)	Set the bits to "10_B".
Mode 3	Asynchronous (LIN mode)	Set the bits to "11_B".

● Types of operating clock and method of selecting an operating clock

Use the external clock select bit (SMR:EXT).

Clock input	External clock select bit (EXT)
To select a dedicated baud rate generator	Set the bit to "0".
To select an external clock	Set the bit to "1".

● Method of controlling the SCK, SIN, and SOT pins

Use the following settings.

	LIN-UART
To set the SCK pin as an input pin	DDR0:P02 = 0 SMR:SCKE = 0
To set the SCK pin as an output pin	SMR:SCKE = 1
To use the SIN pin	DDR0:P04 = 0
To use the SOT pin	SMR:SOE = 1

● Method of enabling/disabling the LIN-UART operation

Use the receive operation enable bit (SCR:RXE).

Control item	Receive operation enable bit (RXE)
To disable reception	Set the bit to "0".
To enable reception	Set the bit to "1".

Use the transmit operation control bit (SCR:TXE).

Control item	Transmit operation control bit (TXE)
To disable transmission	Set the bit to "0".
To enable transmission	Set the bit to "1".

● Method of using an external clock as the serial clock of the LIN-UART

Use the one-to-one external clock input enable bit (SMR:OTO).

Control item	One-to-one external clock input enable bit (OTO)
To enable external clock	Set the bit to "1".

● Method of restarting the reload counter

Use the reload counter restart bit (SMR:REST).

Control item	Reload counter restart bit (REST)
To restart the reload counter	Set the bit to "1".

● Restarting the LIN-UART

Use the LIN-UART programmable clear bit (SMR:UPCL).

Control item	LIN-UART programmable clear bit (UPCL)
To reset the LIN-UART with software reset	Set the bit to "1".

● Method of setting the parity

Use the parity enable bit (SCR:PEN) and the parity select bit (SCR:P).

Operation	Parity control (PEN)	Parity polarity (P)
To use no parity	Set the bit to "0".	-
To use the even parity	Set the bit to "1".	Set the bit to "0".
To use the odd parity	Set the bit to "1".	Set the bit to "1".

● Method of setting the data length

Use the data length select bit (SCR:CL).

Operation	Data length select bit (CL)
To set the bit length to 7 bits	Set the bit to "0".
To set the bit length to 8 bits	Set the bit to "1".

● Method of selecting the stop bit length

Use the stop bit length select bit (SCR:SBL).

Operation	Stop bit length select bit (SBL)
To set the stop bit length to 1	Set the bit to "0".
To set the stop bit length to 2	Set the bit to "1".

● Method of clearing the error flag

Use the receive error flag clear bit (SCR:CRE).

Control item	Receive error flag clear bit (CRE)
To clear the error flag (PE, ORE, FRE)	Set the bit to "0".

● Method of setting the transfer direction

Use the transfer direction select bit (SSR:BDS).

In all operating modes, the transfer direction can be selected from LSB-first and MSB-first.

Control item	Transfer direction select bit (BDS)
To select the LSB-first (from the least significant bit)	Set the bit to "0".
To select the MSB-first (from the most significant bit)	Set to the bit "1".

● Method of clearing the receive completion flag

Use the following method.

Control item	Method
To clear the receive completion flag	Read the RDR register.

Reception starts at the first time the RDR register is read.

● Method of clearing the transmit buffer empty flag

Use the following method.

Control item	Method
To clear the transmit buffer empty flag	Write data to the TDR register.

Transmission starts at the first time data is written to the TDR register.

● Method of selecting the data format (address/data) (only in mode 1)

Use the address/data format select bit (SCR:AD).

Operation	Address/data format select bit (A/D)
To select the data frame	Set the bit to "0".
To select the address frame	Set the bit to "1".

The setting is effective only in transmission. The AD bit is ignored in reception.

● Method of setting the baud rate

See "16.6 LIN-UART Baud Rate".

● Interrupt-related registers

Interrupt level is set by interrupt level setting registers as shown in the following table.

	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: $0FFEA_H$

● Method of enabling/disabling/clearing interrupts

Interrupt request enable flag, interrupt request flag

Use the interrupt request enable bits (SSR:RIE), (SSR:TIE) enable respective interrupts.

	UART reception	UART transmission
	Receive interrupt enable bit (RIE)	Transmit interrupt enable bit (TIE)
To disable interrupt requests	Set the bit to "0"	
To enable interrupt requests	Set the bit to "1"	

Use the following setting to clear interrupt requests.

	UART reception	UART transmission
To clear interrupt requests	The receive data register full flag bit (RDRF) is cleared by reading the LIN-UART serial input register (RDR).	The transmit data register empty flag bit (TDRE) is set to "0" by writing data to the LIN-UART serial output data register (TDR).
	The error flag (PE, ORE or FRE) is set to "0" by writing "1" to the error flag clear bit (CRE).	

CHAPTER 17

8/10-BIT A/D CONVERTER

This chapter describes the functions and operations of the 8/10-bit A/D converter.

17.1 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 by the software and internal clock, with one input signal selected from multiple analog input pins.

■ A/D Conversion Function

The A/D converter converts analog voltage (input voltage) input through an analog input pin to a 10-bit digital value.

- The input signal can be selected from multiple analog input pins.
- The conversion speed can be set in a program. (can be selected according to operating voltage and frequency).
- An interrupt is generated when A/D conversion is completed.
- The completion of conversion can be determined according to the ADI bit in the ADC1 register.

To activate the A/D conversion function, use one of the following methods.

- Activation using the ADI bit in the ADC1 register
- Continuous activation using the 8/16-bit composite timer output TO00

17.2 Configuration of 8/10-bit A/D Converter

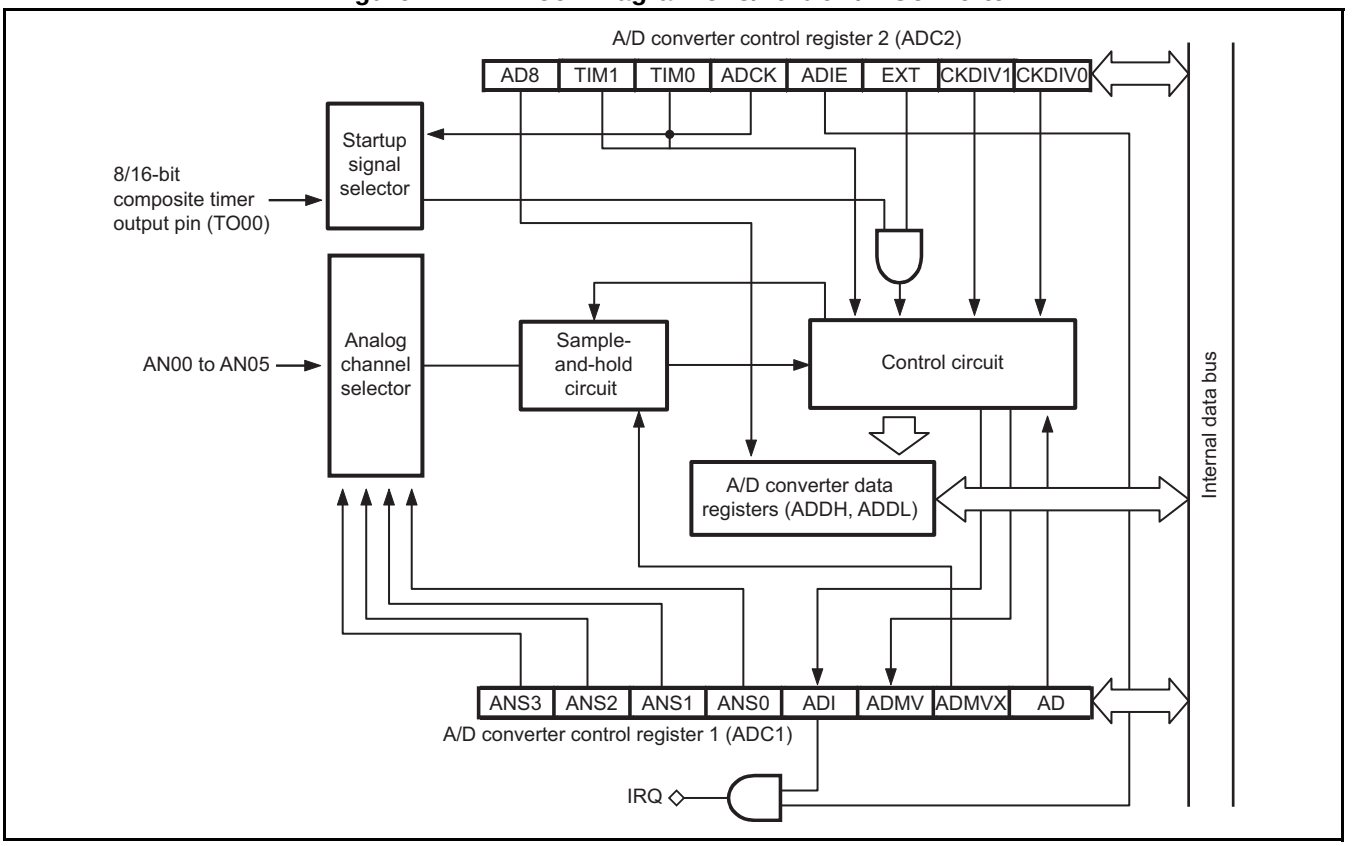
The 8/10-bit A/D converter consists of the following blocks:

- **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 17.2-1 is the block diagram of the 8/10-bit A/D converter.

Figure 17.2-1 Block Diagram of 8/10-bit A/D Converter



- Clock selector

  This selects the A/D conversion clock with continuous activation having been enabled (ADC2:EXT = 1).

- Analog channel selector

  This is the circuit selecting an input channel from several analog input pins.

- Sample-and-hold circuit

  This circuit holds input voltage selected by the analog channel selector. By sampling the input voltage and holding it immediately after A/D conversion starts, this circuit prevents A/D conversion from being affected by the fluctuation in input voltage during the conversion (comparison).

- Control circuit

  The A/D conversion function determines the values in the 10-bit A/D data register sequentially from MSB to LSB based on the voltage compare signal from the comparator. When A/D conversion is completed, the A/D conversion function sets the interrupt request flag bit (ADC1: ADI) to "1".

- A/D converter data registers (ADDH/ADDL)

  The upper two bits of 10-bit A/D data are stored in the ADDH register; the lower eight bits in the ADDL register.

  If the A/D conversion precision bit (ADC2:AD8) is set to "1", the A/D conversion precision becomes 8-bit precision, and the upper eight bits of 10-bit A/D data are to be stored in the ADDL register.

- A/D converter control register 1 (ADC1)

  This register is used to enable and disable different functions, select an analog input pin, and check the status of the A/D converter.

- A/D converter control register 2 (ADC2)

  This register is used to select an input clock, enable and disable interrupts and controls different A/D conversion functions.

■ Input Clock

The 8/10-bit A/D converter uses an output clock from the prescaler as the input clock (operating clock).

17.3 Pins Related to 8/10-bit A/D Converter

This section describes the pins related to the 8/10-bit A/D converter.

■ Pins Related to 8/10-bit A/D Converter

The MB95200H Series has six channels of analog input pin.

The analog input pins are also used as general-purpose I/O ports.

- AN05 pin to AN00 pin

AN05 to AN00: When using the A/D conversion function, input to one of these pins the analog voltage to be converted. A pin of AN05 to AN00 functions as an analog input pin if the bit in the port direction register (DDR) corresponding to that pin is set to "0" and the analog input pin select bits (ADC1:ANS0 to ANS3) are set to the values representing that pin. A pin not used as an analog input pin can be used as a general-purpose I/O port also when the 8/10-bit A/D converter is used.

The MB95210H Series has two channels of analog input pin.

The analog input pins are also used as general-purpose I/O ports.

- AN05 pin to AN04 pin

AN05 to AN04: When using the A/D conversion function, input to one of these pins the analog voltage to be converted. A pin of AN05 and AN04 functions as an analog input pin if the bit in the port direction register (DDR) corresponding to that pin is set to "0" and the analog input pin select bits (ADC1:ANS0 to ANS3) are set to the values representing that pin. A pin not used as an analog input pin can be used as a general-purpose I/O port also when the 8/10-bit A/D converter is used.

■ Block Diagram of Pins of 8/10-bit A/D Converter

Figure 17.3-1 Block Diagram of Pins AN00, AN01(P00/AN00, P01/AN01) of 8/10-bit A/D Converter

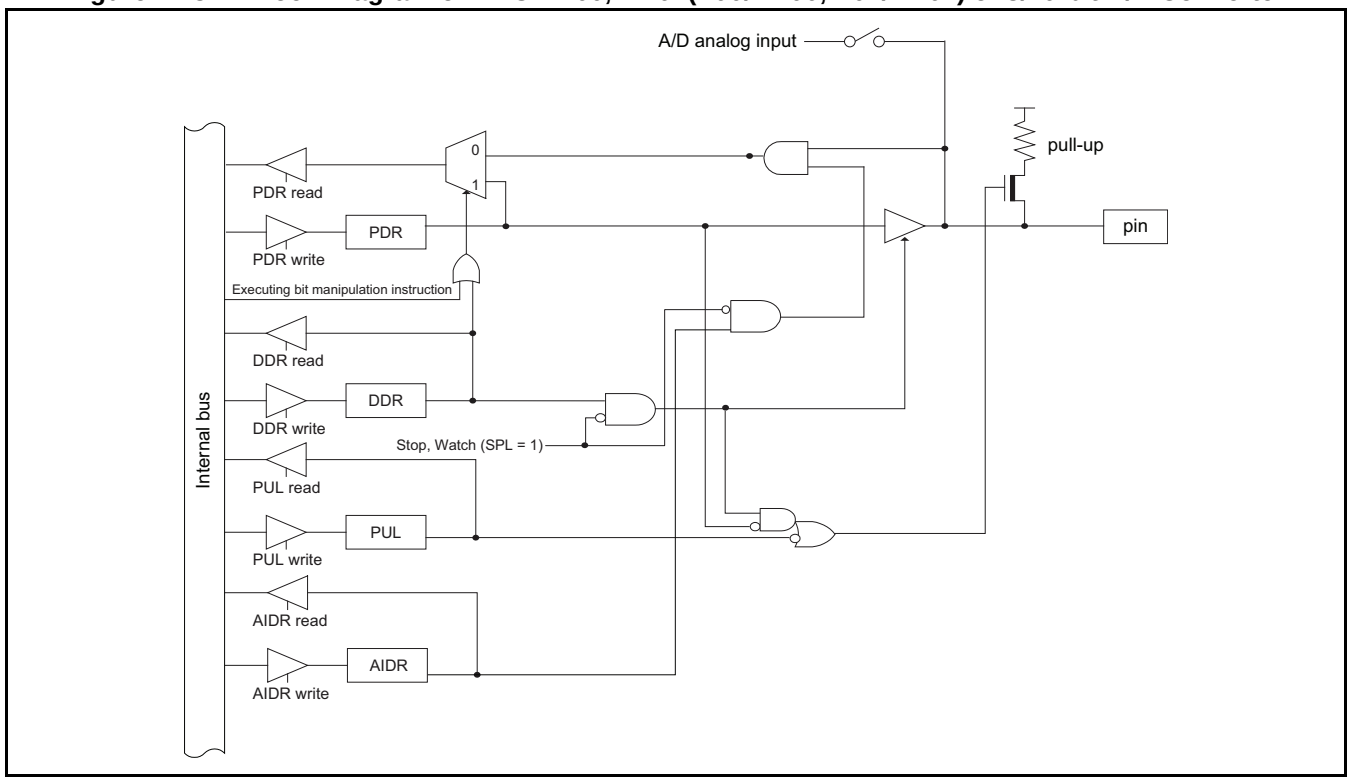


Figure 17.3-2 Block Diagram of Pins AN02, AN03, AN05(P02/INT02/AN02/SCK, P03/INT03/AN03/SOT, P05/INT05/AN05/TO00/HCLK2) Related to 8/10-bit A/D Converter

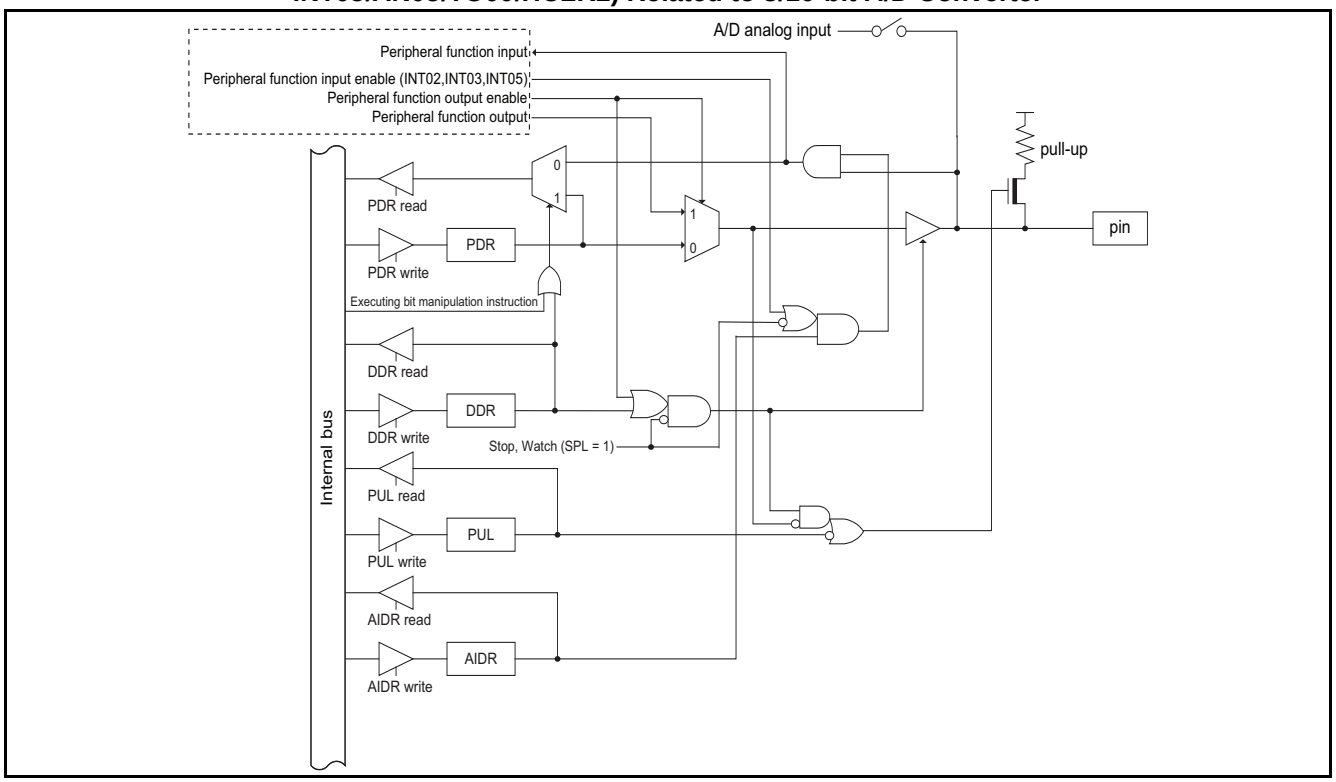
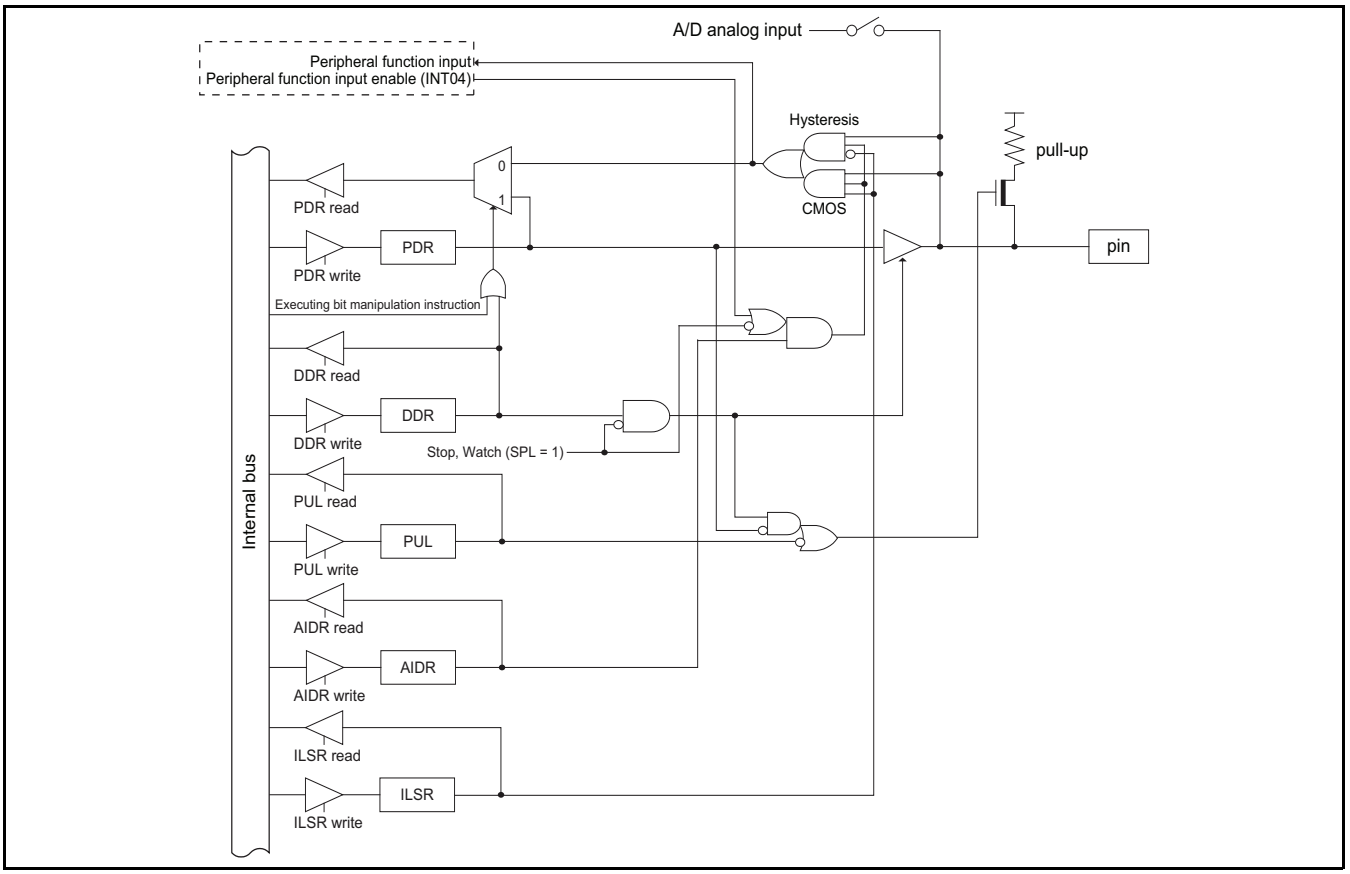


Figure 17.3-3 Block Diagram of Pin AN04 (P04/INT04/AN04/SIN/HCLK1/EC0) Related to 8/10-bit A/D Converter



17.4 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).

■ Registers of 8/10-bit A/D Converter

Figure 17.4-1 lists the registers of the 8/10-bit A/D converter.

Figure 17.4-1 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 (The read value is the same as the write value.)
R0, W:	Write only (Writable. "0" is read.)
R/WX:	Read only (Readable. Writing a value to it has no effect on operation.)
R0/WX:	Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R(RM1), W:	Readable/writable (The read value is different from write value. "1" is read by the read-modify-write (RMW) type of instruction.)

17.4.1 8/10-bit A/D Converter Control Register 1 (ADC1)

The 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 check the status of the converter.

■ 8/10-bit A/D Converter Control Register 1 (ADC1)

Figure 17.4-2 8/10-bit A/D Converter Control Register 1 (ADC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006CH	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD	00000000B
	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 cutoff analog switch control bit
0	Turn on the analog switch only during conversion.
1	Always turn on the analog switch on.

ADMV	Conversion flag bit
0	No conversion
1	Conversion in progress

ADI	Interrupt request flag bit	
	Read	Write
0	Conversion not completed	Clear this bit.
1	Conversion completed	Writing “1” does not change ADI or affect other bits.

ANS3	ANS2	ANS1	ANS0	Analog input pin 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

R/W : Readable/writable (The read value is the same as the write value.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R0,W : Write only (Writable. "0" is read.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
□ : Initial value

Do not select an unusable pin for the MB95200H/210H Series with the analog input pin select bits (ANS3 to ANS0).

Table 17.4-1 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 pin select bits	These bits select an analog input pin to be used from AN00 to AN05. The number of analog input pins differs between the two series. When A/D conversion is started (AD = 1) by the software (ADC2: EXT = 0), these bits can be modified simultaneously. Note: When the ADMV bit is "1", do not modify these bits. Pins not used as analog input pins can be used as general-purpose ports.
bit3	ADI: Interrupt request flag bit	This bit detects the completion of A/D conversion. • When the A/D conversion function is used, the bit is set to "1" immediately after A/D conversion is complete. • Interrupt requests are output when this bit and the interrupt request enable bit (ADC2: ADIE) are both set to "1". • When "0" is written to this bit, it is cleared. Writing "1" to this bit does not change it or affect other bits. • When read by the read-modify-write (RMW) type of instruction, this bit returns "1".
bit2	ADMV: Conversion flag bit	This bit indicates that A/D conversion is in progress. The bit is set to "1" during A/D conversion. This bit is read-only. A value written to this bit is meaningless and has no effect on operation.
bit1	ADMVX: Analog switch for current cutoff control bit	This bit controls the analog switch for cutting off the internal reference power supply. Since rush current flows immediately after A/D conversion starts, when the external impedance of Vcc pin is high, A/D conversion precision may be affected. This can be avoided by setting this bit to "1" before A/D conversion starts. In addition, in order to reduce current consumption, set the bit to "0" before transiting to standby mode.
bit0	AD: A/D conversion start bit	This bit activates A/D conversion function with the software. Writing "1" to the bit activates the A/D conversion function. Note: Writing "0" to this bit cannot stop the operation of the A/D conversion function. The read value of this bit is always "0". When EXT = 1, starting the A/D conversion with this bit is disabled. With EXT = 0, when "1" is written to this bit while A/D conversion is in progress, A/D conversion restarts.

17.4.2 8/10-bit A/D Converter Control Register 2 (ADC2)

The 8/10-bit A/D converter control register 2 (ADC2) is used to control different functions of the 8/10-bit A/D converter, select the input clock, and enable and disable interrupts.

■ 8/10-bit A/D Converter Control Register 2 (ADC2)

Figure 17.4-3 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 ADC1 register
1	Continuous activation by 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	No external start signal is used.
1	Starts by 8/16-bit composite timer output pin (TO00)

TIM1	TIM0	Sampling time select bits
0	0	CKIN x 4
0	1	CKIN x 7
1	0	CKIN x 10
1	1	CKIN x 16

AD8	Precision select bit
0	10-bit precision
1	8-bit precision

MCLK : Machine clock
R/W : Readable/writable (The read value is the same as the write value.)
□ : Initial value

Table 17.4-2 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. **Writing "0":** 10-bit precision is selected. **Writing "1":** 8-bit precision is selected. Reading the ADDL register can obtain 8-bit data. Note: The data bits to be used are different depending on the resolution selected. Modify this bit only when the A/D converter has stopped operating.
bit6, bit5	TIM1, TIM0: Sampling time select bits	These bits set the sampling time. • Modify the sampling time according to operating conditions (voltage and frequency). • The CKIN value is determined by the clock select bits (ADC2:CKDIV1, DKDIV0). Note: Modify these bits only when the A/D converter has stopped operating.
bit4	ADCK: External start signal select bit	This bit selects the start signal for external start (ADC2:EXT = 1).
bit3	ADIE: Interrupt request enable bit	This bit enables or disables outputting interrupts to the interrupt controller. • Interrupt requests are output when both this bit and the interrupt request flag bit (ADC1: ADI) have been set to "1".
bit2	EXT: Continuous activation enable bit	This bit selects whether to activate the A/D conversion function with the software, or to continuously activate the A/D conversion function whenever a rising edge of the input clock is detected.
bit1, bit0	CKDIV1, CKDIV0: Clock select bits	These bits select the clock to be used for A/D conversion. The input clock is generated by the prescaler. See "CHAPTER 6 CLOCK CONTROLLER" for details. • The sampling time varies according to the clock selected by these bits. • Modify these bits according to operating conditions (voltage and frequency). Note: Modify these bits only when the A/D converter has stopped operating.

17.4.3 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) store the results of 10-bit A/D conversion during 10-bit A/D conversion.
The upper two bits of 10-bit data are stored in the ADDH register and the lower eight bits the ADDL register.

■ 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)

Figure 17.4-4 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 a value to it has no effect on operation.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)

The upper two bits of 10-bit A/D data correspond to bit1 and bit0 in the ADDH register and the lower eight bits bit15 to bit8 in the ADDL register.

If the AD8 bit in ADC2 register is set to "1", 8-bit precision is selected. Reading the ADDL register can obtain 8-bit data.

These two registers are read-only registers. Writing data to them has no effect on operation.

In A/D conversion in which 8-bit precision is selected, SAR8 and SAR9 in the ADDH register become "0".

● A/D conversion function

When A/D conversion is started, the results of conversion are finalized and stored in the ADDH and ADDL registers after the conversion time according to the register settings elapses. After A/D conversion is completed and before the next A/D conversion is completed, read A/D data registers (conversion results), and clear the ADI flag bit (bit 3) in the ADC1 register. During A/D conversion, the values of the ADDH and ADDL registers are results of the last A/D conversion.

17.5 Interrupts of 8/10-bit A/D Converter

The completion of conversion during the operation of the A/D converter is an interrupt source of the 8/10-bit A/D converter.

■ 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 has been enabled (ADC2: ADIE = 1), an interrupt request is made 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 to "1" 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 having been enabled (ADC2: ADIE = 1). Always clear the ADI bit in the interrupt service routine.

■ Register and Vector Table Addresses Related to 8/10-bit A/D Converter Interrupts

Table 17.5-1 Register and Vector Table Addresses Related to 8/10-bit A/D Converter Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
8/10-bit A/D	IRQ18	ILR4	L18	$FFD6_H$	$FFD7_H$

See "APPENDIX B Table of Interrupt Sources" for all interrupt request numbers and vector tables addresses.

17.6 Operations of 8/10-bit A/D Converter and Setting Procedure Example

The 8/10-bit A/D converter can activate A/D conversion with the software or activate A/D conversion continuously according to the setting of the EXT bit of the ADC1 register.

■ Operations of 8/10-bit A/D Converter Conversion Function

● Software activation

The settings shown in Figure 17.6-1 are required for activating the A/D conversion function with the software.

Figure 17.6-1 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 retained							

◎: Bit to be used
× : Unused bit
1 : Set to "1"
0 : Set to "0"

When the A/D conversion function is activated, A/D conversion starts. In addition, the A/D conversion function can be re-activated even during conversion.

● Continuous activation

The settings shown in Figure 17.6-2 are required for continuous activation of the A/D conversion function.

Figure 17.6-2 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	

◎: Bit to be used
× : Unused bit
1 : Set to "1"

When continuous activation is enabled, the A/D conversion function is activated at the rising edge of the input clock selected to start A/D conversion. Continuous activation is stopped when disabled (ADC2:EXT = 0).

■ Operations of A/D Conversion Function

This section explains the operations of 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 in the analog input pin is loaded into a sample-and-hold capacitor in the sample-and-hold circuit during the sampling cycle. This voltage is held until A/D conversion is completed.
3) The comparator in the control circuit compares the voltage loaded into 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 transfers the results to the ADDH and ADDL registers.
   After the results have been transferred to the two registers, the conversion flag bit is cleared (ADC1:ADMV = 0) and the interrupt request flag bit is set to "1" (ADC1:ADI = 1).

Notes:

- The contents of the ADDH and ADDL registers are retained until the end of A/D conversion. Therefore, during A/D conversion, the values resulting from last conversion will be returned if the two registers are read.
- Do not change the analog input pin (ADC1: ANS3 to ANS0) while AD conversion function is being used. During continuous activation in particular, disable continuous activation (ADC2: EXT = 0) before changing the analog input pin.
- The start of the reset mode, the stop mode or the watch mode causes the A/D converter to stop and the ADMV bit to be cleared to "0".

■ Setting Procedure Example

Below is an example of procedure for setting the 8/10-bit A/D converter:

● Initial settings

1) Set the input port (DDR1).
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 precision (ADC2:AD8).
7) Select the operating mode (ADC2:EXT).
8) Select the start trigger (ADC2:ADCK).
9) Enable interrupts (ADC2:ADIE = 1).
10) Activate the A/D conversion function (ADC1:AD = 1).

● Interrupt processing

1) Clear the interrupt request flag (ADC1:ADI = 0).
2) Read converted values (ADDH, ADDL).
3) Activate the A/D conversion function (ADC1:AD = 1).

17.7 Notes on Using 8/10-bit A/D Converter

This section provides notes on using the 8/10-bit A/D converter.

■ Notes on Using 8/10-bit A/D Converter

● Notes on setting the 8/10-bit A/D converter with a program

- The contents of the ADDH and ADDL registers are retained until the end of A/D conversion. Therefore, during A/D conversion, the values resulting from last conversion will be returned if the two registers are read.
- Do not change the analog input pin (ADC1: ANS3 to ANS0) while AD conversion function is being used. During continuous activation in particular, disable continuous activation (ADC2: EXT = 0) before changing the analog input pin.
- The start of the reset mode, the stop mode or the watch mode causes the A/D converter to stop and the ADMV bit to be cleared to "0".
- The CPU cannot return from interrupt processing if the interrupt request flag bit (ADC1: ADI) is "1" with interrupt requests having been enabled (ADC2: ADIE = 1). Always clear the ADI bit in the interrupt service routine.

● Note on interrupt requests

If the restart of A/D conversion (ADC1: AD = 1) and the completion of A/D conversion occur simultaneously, the interrupt request flag bit (ADC1: ADI) is set.

● A/D conversion error

As | Vcc - Vss | decreases, the A/D conversion error increases proportionately.

● 8/10-bit A/D converter analog input sequences

Apply the analog input (AN00 to AN05) and the digital power supply (V_{CC}) simultaneously, or apply the analog input after applying the digital power supply.

Disconnect the digital power supply (V_{CC}) at the same time as the analog input (AN00 to AN05), or after disconnecting analog input (AN00 to AN05).

Ensure that the analog input voltage does not exceed the voltage of digital power supply when turning on or off the power of the 8/10-bit A/D converter.

● Conversion time

The conversion speed of A/D conversion function is affected by clock mode, main clock oscillation frequency and main clock speed switching (gear function).

Example: Sampling time = CKIN × (ADC2: TIM1/TIM0 setting)

Compare time = CKIN x 10 (fixed value) + MCLK

A/D converter startup time: minimum = MCLK + MCLK

maximum =MCLK + CKIN

Conversion time = A/D converter startup time + sampling time + compare time

- The conversion time may have an error of up to (1 CKIN – 1 MCLK), depending on the time at which A/D conversion starts.
- When setting the A/D converter in software, ensure that the settings satisfy the specifications of "sampling time" and "compare time" of the A/D converter mentioned in the data sheet of the MB95200H/210H Series.

17.8 Sample Programs for 8/10-bit A/D Converter

Fujitsu Microelectronics provides sample programs for operating the 8/10-bit A/D converter.

■ Sample Programs for 8/10-bit A/D Converter

For the sample programs of the 8/10-bit A/D converter, see "■ Sample Programs" in "PREFACE".

■ Setting Methods Other than Sample Programs

● Method of selecting an operating clock for the 8/10-bit A/D converter

Use the clock select bits (ADC2:CKDIV1/CKDIV0) to select an operating clock.

● Method of selecting the sampling time of the 8/10-bit A/D converter

Use the sampling time select bits (ADC2:TIM1/TIM0) to select sampling time.

● Method of controlling the analog switch for cutting off the internal reference power supply of the 8/10-bit A/D converter

Use the analog switch control bit (ADC1:ADMVX) to control the analog switch for cutting off internal reference power.

Control item	Analog switch for current cutoff control bit (ADMVX)
To switch off internal reference power supply	Set the bit to "0".
To switch on internal reference power supply	Set the bit to "1".

● Method of selecting the method of activating the 8/10-bit A/D conversion function

Use the continuous activation enable bit (ADC2:EXT) to select an activation trigger.

A/D conversion activation source	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".

- Method of 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".

- Method of activating the A/D conversion function using the input clock
  An activation trigger is generated at the rising edge of the input clock.

To select the input clock, use external start signal select bit (ADC2:ADCK).

Input clock	External start signal select bit (ADCK)
Do not use any external start signal	Set the bit to "0".
To select the 8/16-bit composite timer output pin (TO00)	Set the bit to "1".

● Method of selecting A/D conversion precision

Use the precision select bit (ADC2:AD8) to select the precision of conversion results.

Operating 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".

● Method of using analog input pins

Use the analog input pin select bits (ADC1:ANS3 to ANS0) to select an analog input pin.

Operation	Analog input pin select bits (ANS3 to ANS0)
To use the AN0 pin	Set the bits to "0000_B".
To use the AN1 pin	Set the bits to "0001_B".
To use the AN2 pin	Set the bits to "0010_B".
To use the AN3 pin	Set the bits to "0011_B".
To use the AN4 pin	Set the bits to "0100_B".
To use the AN5 pin	Set the bits to "0101_B".

● Method of checking the completion of conversion

There are two methods of checking whether conversion has been completed or not.

- Checking with the interrupt request flag bit (ADC1:ADI)

Interrupt request flag bit (ADI)	Meaning
The read value is "0".	No A/D conversion completion interrupt request
The read value is "1".	A/D conversion completion interrupt request made

- Checking with the conversion flag bit (ADC1:ADMV)

Conversion flag bit (ADMV)	Meaning
The read value is "0".	A/D conversion completed (stopped)
The read value is "1".	A/D conversion in progress

● Interrupted-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 AD converter	Interrupt level register (ILR4) Address: $0007D_H$	#18 Address: $0FFD6_H$

● Method of enabling, disabling, and clearing interrupts

Use the interrupt request enable bit (ADC2:ADIE) to enable interrupts.

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".

Use the interrupt request bit (ADC1:ADI) to clear an interrupt request.

Control item	Interrupt request bit (ADI)
To clear an interrupt request	Set the bit to "1" or activate the A/D conversion function.

CHAPTER 18

LOW-VOLTAGE DETECTION RESET CIRCUIT

This chapter describes the function and operation of the low-voltage detection reset circuit. (The low-voltage detection reset circuit is available in MB95F204K/F203K/F202K/F214K/F213K/F212K only.)

18.1 Overview of Low-voltage Detection Reset Circuit

The low-voltage detection reset circuit monitors power supply voltage and generates a reset signal if the power supply voltage drops below the low-voltage detection voltage level (available in MB95F204K/F203K/F202K/F214K/F213K/F212K only).

■ Low-voltage Detection Reset Circuit

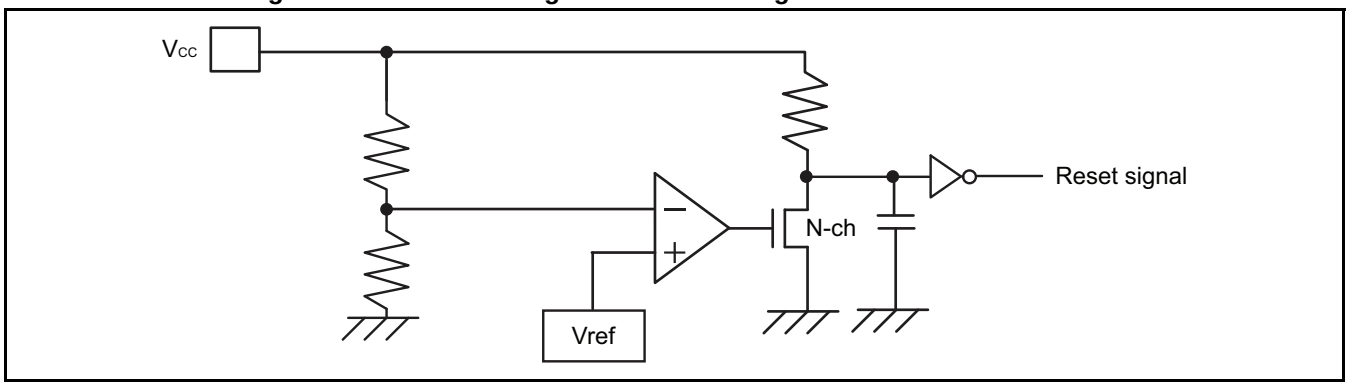
This circuit monitors power supply voltage and generates a reset signal if the power supply voltage drops below the detection voltage level. The circuit is available in MB95F204K/F203K/F202K/F214K/F213K/F212K only. Refer to the data sheet of the MB95200H/210H Series for details of the electrical characteristics.

18.2 Configuration of Low-voltage Detection Reset Circuit

Figure 18.2-1 is the block diagram of the low-voltage detection reset circuit.

■ Block Diagram of Low-voltage Detection Reset Circuit

Figure 18.2-1 Block Diagram of Low-voltage Detection Reset Circuit



18.3 Pins of Low-voltage Detection Reset Circuit

This section describes the pins of the low-voltage detection reset circuit.

■ Pins Related to Low-voltage Detection Reset Circuit

- V_{CC} pin

  The low-voltage detection reset circuit monitors the voltage of this pin.

- V_{SS} pin

  This is the GND pin serving as the reference for voltage detection.

- $\overline{RST}$ pin

  The low-voltage detection reset signal is output inside the microcontroller and to this pin.

18.4 Operation of Low-voltage Detection Reset Circuit

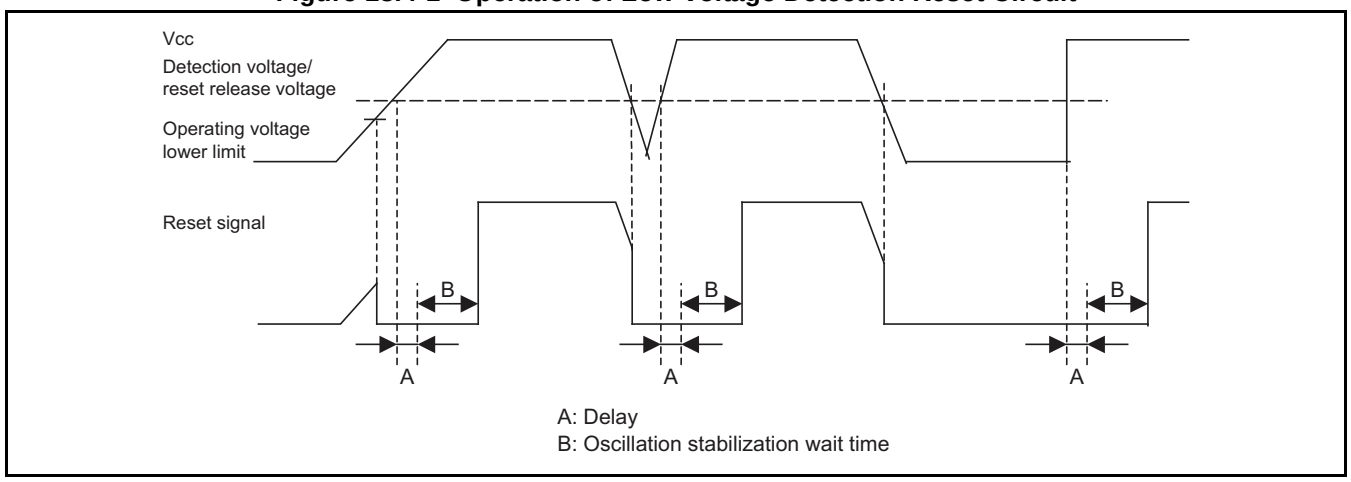
The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage.

■ Operation of Low-voltage Detection Reset Circuit

The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the low-voltage detection voltage. Afterward, if the low-voltage detection reset circuit detects the low-voltage detection reset release voltage, it outputs a reset signal lasting for the oscillation stabilization wait time and then releases the reset.

For details of the electrical characteristics, refer to the data sheet of the MB95200H/210H Series.

Figure 18.4-1 Operation of Low-voltage Detection Reset Circuit



■ Operation in Standby Mode

The low-voltage detection reset circuit keeps operating even in standby mode (stop mode, sleep mode, subclock mode and watch mode).

CHAPTER 19

CLOCK SUPERVISOR COUNTER

This chapter describes the functions and operations of the clock supervisor counter.

19.1 Overview of Clock Supervisor Counter

The clock supervisor counter can check the external clock frequency to detect the abnormal state of the external clock.

■ Overview of Clock Supervisor Counter

The clock supervisor counter can check the external clock frequency to detect the abnormal state of the external clock.

The clock supervisor counter automatically enables and disables its operation according to one of the eight different timebase timer intervals, and counts up the counter based on the external clock input.

The count clock of this module can be selected from the main oscillation clock and the sub-oscillation clock.

Note:

The clock supervisor counter must operate in main CR clock mode with the hardware watchdog timer (running in standby mode).

Otherwise, it cannot detect the abnormal state of the external clock correctly and will hang up if the external clock stops.

See "CHAPTER 11 HARDWARE/SOFTWARE WATCHDOG TIMER" for the hardware watchdog timer (running in standby mode).

19.2 Configuration of Clock Supervisor Counter

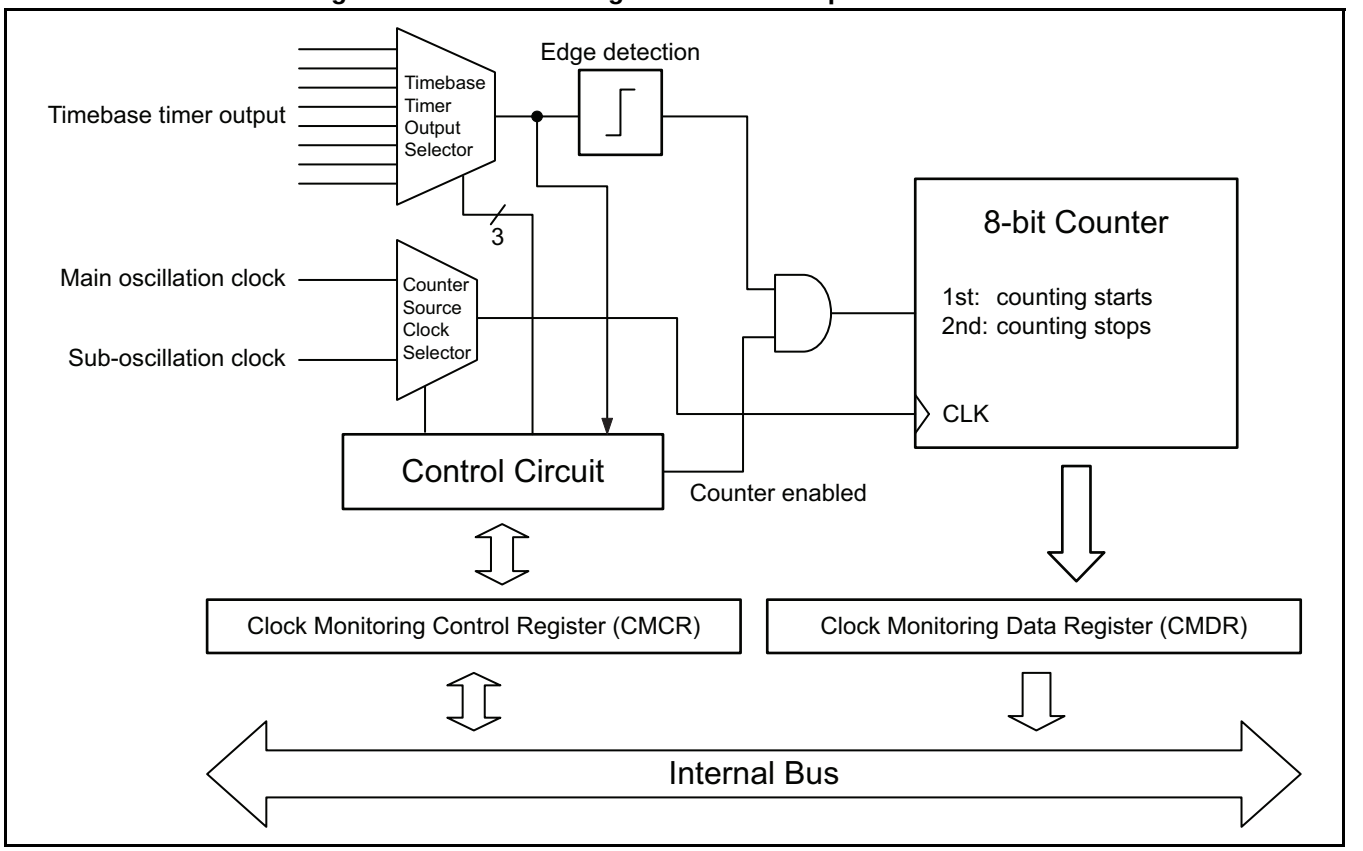
The clock supervisor counter consists of the following blocks:

- **Control circuit**
- **Clock Monitoring Control Register (CMCR)**
- **Clock Monitoring Data Register (CMDR)**
- **Timebase timer output selector**
- **Counter source clock selector**

■ Block Diagram of Clock Supervisor Counter

Figure 19.2-1 is the block diagram of the clock supervisor counter.

Figure 19.2-1 Block Diagram of Clock Supervisor Counter



- Control circuit

  This block controls the start and stop of the counter, the counter clock source, and the counter enable period based on the settings of the clock monitoring control register (CMCR).

- Clock Monitoring Control Register (CMCR)

  This register is used to select the counter source clock, select the counter enable period from the eight different timebase timer intervals, start the counter and check whether the counter is operating or not.

- Clock Monitoring Data Register (CMDR)

  This register block is used to read the counter value after the counter stops. The software can determine whether the external clock frequency is correct or not according to the contents of this register.

- Timebase timer interval selector

  This block is used to select the counter enable period from eight different timebase timer intervals.

- Counter source clock selector

  This block is used to select the counter source clock from the main oscillation clock and the sub-oscillation clock.

19.3 Registers of Clock Supervisor Counter

This section describes the registers of the clock supervisor counter.

■ Registers of Clock Supervisor Counter

Figure 19.3-1 shows the registers of the clock supervisor counter.

Figure 19.3-1 Clock Supervisor Counter Registers

Clock monitoring data register (CMDR)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FEA$_H$	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0
Read/Write	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX
Initial Value	0	0	0	0	0	0	0	0

Clock monitoring control register (CMCR)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FE9$_H$	-	-	Reserved	CMCSEL	TBTSEL2	TBTSEL1	TBTSEL0	CMCEN
Read/Write	-	-	R/WX	R/W	R/W	R/W	R/W	R/W
Initial Value	-	-	0	0	0	0	0	0

R/W : Readable/writable (The read value is the same as the write value.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)

19.3.1 Clock Monitoring Data Register (CMDR)

The clock monitoring data register (CMDR) is used to read the count value after the clock supervisor counter stops. The software can determine whether the external clock frequency is correct or not according to the contents of this register.

■ Clock Monitoring Data register (CMDR)

Figure 19.3-2 Clock Monitoring Data Register (CMDR)

Clock monitoring data register (CMDR)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FEA$_H$	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0
Read/Write	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX
Initial Value	0	0	0	0	0	0	0	0

R/WX: Read only (Readable. Writing a value to it has no effect on operation.)

The clock monitoring data register (CMDR) is used to read the counter value after the clock supervisor counter stops.

- The counter value can be read from this clock monitoring data register (CMDR). The software can check whether the external clock frequency is correct or not according to the counter value read and the timebase timer interval selected.

Table 19.3-1 Functions of Bits in Clock Monitoring Data Register (CMDR)

Bit name		Function
bit7 to bit0	CMDR7-CMDR0	The CMDR register is a data register indicating the clock supervisor counter value after the counter stops. This register is cleared if one of the following events occurs: • Reset • The CMCEN bit is modified from "0" to "1" by the software. • The CMCEN bit is modified from "1" to "0" by the software while the counter is running. • After the external clock stops, the falling edge of the selected timebase timer clock is detected twice (See Figure 19.5-2 Clock Supervisor Counter Operation 2).

Note:

This register is "0" as long as the counter is operating (CMCEN = 1).

19.3.2 Clock Monitoring Control Register (CMCR)

The clock monitoring control register (CMCR) is used to select the counter source clock, select the timebase timer interval as the counter enable period, start the counter and check whether the counter is running or not.

■ Clock Monitoring Control register (CMCR)

Figure 19.3-3 Clock Monitoring Control Register (CMCR)

Clock monitoring control register (CMCR)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FE9H	-	-	Reserved	CMCSEL	TBTSEL2	TBTSEL1	TBTSEL0	CMCEN
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	-	-	0	0	0	0	0	0

CMCEN	Counter enable bit
0	Disables the counter
1	Enables the counter

TBTSEL2	TBTSEL1	TBTSEL0	Timebase timer interval select bits
0	0	0	$2^3 \times 1/F_{CRH}$
0	0	1	$2^5 \times 1/F_{CRH}$
0	1	0	$2^7 \times 1/F_{CRH}$
0	1	1	$2^9 \times 1/F_{CRH}$
1	0	0	$2^{11} \times 1/F_{CRH}$
1	0	1	$2^{13} \times 1/F_{CRH}$
1	1	0	$2^{15} \times 1/F_{CRH}$
1	1	1	$2^{17} \times 1/F_{CRH}$

CMCSEL	Counter clock select bit
0	Main oscillation clock
1	Sub-oscillation clock

R/W : Readable/writable (The read value is the same as the write value.)
: Initial value

Table 19.3-2 Functions of Bits in Clock Monitoring Control Register (CMCR)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit5</td><td>Reserved bit</td><td>This bit is reserved.
Write "0" to this bit. The read value is always "0".</td></tr>
<tr><td>bit4</td><td>CMCSEL:
Counter clock
select bit</td><td>This bit selects the counter clock source.
Writing "0": selects the external main oscillation clock as the source clock of the counter.
Writing "1": selects the external sub-oscillation clock as the source clock of the counter.</td></tr>
<tr><td>bit3 to bit1</td><td>TBTSEL2, TBTSEL1, TBTSEL0:
Timebase timer counter output select bit</td><td>These bits select the timebase timer interval.
The operation of the clock supervisor counter is enabled and disabled according to the timebase timer counter output selected by these bits.
The first rising edge of the interval selected enables the counter operation and the second rising edge of the same output disables the counter operation.
<table>
<tr><th>TBTSEL2</th><th>TBTSEL1</th><th>TBTSEL0</th><th>Timebase timer counter output bit</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$2^3 \times 1/F_{CRH}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$2^5 \times 1/F_{CRH}$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$2^7 \times 1/F_{CRH}$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$2^9 \times 1/F_{CRH}$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$2^{11} \times 1/F_{CRH}$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$2^{13} \times 1/F_{CRH}$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$2^{15} \times 1/F_{CRH}$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$2^{17} \times 1/F_{CRH}$</td></tr>
</table></td></tr>
<tr><td>bit0</td><td>CMCEN:
Counter enable bit</td><td>This bit enables and disables the clock supervisor counter.
Writing "0": stops the counter and clears the CMDR register.
Writing "1": enables the counter. The counter starts counting when detecting the rising edge of the timebase timer interval. It stops counting when detecting the second rising edge of the same interval.
This bit is automatically set to "0" when the counter stops.</td></tr>
</table>

Notes:

- Do not modify the CMCSEL bit when CMCEN = 1.
- Do not modify the TBTSEL2-0 bits when CMCEN = 1.

19.4 Operations of Clock Supervisor Counter

This section describes the operations of the clock supervisor counter.

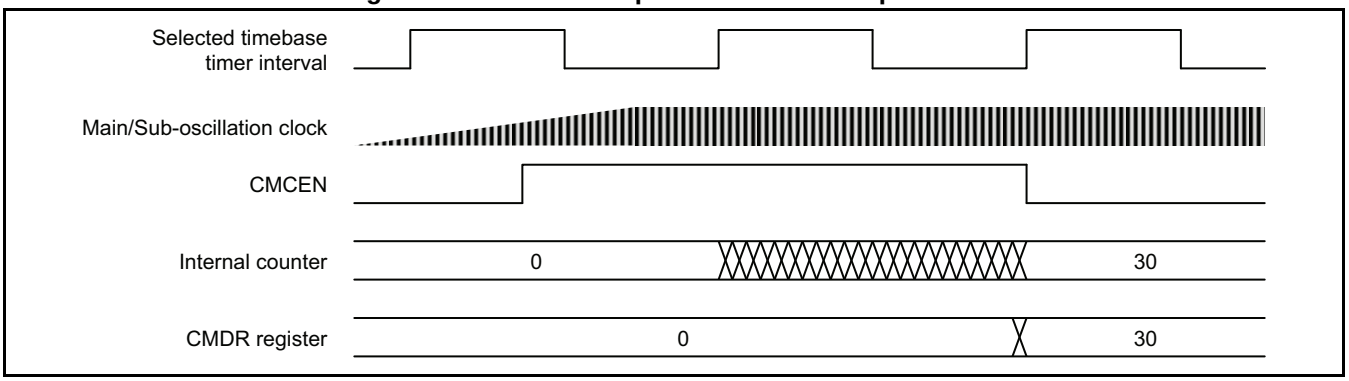
■ Clock supervisor counter

● Clock Supervisor Counter Operation 1

The clock supervisor counter is first enabled by the software (CMCEN = 1), and then the clock supervisor counter operates with the timebase timer interval selected from eight options by the TBTSEL [2:0] bits. Between two rising edges of the timebase timer interval selected, the internal counter is clocked by the external clock.

The count clock of this module can be selected from the main oscillation clock and the sub-oscillation clock.

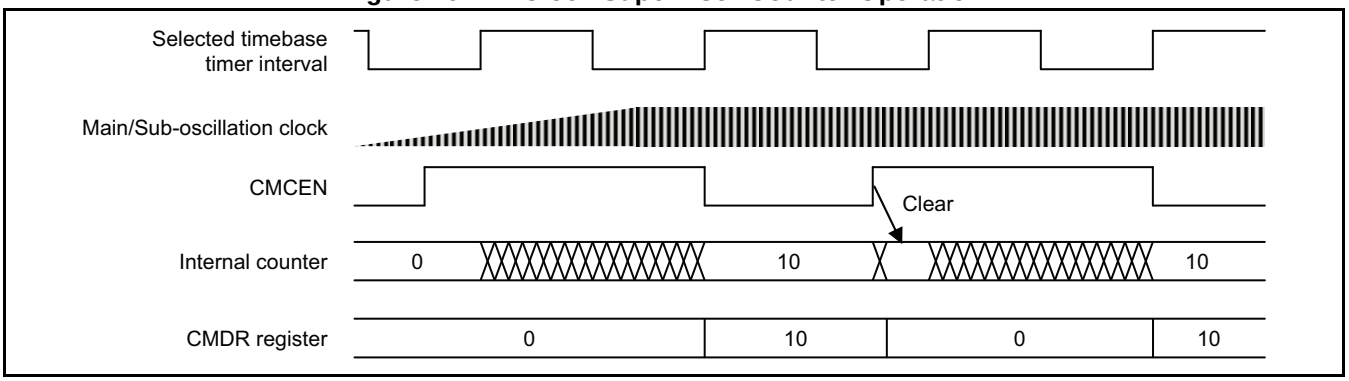
Figure 19.4-1 Clock Supervisor Counter Operation 1



● Clock Supervisor Counter Operation 2

The CMDR register is cleared when the CMCEN bit changes from "0" to "1".

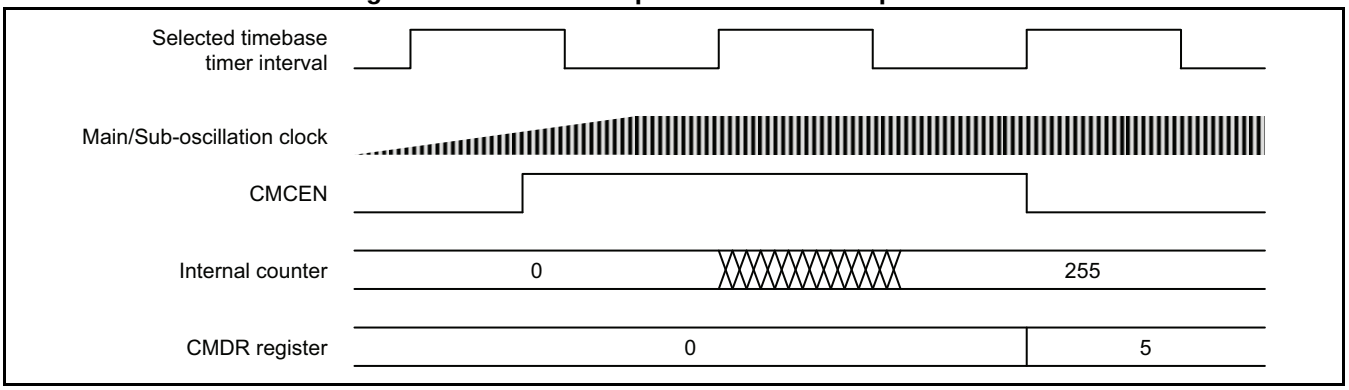
Figure 19.4-2 Clock Supervisor Counter Operation 2



● Clock Supervisor Counter Operation 3

The counter stops counting if it reaches "255". It cannot count further.

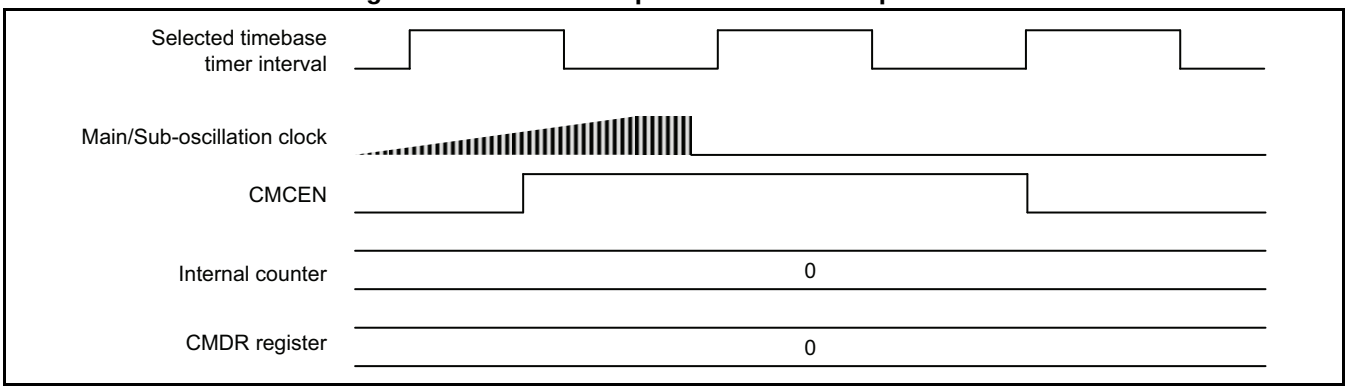
Figure 19.4-3 Clock Supervisor Counter Operation 3



● Clock Supervisor Counter Operation 4

If the external clock selected stops, the counter stops counting. The software can then identify that the external clock selected is in the abnormal state.

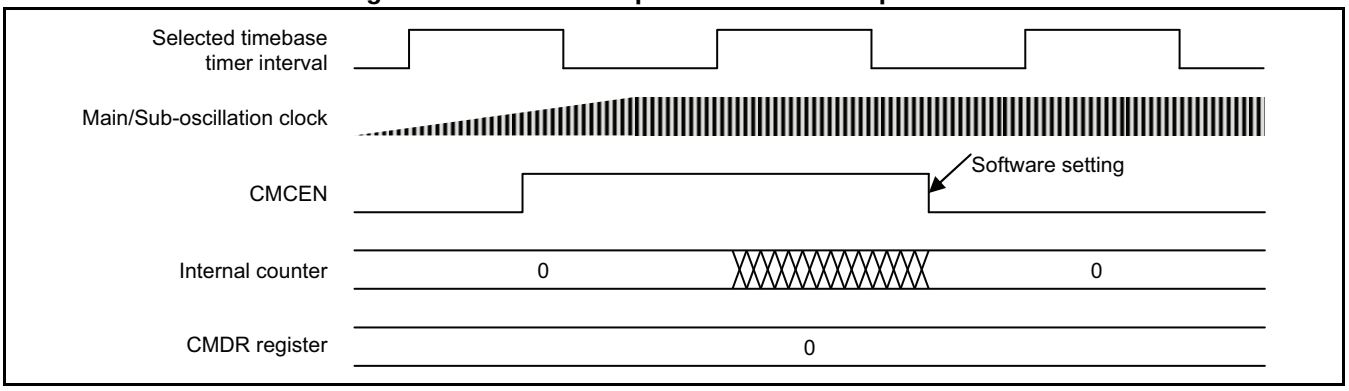
Figure 19.4-4 Clock Supervisor Counter Operation 4



● Clock Supervisor Counter Operation 5

The counter is cleared to "0" by the software if the CMCEN is set to "0" while the counter is operating.

Figure 19.4-5 Clock Supervisor Counter Operation 5



■ Table of Timebase Timer Intervals & Clock Supervisor Counter Values

Table 19.4-1 shows timebase timer intervals suitable for using different main CR clock frequency to measure different external clocks.

Table 19.4-1 Table of Counter Values in Relation to TBTSEL Settings

Main CR (F_{CRH}) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measure-ment error	TBTSEL2 - TBTSEL0							
				"000"	"001"	"010"	"011"	"100"	"101"	"110"	"111"
				(2^3×1/F_{CRH})	(2^5×1/F_{CRH})	(2^7×1/F_{CRH})	(2^9×1/F_{CRH})	(2^{11}×1/F_{CRH})	(2^{13}×1/F_{CRH})	(2^{15}×1/F_{CRH})	(2^{17}×1/F_{CRH})
1	0.03277	+5%	-1	0	0	0	6	30	126	510	2044
		-5%	+1	1	1	3	9	36	142	566	2261
	0.5	+5%	-1	0	6	29	120	486	1949	7800	31206
		-5%	+1	3	9	34	135	539	2156	8624	34493
	1	+5%	-1	2	14	59	242	974	3899	15602	62414
		-5%	+1	5	17	68	270	1078	4312	17247	68986
	4	+5%	-1	14	59	242	974	3899	15602	62414	249659
		-5%	+1	17	68	270	1078	4312	17247	68986	275942
	6	+5%	-1	21	90	364	1461	5850	23404	93621	374490
		-5%	+1	26	102	405	1617	6468	25870	103478	413912
	10	+5%	-1	37	151	608	2437	9751	39008	156037	624151
		-5%	+1	43	169	674	2695	10779	43116	172464	689853
	20	+5%	-1	75	303	1218	4875	19503	78018	312075	1248303
		-5%	+1	85	337	1348	5390	21558	86232	344927	1379706
	32.5	+5%	-1	122	494	1979	7922	31694	126779	507122	2028494
		-5%	+1	137	548	2190	8758	35032	140127	560506	2242022
8	0.03277	+5%	-1	0	0	0	0	2	14	62	254
		-5%	+1	1	1	1	2	5	18	71	283
	0.5	+5%	-1	0	0	2	14	59	242	974	3899
		-5%	+1	1	2	5	17	68	270	1078	4312
	1	+5%	-1	0	0	6	29	120	486	1949	7800
		-5%	+1	1	3	9	34	135	539	2156	8624
	4	+5%	-1	0	6	29	120	486	1949	7800	31206
		-5%	+1	3	9	34	135	539	2156	8624	34493
	6	+5%	-1	1	10	44	181	730	2924	11701	46810
		-5%	+1	4	13	51	203	809	3234	12935	51739
	10	+5%	-1	3	18	75	303	1218	4875	19503	78018
		-5%	+1	6	22	85	337	1348	5390	21558	86232
	20	+5%	-1	8	37	151	608	2437	9751	39008	156037
		-5%	+1	11	43	169	674	2695	10779	43116	172464
	32.5	+5%	-1	14	60	246	989	3960	15846	63389	253560
		-5%	+1	18	69	274	1095	4379	17516	70064	280253

Table 19.4-1 Table of Counter Values in Relation to TBTSEL Settings

Main CR (F_{CRH}) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measure-ment error	TBTSEL2 - TBTSEL0							
				"000"	"001"	"010"	"011"	"100"	"101"	"110"	"111"
				($2^3 \times 1/F_{CRH}$)	($2^5 \times 1/F_{CRH}$)	($2^7 \times 1/F_{CRH}$)	($2^9 \times 1/F_{CRH}$)	($2^{11} \times 1/F_{CRH}$)	($2^{13} \times 1/F_{CRH}$)	($2^{15} \times 1/F_{CRH}$)	($2^{17} \times 1/F_{CRH}$)
10	0.03277	+5%	-1	0	0	0	0	2	11	50	203
		-5%	+1	1	1	1	1	4	15	57	227
	0.5	+5%	-1	0	0	2	11	47	194	779	3119
		-5%	+1	1	1	4	14	54	216	863	3450
	1	+5%	-1	0	0	5	23	96	389	1559	6240
		-5%	+1	1	2	7	27	108	432	1725	6899
	4	+5%	-1	0	5	23	96	389	1559	6240	24965
		-5%	+1	2	7	27	108	432	1725	6899	27595
	6	+5%	-1	1	8	35	145	584	2339	9361	37448
		-5%	+1	3	11	41	162	647	2587	10348	41392
	10	+5%	-1	2	14	59	242	974	3899	15602	62414
		-5%	+1	5	17	68	270	1078	4312	17247	68986
	20	+5%	-1	6	29	120	486	1949	7800	31206	124829
		-5%	+1	9	34	135	539	2156	8624	34493	137971
	32.5	+5%	-1	11	48	197	791	3168	12677	50711	202848
		-5%	+1	14	55	219	876	3504	14013	56051	224203
12.5	0.03277	+5%	-1	0	0	0	0	1	9	39	162
		-5%	+1	1	1	1	1	3	12	46	181
	0.5	+5%	-1	0	0	1	8	38	155	623	2495
		-5%	+1	1	1	3	11	44	173	690	2760
	1	+5%	-1	0	0	3	18	77	311	1247	4992
		-5%	+1	1	2	6	22	87	345	1380	5519
	4	+5%	-1	0	3	18	77	311	1247	4992	19971
		-5%	+1	2	6	22	87	345	1380	5519	22076
	6	+5%	-1	0	6	28	116	467	1871	7488	29958
		-5%	+1	3	9	33	130	518	2070	8279	33113
	10	+5%	-1	2	11	47	194	779	3119	12482	49931
		-5%	+1	4	14	54	216	863	3450	13798	55189
	20	+5%	-1	5	23	96	389	1559	6240	24965	99863
		-5%	+1	7	27	108	432	1725	6899	27595	110377
	32.5	+5%	-1	8	38	157	632	2534	10141	40568	162278
		-5%	+1	11	44	176	701	2803	11211	44841	179362

: Recommended setting

: The counter value becomes "0" or "255".

If the timebase timer interrupt is used to make the clock supervisor counter wait for the oscillation stabilization time, please satisfy the following condition:

Timebase Timer Interval > Main/Sub-oscillation Stabilization Time × 1.05

e.g. F_{CH} = 4 MHz, F_{CRH} = 1 MHz, MWT3-0 = 1111 (in WATR register)

$$\text{Timebase Timer Interval} > \frac{(2^{14} - 2)}{4 \times 10^{6}} \times 1.05 \approx 4.3[\text{ms}]$$

↓

TBC3-0 = 0110 ($2^{13} \times 1/F_{CRH}$)

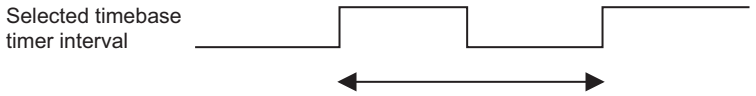
Notes:

- See "10.1 Overview of Timebase Timer" for timebase timer interval settings.
- See "6.4 Oscillation Stabilization Wait Time Setting Register (WATR)" for main/sub-oscillation stabilization time settings.

Table 19.4-1 is calculated by the following equation:

$$\text{Counter value} = \frac{\left\{\begin{array}{l} 2^{3} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{5} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{7} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{9} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{11} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{13} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{15} \times 1F_{CRH}(\text{TBTSEL=00}) \\ 2^{17} \times 1F_{CRH}(\text{TBTSEL=00}) \end{array}\right\} \times \text{Main/ Sub-Oscillation Clock Frequency}}{2} \pm 1 \text{ (Measurement error)}$$

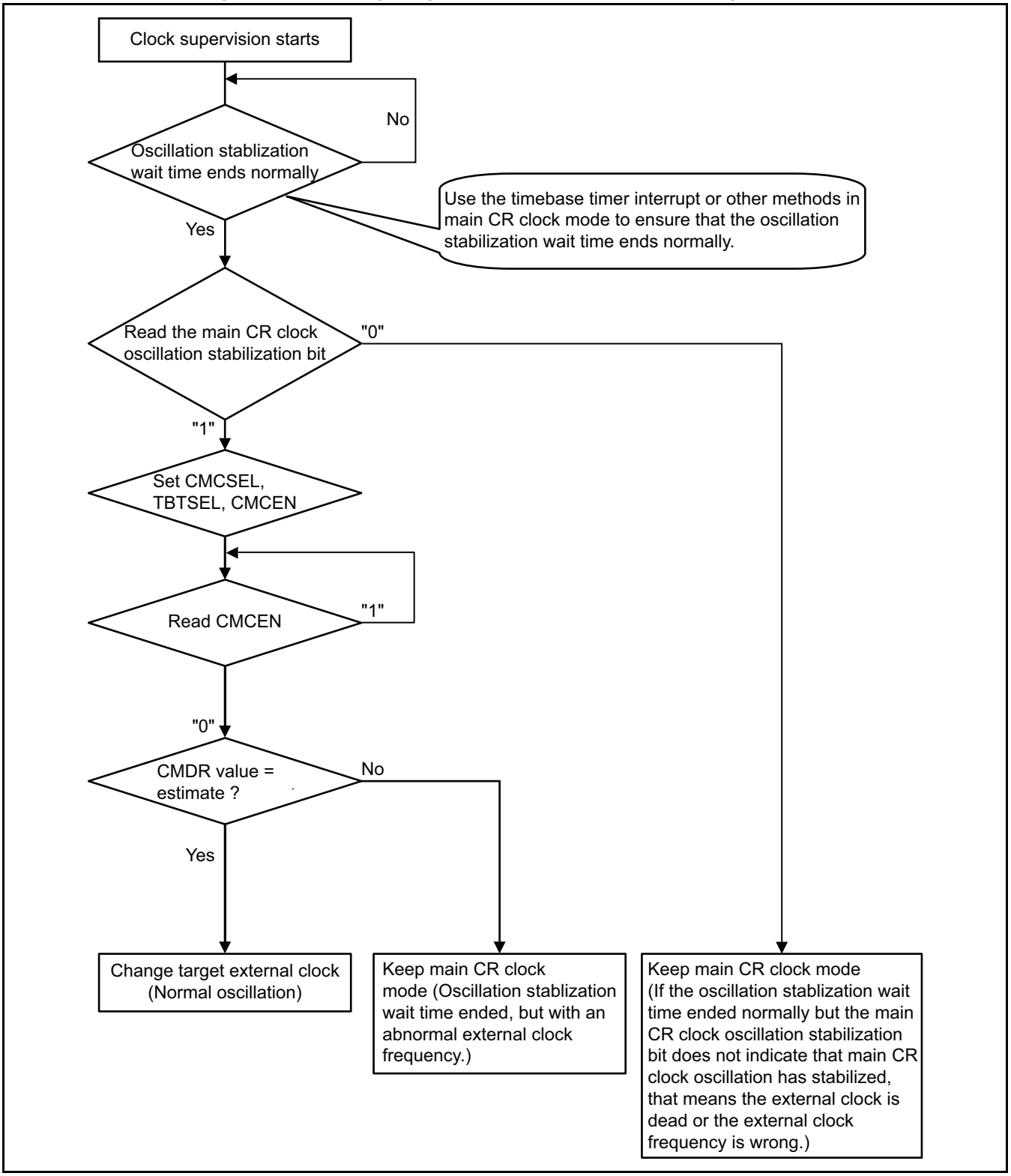
*Omit the decimal places of “Value”.



With in this period, the “Value” in the above equation is counted by the main/sub oscillation clock.

■ Sample Operation Flow Chart of Clock Supervisor

Figure 19.4-6 Sample Operation Flow Chart of Clock Supervisor



19.5 Notes on Using Clock Supervisor Counter

This section provides notes on using the clock supervisor counter.

■ Notes on Using Clock Supervisor Counter

● Restrictions

- The clock supervisor counter must operate in main CR clock mode with the hardware watchdog timer (running in standby mode). Otherwise, it cannot detect the abnormal state of the external clock correctly and will hang up if the external clock stops. See "CHAPTER 11 HARDWARE/SOFTWARE WATCHDOG TIMER" for the hardware watchdog timer (running in standby mode).
- Use main CR clock mode only. DO NOT use any other clock mode.
- If the timebase timer stops, the internal counter stops working. DO NOT clear the timebase timer while the clock supervisor counter is counting with the external clock.
- Select a timebase timer interval that is sufficiently long for the clock supervisor counter to operate. See Table 19.4-1 for timebase timer intervals.
- Read the CMDR register when CMCEN = 0. (The value of CMDR remains "0" while the clock supervisor counter is operating (CMCEN = 1).)
- When using the clock supervisor counter, ensure that the machine clock cycle is shorter than half the timebase timer interval selected. If the machine clock cycle is longer than half the timebase timer interval selected, CMCEN may remain "1" even after the clock supervisor counter stops.

Table 19.5-1 below shows the appropriate clock gear setting for each TBTSEL setting.

Table 19.5-1 Appropriate Clock Gear Setting for Respective TBTSEL

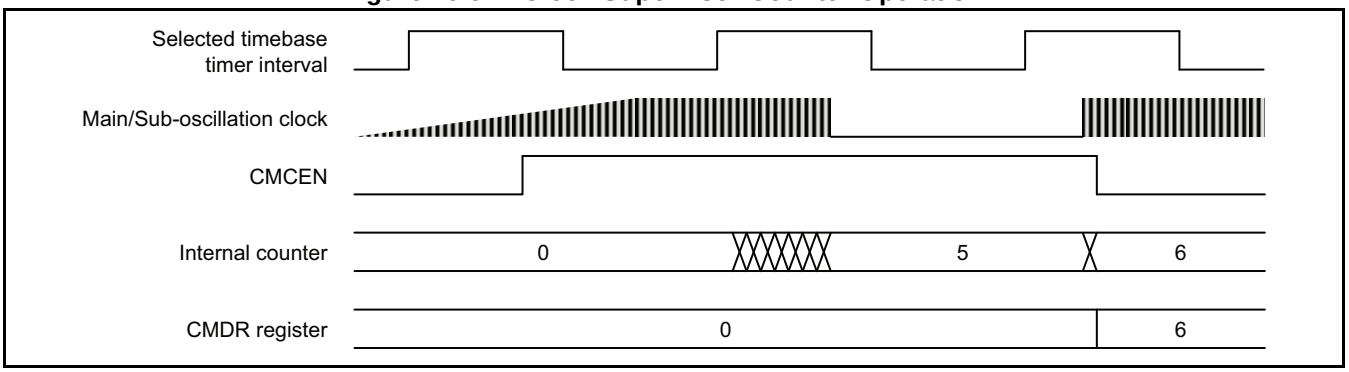
DIV (clock gear setting)	TBTSEL2 - TBTSEL0		
	000	001	010 - 111
	$2^3 \times 1/F_{RCH}$	$2^5 \times 1/F_{RCH}$	$2^7 \times 1/F_{RCH}$ - $2^{17} \times 1/F_{RCH}$
00 ($1 \times 1/F_{RCH}$)	❍	❍	❍
01 ($4 \times 1/F_{RCH}$)	x	❍	❍
10 ($8 \times 1/F_{RCH}$)	x	❍	❍
11 ($16 \times 1/F_{RCH}$)	x	x	❍

❍: Recommended

x: Prohibited

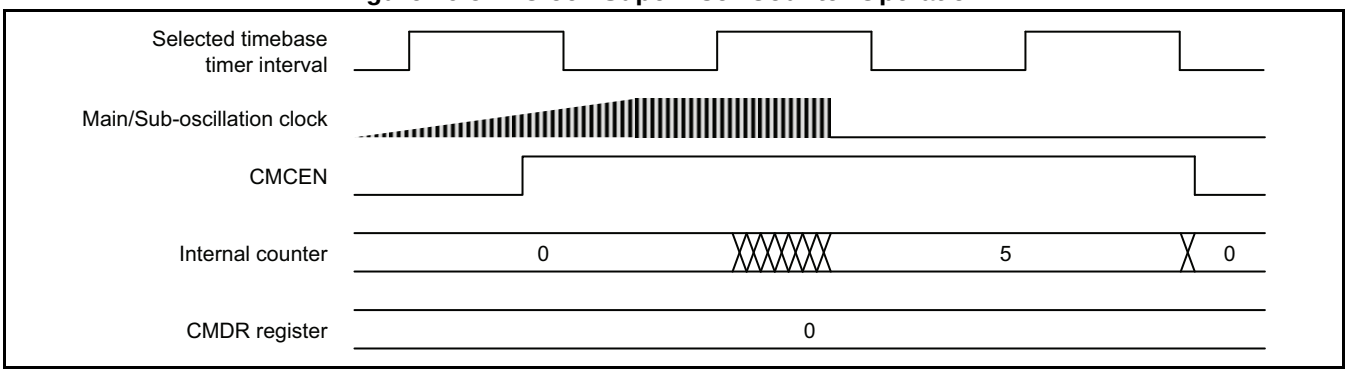
- If the external clock stops while the clock supervisor counter is operating, and it restarts after the second rising edge of the timebase timer interval selected, CMCEN is set to "0" after the external clock restarts.

Figure 19.5-1 Clock Supervisor Counter Operation 1



- After the clock supervisor counter stops, CMCEN is set to "0" when a falling edge of the timebase timer interval selected is detected after the second rising edge of the same interval. The counter is cleared at the same falling edge.

Figure 19.5-2 Clock Supervisor Counter Operation 2



CHAPTER 20

32/64/128-KBIT FLASH MEMORY

This chapter describes the function and operations of the 32/64/128-kbit flash memory.

20.1 Overview of 32/64/128-kbit Flash Memory

The following methods can be used to program (write) data into and erase data from the flash memory:

- **Programming/erasing using a dedicated serial programmer**
- **Programming/erasing by program execution**

This section describes "programming/erasing by program execution".

■ Overview of 32/64/128-kbit Flash Memory

32/64/128-kbit flash memory is located from $F000_H$ to $FFFF_H$ / $E000_H$ to $FFFF_H$ / $C000_H$ to $FFFF_H$ on the CPU memory map. The flash memory interface circuit enables read access and program access from the CPU to the flash memory.

Since data can be programmed into and erased from the flash memory by instructions from the CPU via the flash memory interface circuit, program code and data can be efficiently re-programmed with the device mounted on a circuit board.

■ Features of 32/64/128-kbit Flash Memory

- Sector configuration: 4 KB × 8 bits / 8 KB × 8 bits / 16 KB × 8 bits
- Automatic program algorithm (Embedded Algorithm)
- Detecting the completion of programming/erasing using the data polling flag
- Detecting the completion of programming/erasing by CPU interrupts
- Compatible with JEDEC standard commands
- Erase/program cycle (minimum): 100000 times

■ Programming and Erasing Flash Memory

- Writing to and reading from the flash memory cannot be executed simultaneously.
- To program data into or erase data from the flash memory, copy program code in the flash memory to the RAM temporarily and then execute the program code to write data to the flash memory.

■ High Voltage Supply on $\overline{\text{RST}}$ Pin

Apply a high DC voltage (+10 V) to the $\overline{\text{RST}}$ pin while writing data to or erasing all data from the flash memory. After applying the high voltage, wait for 10 ms before writing data to or erasing all data from the flash memory. Keep the voltage at the $\overline{\text{RST}}$ pin until data writing or data erasing is complete.

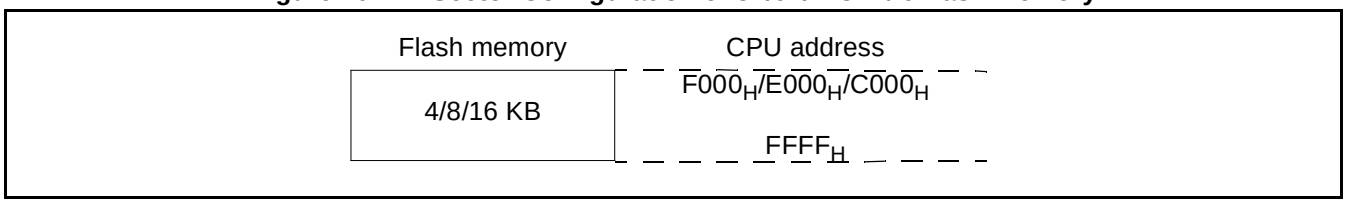
20.2 Sector Configuration of Flash Memory

This section shows the sector configuration of the flash memory.

■ Sector Configuration of 32/64/128-kbit Flash Memory

Figure 20.2-1 shows the sector configuration of the 32/64/128-kbit flash memory. The upper and lower addresses of each sector are shown in the figure.

Figure 20.2-1 Sector Configuration of 32/64/128-kbit Flash Memory



20.3 Register of Flash Memory

This section shows the register of flash memory.

■ Register of Flash Memory

Figure 20.3-1 Register of 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 (The read value is the same as write value.)
R(RM1), W: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write type of instruction.)
R/WX: Read only (Readable. Writing a value to it has no effect on operation.)
R/W0: Reserved bit (The write value is "0"; the read value is the same as the write value.)
R0/WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)

20.3.1 Flash Memory Status Register (FSR)

Figure 20.3-2 shows the bit configuration of the flash memory status register (FSR).

■ Flash Memory Status Register (FSR)

Figure 20.3-2 Flash Memory Status Register (FSR)

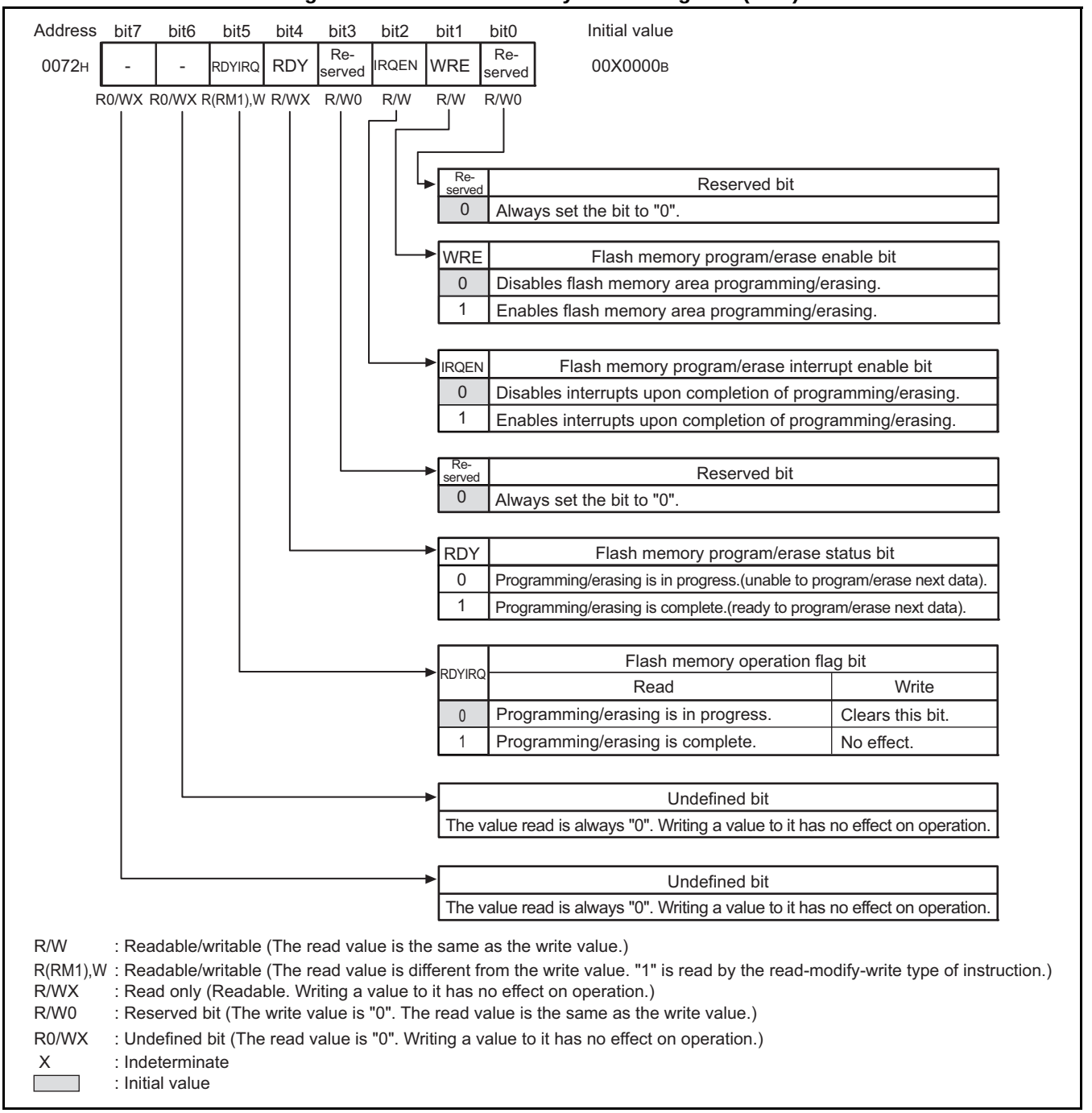


Table 20.3-1 Functions of Flash Memory Status Register (FSR)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5	RDYIRQ: Flash memory operation flag bit	This bit shows the operating state of the flash memory. After the flash memory programming erasing is completed, the RDYIRQ bit is set to "1" at the point when the automatic algorithm of the flash memory ends. • With the interrupt triggered by the completion of flash memory programming/erasing having been enabled (FSR:IRQEN = 1), if the RDYIRQ bit is set to "1", an interrupt request occurs. • After flash memory programming/erasing is completed, if the RDYIRQ bit is set to "0", flash memory programming/erasing is disabled. **Writing "0":** clears the bit. **Writing "1":** has no effect on operation. When read by the read-modify-write (RMW) instruction type of instruction, this bit always returns "1".
bit4	RDY: Flash memory program/erase status bit	This bit shows the program/erase status of the flash memory. • When the RDY bit is "0", programming data into and erasing data from the flash memory are disabled. • The read/reset command can still be accepted when the RDY bit is "0". When programming or erasing ends, the RDY bit is set to "1". • After a program/erase command is issued, there is a delay of two machine clock (MCLK) cycles before the RDY bit becomes "0". After the issue of a program/erase command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit.
bit3	Reserved: Reserved bit	Always set this bit to "0".
bit2	IRQEN: Flash memory program/erase interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the completion of flash memory programming/erasing. **Writing "0":** no interrupt request occurs even when the flash memory operation flag bit (FSR:RDYIRQ) is set to "1". **Writing "1":** an interrupt request occurs when the flash memory operation flag bit (FSR:RDYIRQ) is set to "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. **Writing "0":** no program/erase signals are generated even when a program/erase command is input. **Writing "1":** programming data into and erasing data from the flash memory are enabled after a program/erase command is input. • When not programming data into or erasing data from the flash memory, set the WRE bit to "0" in order to prevent data from being accidentally programmed into or erased from the flash memory.
bit0	Reserved: Reserved bit	Always set this bit to "0".

20.4 Invoking Flash Memory Automatic Algorithm

There are three commands that invoke the flash memory automatic algorithm: read/ reset, program and chip-erase.

■ Command Sequence Table

Table 20.4-1 lists commands used in programming/erasing flash memory.

Table 20.4-1 Command Sequence

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	-	-	-	-
Program	4	$UAAA_H$	AA_H	$U554_H$	55_H	$UAAA_H$	$A0_H$	PA	PD	-	-	-	-
Chip erase	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 : Program address
- RD : Read data
- PD : Program data
- U : Upper 4 bits same as RA and PA.
- F_X : FF/FE
- X : Arbitrary address

*: Both commands can reset the flash memory to read mode.

Notes:

- Addresses in the table above are values on the CPU memory map. All addresses and data are in hexadecimal notation. However, "X" is an arbitrary value.
- "U" in an address in the table above is not arbitrary, but represents the upper four bits (bit 15 to bit 12) of an address. Its value must be the same as the upper four bits of RA or 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 command sequence table:

Since the first command, make the value "U", which represents the upper four bits (bit 15 to bit 12) of an address, the same as the upper four bits of RA or PA.

If "U" and the upper four bits of RA or PA are not made the same, commands cannot be recognized properly. To cope with the problem above, it is necessary to initialize the command sequencer in the flash memory with a reset.

20.5 Checking Automatic Algorithm Execution Status

Since the flash memory uses the automatic algorithm to execute the program/erase flow, its internal operating status can be checked through the hardware sequence flag.

■ Hardware Sequence Flag

● Overview of hardware sequence flag

The hardware sequence flag consists of the following 1-bit output:

• Execution timeout flag (DQ5)

The hardware sequence flag can tell whether the automatic algorithm has been executed beyond the designated time.

The value of the hardware sequence flag can be checked by a read access to the address of a target sector in the flash memory after the setting of a command sequence.

Table 20.5-1 shows the bit allocation of the hardware sequence flag.

Table 20.5-1 Bit Allocation of Hardware Sequence Flag

Bit no.	7	6	5	4	3	2	1	0
Hardware sequence flag	-	-	DQ5	-	-	-	-	-

- To decide whether automatic program or chip erase is in progress or complete, check the hardware sequence flag or the flash memory program/erase status bit (RDY) in the flash memory status register (FSR). After automatic program or chip erase is complete, the flash memory returns to the read/reset state.
- When creating an automatic program/chip erase program, make the program check that automatic program/chip erase is complete with the flash memory program/erase status bit (RDY) in the flash memory status register (FSR) the DQ5 flag before reading data.

● Description of hardware sequence flag

Table 20.5-2 lists the functions of the hardware sequence flag.

Table 20.5-2 List of Hardware Sequence Flag Functions

State		DQ5
State transition during normal operation	Programming → Programming completed (when the write address has been specified)	0 → DATA: 5
	Chip erasing → Erasing completed	0 → 1
Abnormal operation	Programming	1
	Chip erasing	1

20.5.1 Execution Timeout Flag (DQ5)

The execution timeout flag (DQ5) is a hardware sequence flag indicating that the execution time of the automatic algorithm exceeds a specified time (required for programming/erasing) in the flash memory.

■ Execution Timeout Flag (DQ5)

Table 20.5-3 and Table 20.5-4 show the state transition of the execution timeout flag.

Table 20.5-3 State Transition of Execution Timeout Flag (During Normal Operation)

Operating state	Programming → Programming completed	Chip erasing → Erasing completed
DQ5	0 → DATA: 5	0 → 1

Table 20.5-4 State Transition of Execution Timeout Flag (During Abnormal Operation)

Operating state	Programming	Chip erasing
DQ5	1	1

● At programming and chip erasing

When read access is made with 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 timeout flag (DQ5) can be used to check whether programming/erasing has succeeded or failed regardless of whether the automatic algorithm is running or has terminated. When the execution timeout flag (DQ5) outputs "1" and the flash memory program/erase status bit (RDY) in the flash memory status register (FSR) reads "0", it can be decided that programming has failed.

For instance, if an attempt to write "1" to a flash memory address to which "0" has been written is made, the flash memory is locked. Thus the automatic algorithm cannot terminate, and the execution timeout flag (DQ5) outputs "1" because the execution time exceeds a specific amount of time set in the flash memory. DQ5 outputting "1" does not mean that the flash memory is defective, but means that the flash memory has not been used correctly. When DQ5 outputs "1", execute the reset command.

20.6 Details of Programming/Erasing Flash Memory

This section describes procedures for reading/resetting the flash memory, programming and chip-erasing by entering 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 the flash memory from the CPU. Always write the commands of a command sequence consecutively from the CPU to the flash memory. The termination of the automatic algorithm can be checked by reading the flash memory program/erase status bit (RDY) in the flash memory status register (FSR). After the automatic algorithm terminates normally, the flash memory returns to the read/reset state.

The operations are explained in the following order:

- Enter the read/reset state.
- Program data.
- Erase all data (chip-erase).

20.6.1 Placing Flash Memory in Read/Reset State

This section explains the procedure for entering the read/reset command to place the flash memory in read/reset state.

■ Placing Flash Memory in Read/Reset State

- To place the flash memory in the read/reset state, send read/reset commands in the command sequence table consecutively from the CPU to the 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 power-on or the normal termination of a command. The read/reset state is also regarded as the command input wait state.
- In the read/reset state, data in the flash memory can be read by a read access to the flash memory. The flash memory can be accessed from the CPU by the program access, in the same way as the masked ROM.
- In the case of read access to the flash memory, the read/reset command is not necessary. If a command does not terminate normally, use a read/reset command to initialize the automatic algorithm.

20.6.2 Programming Data into Flash Memory

This section explains the procedure for entering the program command to program data into the flash memory.

■ Programming Data into Flash Memory

- To invoke the automatic algorithm for programming data into the flash memory, send program commands in the command sequence table consecutively from the CPU to the flash memory.
- When programming data to a target address ends in the fourth cycle, the automatic algorithm is invoked and starts automatic programming.

● Addressing method

- Programming can be performed in any order of addresses and across a sector boundary. Data that can be written by a single program command is only one byte.

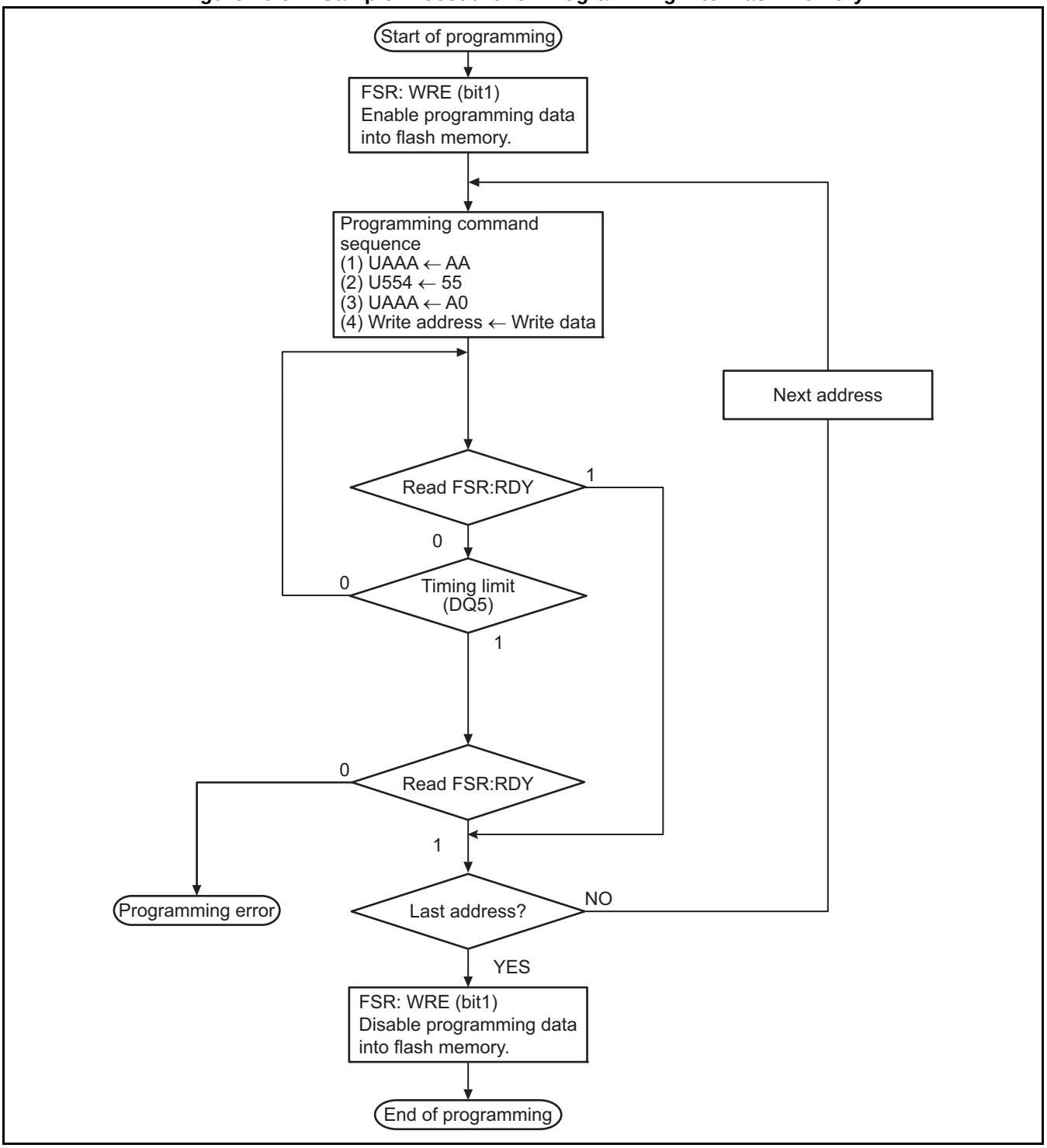
● Notes on programming data

- Bit data cannot be returned from "0" to "1" by programming. When "0" is programmed to bit data that has been "1", it is determined that flash memory component is defective, and the execution timeout flag (DQ5) indicates that an error has occurred because the execution time of the automatic algorithm exceeds the programming time specified. When data is read in the read/reset state, the bit data remains "0". To make the bit data return from "0" to "1", erase the flash memory.
- All commands are ignored during automatic programming.
- During programming, if a hardware reset occurs, the integrity of data being programmed to the current address is not guaranteed. Start programming the data from the chip-erase command again.

■ Flash Memory Programming Procedure

- Figure 20.6-1 gives an example of the procedure for programming data into the flash memory. The hardware sequence flag can be used to check the operating state of the automatic algorithm in the flash memory. The flash memory program/erase status bit (RDY) in the flash memory status register (FSR) is used for checking the end of programming data into flash memory in this example.
- Data for flag checking is read from the address to which data has been last written.
- When RDY is "0", the value of DQ5 is read from the flash memory. DQ5 outputting "1" indicates that the execution time exceeds a specific amount of time set in the flash memory. When RDY is "1", only data of the flash memory can be read from the flash memory.

Figure 20.6-1 Sample Procedure for Programming into Flash Memory



20.6.3 Erasing All Data from Flash Memory (Chip Erase)

This section explains the procedure for issuing the chip erase command to erase all data from the flash memory.

■ Erasing Data from Flash Memory (Chip Erase)

- To erase all data from the flash memory, send chip erase commands in the command sequence table consecutively from the CPU to the flash memory.
- The chip erase command is executed in six bus operations. Chip erasing starts at the point when the sixth cycle of programming commands is complete.
- In chip erase, the user does not need to program data into the flash memory before starting erasing data. While the automatic erase algorithm is running, "0" is automatically programmed to all cells in the flash memory before data in the flash memory is erased.

■ Notes on Chip Erase

During chip erase, if a hardware reset occurs, the integrity of data in the flash memory is not guaranteed.

20.7 Features of Flash Security

The flash security controller function prevents contents of the flash memory from being read by external pins.

■ Features of Flash Security

Writing protection code "01_H" to the flash memory address ($FFFC_H$) restricts access to the flash memory, disabling any read/write access to the flash memory from any external pin. Once the protection of the flash memory is enabled, the function cannot be unlocked until the chip erase command is executed.

It is advisable to write the protection code at the end of flash programming to avoid enabling unnecessary protection during programming.

Once the protection of the flash memory is enabled, the chip erase operation is required before data can be programmed to the flash memory again.

For details, consult local representatives of Fujitsu Microelectronics.

CHAPTER 21

EXAMPLE OF SERIAL PROGRAMMING CONNECTION

This chapter describes the example of serial programming connection.

21.1 Basic Configuration of Serial Programming Connection for Flash Memory Products

MB95F204H/F204K/F203H/F203K/F202H/F202K/F214H/F214K/F213H/F213K/F212H/F212K support flash ROM serial on-board programming. This section describes the configuration.

■ Basic Configuration of Serial Programming Connection for Flash Memory Products

Figure 21.1-1 shows the basic configuration of serial programming connection for flash memory products.

Figure 21.1-1 Basic Configuration of Serial Programming Connection for Flash Memory Products

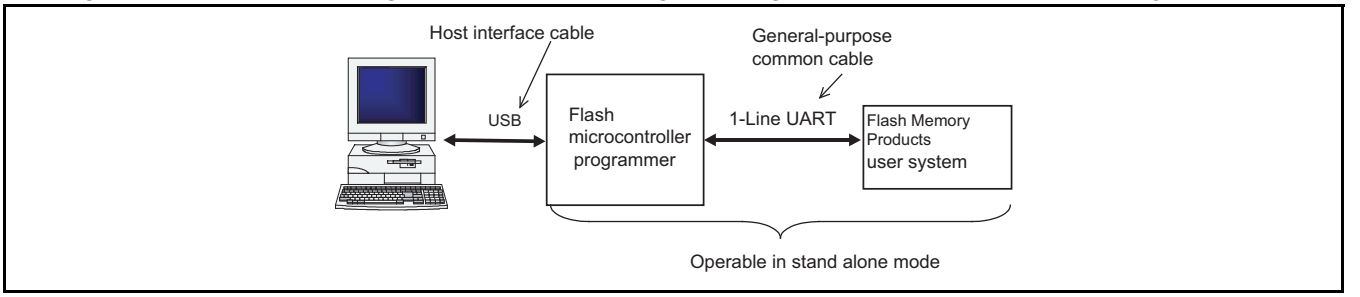


Table 21.1-1 Pins Used for Fujitsu Microelectronics Standard Serial Onboard Programming

Pin	Function	Description
V_{CC}	Power supply voltage supply pin	The write voltage (4.5 V to 5.5 V) is supplied from the user system.
V_{SS}	GND pin	It is shared with the GND of the flash microcontroller programmer.
C	Capacitor connection	Connect it to a bypass capacitor and then to the ground.
RSTX	Reset	The RSTX pin is pulled up to V_{CC} by the programmer during the normal operation. The programmer supplies 10 V directly to the RSTX pin during the flash erase/program operation.
DBG	1-line UART setting serial write mode	The DBG pin provides 1-line UART communication with the programmer. Serial write mode is set if voltage is supplied to the DBG pin and the V_{CC} pin at specific timings. (For the timings, see Figure 21.2-2.)

● Oscillation Clock Frequency

The UART clock is provided by the internal CR clock. The UART baud rate needs to be set to 31250 bps or 62500 bps depending on the flash memory operation to be executed.

21.2 Example of Serial Programming Connection

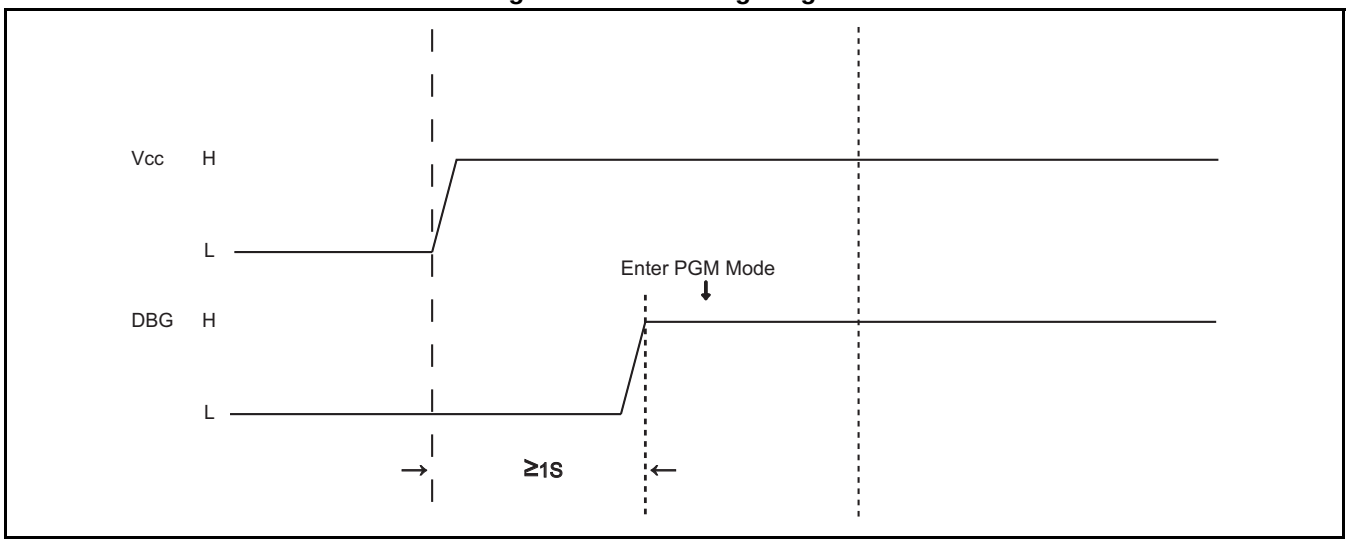
The microcontroller enters the PGM mode at the following timing.

■ MCU Entering PGM mode

The microcontroller enters the PGM mode at the following timing.

The serial programmer controls the DBG pin according to V_{CC} input.

Figure 21.2-1 Timing Diagram

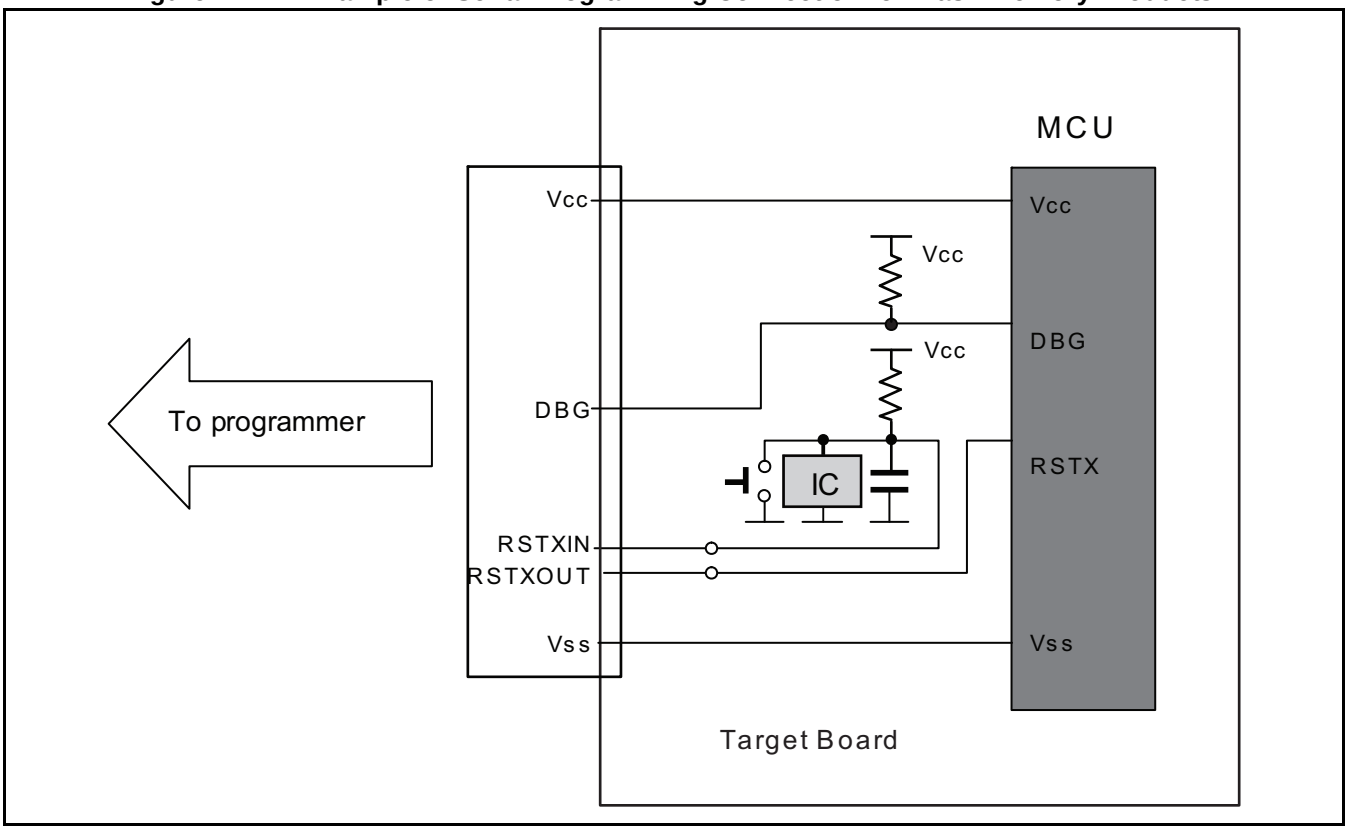


■ Example of Serial Programming Connection

Figure 21.2-2 shows an example of connection for serial writing in the flash memory products.

The power is supplied from the programmer through the V_{CC} pin to the adaptor.

Figure 21.2-2 Example of Serial Programming Connection for Flash Memory Products



CHAPTER 22

NON-VOLATILE REGISTER FUNCTION (NVR)

This chapter describes the functions and operations of the NVR interface.

22.1 Overview of NVR Interface

The NVR (Non-Volatile Register) area is a reserved area in the flash that stores system information and option settings. After a reset, data in the NVR flash area will be fetched and stored in registers in the NVR IO area. In the MB95200H/210H Series, the NVR interface is used to store the following data:

- **Frequency selection for main CR Clock (2 bits)**
- **Coarse trimming value for main CR Clock (5 bits)**
- **Fine trimming value for main CR Clock (5 bits)**
- **Watchdog Timer Selection ID (16 bits)**

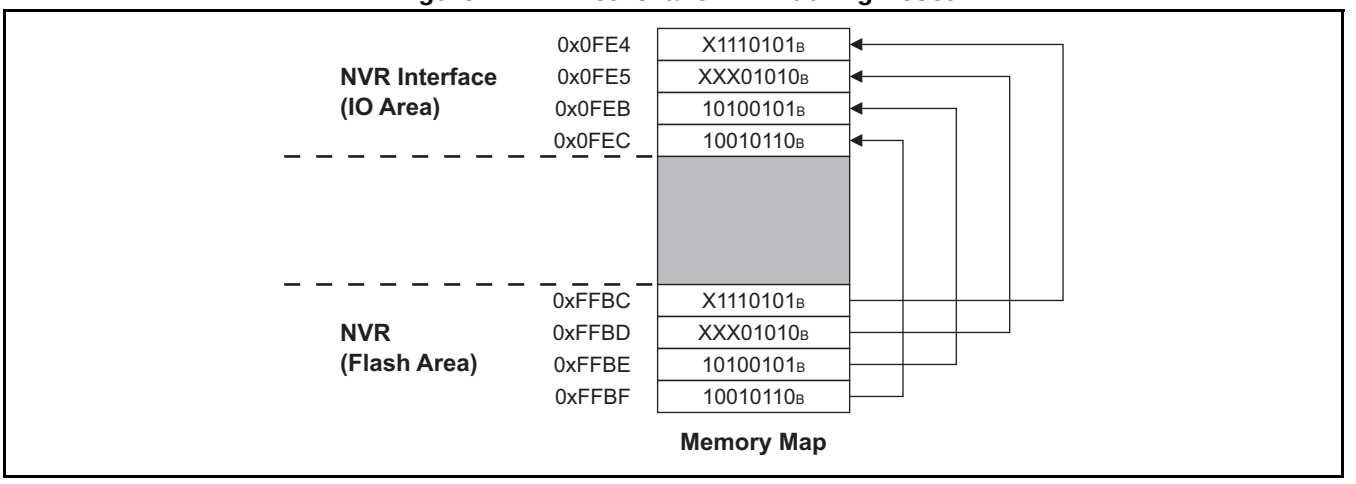
■ Functions of NVR Interface

Functions of the NVR interface are as follows:

1. The NVR interface retrieves all data from the NVR flash area and stores it in the registers in the NVR IO area after a reset. (See Figure 22.1-1 and Figure 22.2-1 below.)
2. The NVR interface enables the user to choose the frequency of the main CR clock (1 MHz/8 MHz/10 MHz/ 12.5 MHz) by setting the frequency selection bits.
3. The NVR interface enables the user to know the value of the initial CR trimming setting.
4. The NVR interface enables the user to select the hardware watchdog timer or software watchdog timer by modifying the 16-bit watchdog timer selection ID (The watchdog timer selection ID cannot be modified while the CPU is running.)

Figure 22.1-1 shows the basic configuration of serial programming connection for the flash memory products.

Figure 22.1-1 Retrieval of NVR during Reset



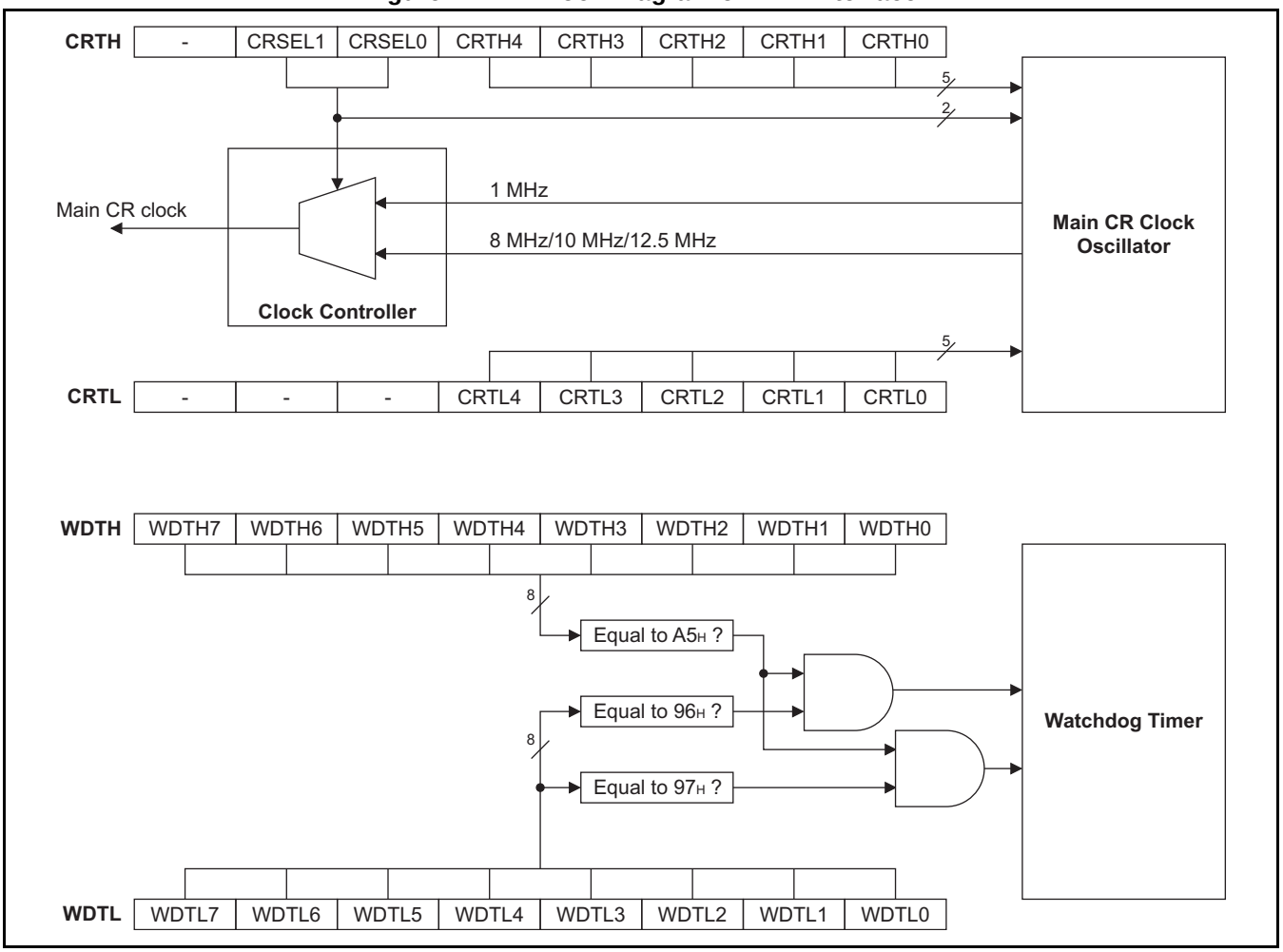
22.2 Configuration of NVR Interface

The NVR interface consists of the following blocks:

- **Main CR Clock Frequency Selection (CRSEL)**
- **Trimming of Main CR Clock (CRTH and CRTL)**
- **Watchdog Timer Selection ID (WDTH and WDTL)**

■ Block Diagram of NVR Interface

Figure 22.2-1 Block Diagram of NVR Interface



22.3 Registers of NVR Interface

This section lists the registers of the NVR interface.

■ Registers of NVR Interface

Figure 22.3-1 Registers of NVR Interface

Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FE4	CRTH	–	CRSEL1	CRSEL0	CRTH4	CRTH3	CRTH2	CRTH1	CRTH0	1XXXXXXX
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FE5	CRTL	–	–	–	CRTL4	CRTL3	CRTL2	CRTL1	CRTL0	000XXXXX
		R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	
Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FEB	WDTH	WDTH7	WDTH6	WDTH5	WDTH4	WDTH3	WDTH2	WDTH1	WDTH0	XXXXXXXX
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	
Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FEC	WDTL	WDTL7	WDTL6	WDTL5	WDTL4	WDTL3	WDTL2	WDTL1	WDTL0	XXXXXXXX
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W : Readable / Writable (The read value is the same as the write value.)
R0/WX: Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)
R/WX : Read only (This bit is readable. Writing a value to it has no effect on operation.)

22.3.1 Main CR Clock Trimming Register (Upper) (CRTH)

Figure 22.3-2 shows the main CR clock trimming register (upper) (CRTH).

■ Main CR Clock Trimming Register (Upper) (CRTH)

Figure 22.3-2 Main CR Clock Trimming Register (Upper) (CRTH)

Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FE4	CRTH	–	CRSEL1	CRSEL0	CRTH4	CRTH3	CRTH2	CRTH1	CRTH0	1XXXXXXX
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CRTH	Main CR Clock Coarse Trimming
00000	The main CR clock is tuned to the highest frequency.
⋮	⋮
11111	The main CR clock is tuned to the lowest frequency.

CRSEL	Main CR Clock Frequency Selection
00	The main CR clock is set to 1 MHz.
01	The main CR clock is set to 12.5 MHz.
10	The main CR clock is set to 10 MHz.
11	The main CR clock is set to 8 MHz.

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

Table 22.3-1 Functions of Bits in Main CR Clock Trimming Register (Upper) (CRTH)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Reserved</td><td>This bit is reserved. Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit6,bit5</td><td>CRSEL:
Main CR frequency selection</td><td>These two bits are loaded from the flash address 0xFFBC (bit6 to bit5) after a reset. Their initial values are determined by the pre-loaded values in the NVR flash area.
The frequency of the main CR clock can be selected by modifying the values of CRSEL.
<table>
<tr><th>CRSEL[1:0]</th><th>Main CR clock frequency</th></tr>
<tr><td>00_B</td><td>1 MHz</td></tr>
<tr><td>01_B</td><td>12.5 MHz</td></tr>
<tr><td>10_B</td><td>10 MHz</td></tr>
<tr><td>11_B</td><td>8 MHz</td></tr>
</table>
See "22.5 Notes on Using NVR" for notes on changing the main CR frequency selection.</td></tr>
<tr><td>bit4 to bit0</td><td>CRTH:
Main CR coarse trimming</td><td>These five bits are loaded from the flash address 0xFFBC (bit4 to bit0) after a reset. Their initial values are determined by the pre-loaded values in the NVR flash area.
Coarse trimming modifies the main CR clock frequency with a bigger step. Increasing the coarse trimming value can decrease the main CR clock frequency. See the table below.
<table>
<tr><th>CRTH [4:0]</th><th>Main CR clock frequency</th></tr>
<tr><td>00000_B</td><td>Highest</td></tr>
<tr><td>:</td><td>:</td></tr>
<tr><td>11111_B</td><td>Lowest</td></tr>
</table>
See "22.4 Notes on Main CR Clock Trimming" and "22.5 Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.</td></tr>
</table>

22.3.2 Main CR Clock Trimming Register (Lower) (CRTL)

Figure 22.3-3 shows the main CR clock trimming register (lower) (CRTL).

■ Main CR Clock Trimming Register (Lower) (CRTL)

Figure 22.3-3 Main CR Clock Trimming Register (Lower) (CRTL)

Address		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
0FE5	CRTL	–	–	–	CRTL4	CRTL3	CRTL2	CRTL1	CRTL0	000XXXXX
		R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	

CRTL	Main CR Clock Fine Trimming
00000	The main CR clock is tuned to the highest frequency.
⋮	⋮
11111	The main CR clock is tuned to the lowest frequency.

R/W : Readable / Writable (The read value is the same as the write value.)
R0/WX : Undefined bit (The read value is "0". Writing a value to it has no effect on operation.)

Table 22.3-2 Functions of Bits in CR Trimming Register (Lower) (CRTL)

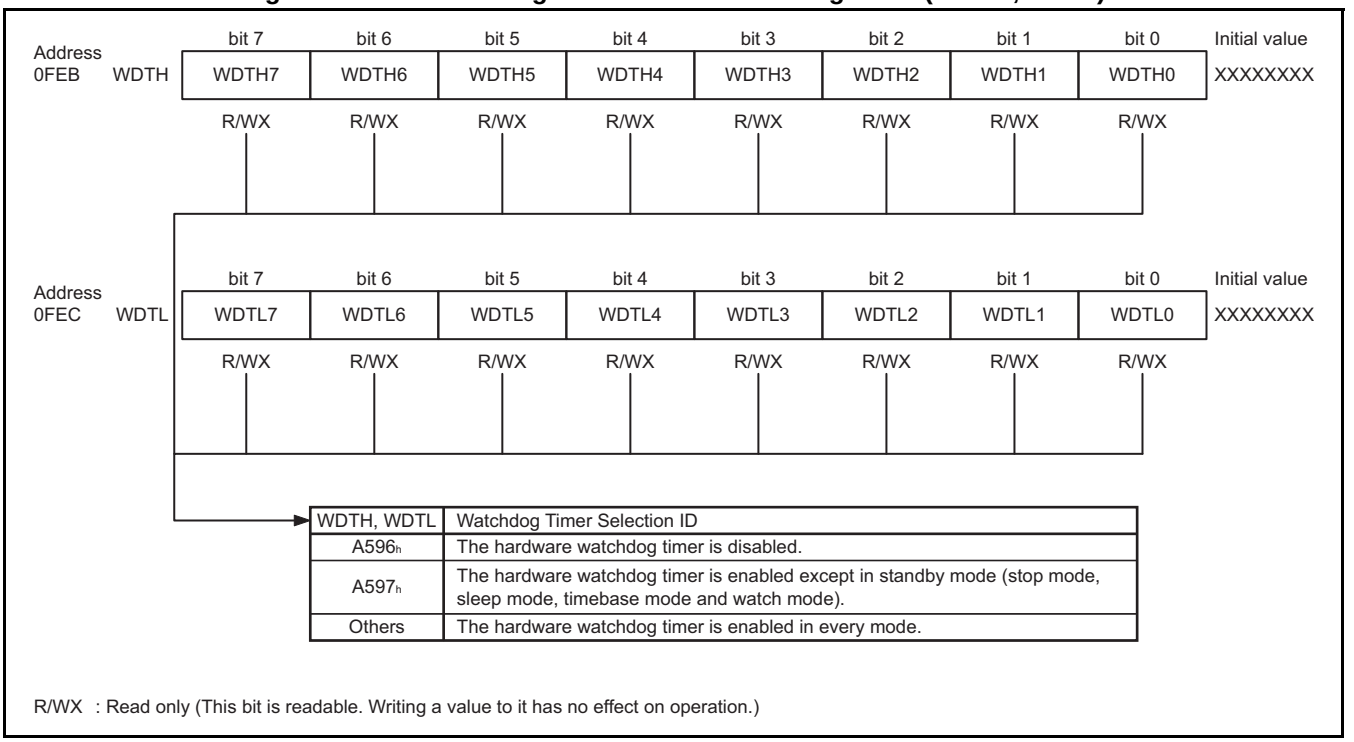
<table>
<tr><th colspan="2">Bit Name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Reserved</td><td>This bit is reserved. It always returns "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit6</td><td>Reserved</td><td>This bit is reserved. It always returns "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit5</td><td>Reserved</td><td>This bit is reserved. It always returns "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit4 to bit0</td><td>CRTL:
Main CR Fine Trimming</td><td>These five bits are loaded from the flash address 0xFFBD (bit4 to bit0) after a reset. Their initial values are determined by the pre-load values in the NVR flash area.
Fine trimming modifies the main CR clock frequency with a smaller step.
Increasing the fine trimming value can decrease the main CR clock frequency.
<table>
<tr><th>CRTL [4:0]</th><th>Main CR clock frequency</th></tr>
<tr><td>00000_B</td><td>Highest</td></tr>
<tr><td>:</td><td>:</td></tr>
<tr><td>11111_B</td><td>Lowest</td></tr>
</table>
See "22.4 Notes on Main CR Clock Trimming" and "22.5 Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.</td></tr>
</table>

22.3.3 Watchdog Timer Selection ID Registers (WDTH,WDTL)

Figure 22.3-4 shows watchdog timer selection ID registers (WDTH, WDTL).

■ Watchdog Timer Selection ID Registers (WDTH, WDTL)

Figure 22.3-4 Watchdog Timer Selection ID Registers (WDTH,WDTL)



WDTH, WDTL	Watchdog Timer Selection ID
A596h	The hardware watchdog timer is disabled.
A597h	The hardware watchdog timer is enabled except in standby mode (stop mode, sleep mode, timebase mode and watch mode).
Others	The hardware watchdog timer is enabled in every mode.

R/WX : Read only (This bit is readable. Writing a value to it has no effect on operation.)

Table 22.3-3 Functions of Bits in Watchdog Timer ID Register (Upper) (WDTH)

Bit name		Function
bit7 to bit0	**WDTH[7:0]** Watchdog timer selection ID (Upper)	These 8 bits are loaded from the flash address 0xFFBE (bit7 to bit0) after a reset. The initial values are determined by the pre-loaded values in the NVR flash area. This register cannot be modified while the CPU is running. See Table 22.3-5 for watchdog timer selection. See "22.5 Notes on Using NVR" for notes on writing NVR values.

Table 22.3-4 Functions of Bits in Watchdog Timer ID Register (Lower) (WDTL)

Bit name		Function
bit7 to bit0	**WDTL[7:0]** Watchdog timer selection ID (Lower)	These 8 bits are loaded from the flash address 0xFFBF (bit7 to bit0) after a reset. The initial values are determined by the pre-loaded values in the NVR flash area. This register cannot be modified while the CPU is running. See Table 22.3-5 for Watchdog Timer Selection. See "22.5 Notes on Using NVR" for notes on writing NVR values.

Table 22.3-5 Watchdog Timer Selection ID

WDTH[7:0],WDTL[7:0]	Function
$A596_H$	The hardware watchdog timer is disabled.
$A597_H$	The hardware watchdog timer is enabled except in standby mode (stop mode, sleep mode, timebase timer mode and watch mode).
Other than the above	The hardware watchdog timer is enabled in every mode.

22.4 Notes on Main CR Clock Trimming

This section provides notes on main CR clock trimming.

After a hardware reset, the 10-bit CR clock trimming value will be loaded from the NVR flash area to registers in the NVR IO area.

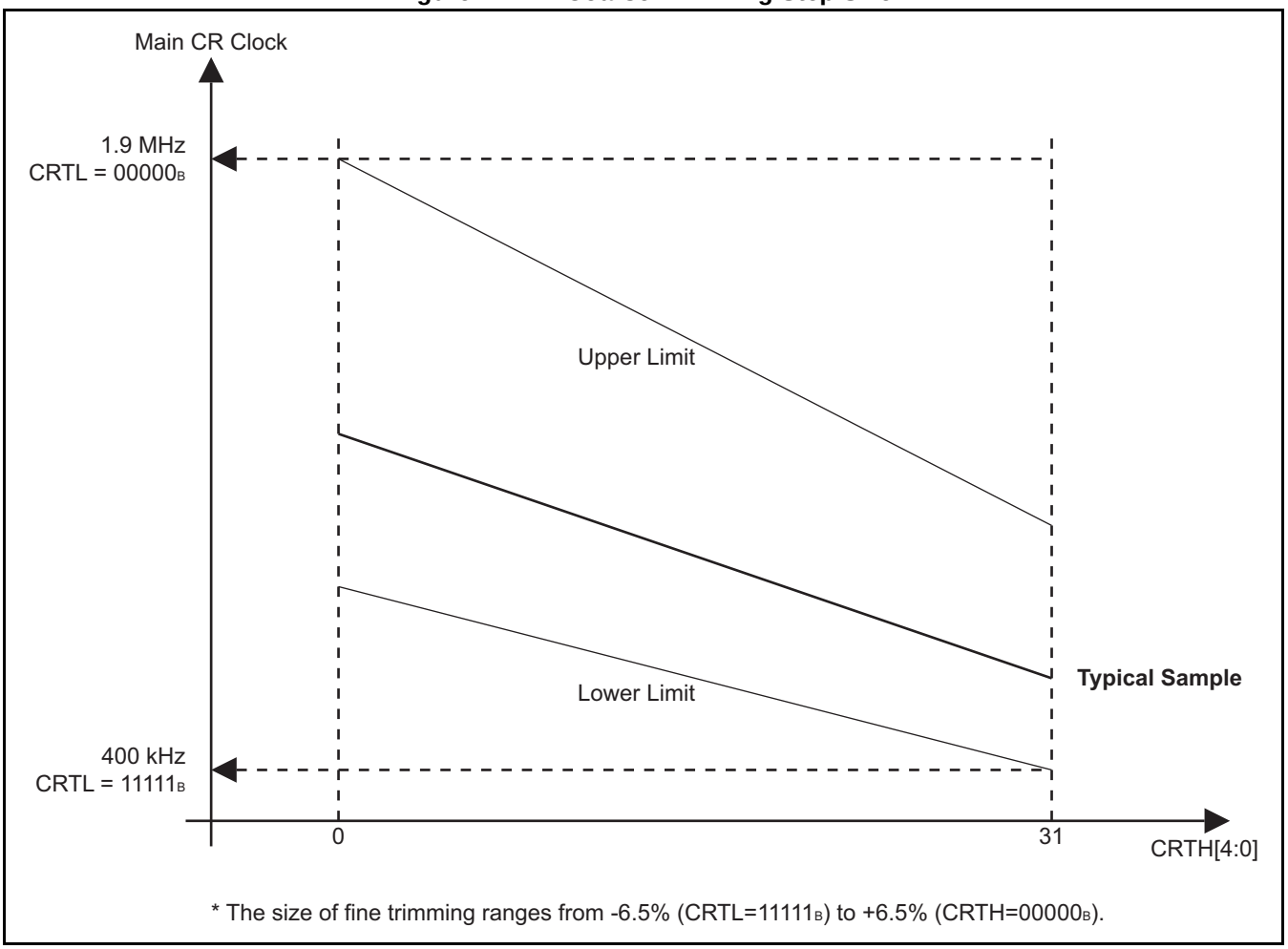
Table 22.4-1 shows the step size of CR Trimming.

Table 22.4-1 Step Size of CR Trimming

Function	Coarse trimming value CRTH[4:0]	Fine trimming value CRTL[4:0]
To achieve minimum frequency	11111_B	11111_B
To achieve maximum frequency	00000_B	00000_B
Step Size	-20 kHz to -50 kHz	-1.6 kHz to -8 kHz

The relationship between coarse trimming step size and CR frequency is shown in the diagram below.

Figure 22.4-1 Coarse Trimming Step Size



22.5 Notes on Using NVR

This section provides notes on using NVR.

■ Notes on Changing Main CR Frequency

1. The frequency of the main CR clock can be selected by writing different values to the bits CRTH:CRSEL1, CRSEL0. However, unstable oscillation occurs for a certain period of time after the modification of clock frequency has been initiated. To prevent such oscillation, it is strongly recommended that the following actions should be taken. Firstly, switch the CPU clock source from the main CR clock to another clock (main clock / subclock / sub-CR clock), then modify the main CR parameters, and switch back to the main CR clock.
2. Please note that the NVR interface does not program a modified value to the NVR flash area. If the CRTH and CRTL registers are modified, the modified value is programmed to the NVR flash area by the flash writer.

■ Notes on Flash Erase and Trimming Value

1. A flash erase operation will erase all NVR data.

   The flash writer carries out the following procedure to keep original system settings.

   (1) Make a backup of data in CRTH:CRTH4-CRTH0 and CRTL:CRTL4-CRTL0.

   (2) Erase the flash.

   (3) Restore all data in CRTH:CRTH4-CRTH0 and CRTL:CRTL4-CRTL0 to the NVR flash area.

   If there is new data in CRTH:CRTH4-CRTH0 and CRTL:CRTL4-CRTL0, the flash writer will program the new data to the NVR flash area.
2. The trimming value has been preset before this device is shipped. If the preset trimming value is modified after the device has been shipped, Fujitsu Microelectronics does not warrant proper operation of the device with respect to use based on the modified trimming value.
3. If the flash operation is performed by the user program code, the original trimming data should also be restored to the NVR flash area by the user program code. Otherwise, the trimming value, which has been preset before this device is shipped, is erased by the flash erase operation.

CHAPTER 23

CLOCK & RESET SYSTEM CONFIGURATION CONTROLLER

This chapter describes the functions and operations of the clock & reset system configuration controller (called the "controller" in this chapter).

23.1 Overview of System Configuration Register (SYSC)

The controller consists of the SYSC register, which is an 8-bit (bit 2 not used) register used to configure the clock and reset system.

■ Functions of SYSC

- Selection of the port/reset function for the PF2/RSTX pin
- Enabling/disabling reset output for the RSTX pin
- Selection of the port/oscillation function for the PG1/X0A pin and that for the PG2/X1A pin
- Selection of the port/oscillation function for the PF0/X0 pin and that for the PF12/X1 pin
- Selection of the external clock input function for the HCLK1 pin and the HCLK2 pin
- Selection of the EC0 input pin as the external count clock input pin for the 8/16-bit composite timer

23.2 System Configuration Register (SYSC)

This section provides details of the SYSC register.

■ System Configuration Register (SYSC)

Figure 23.2-1 System Configuration Register (SYSC)

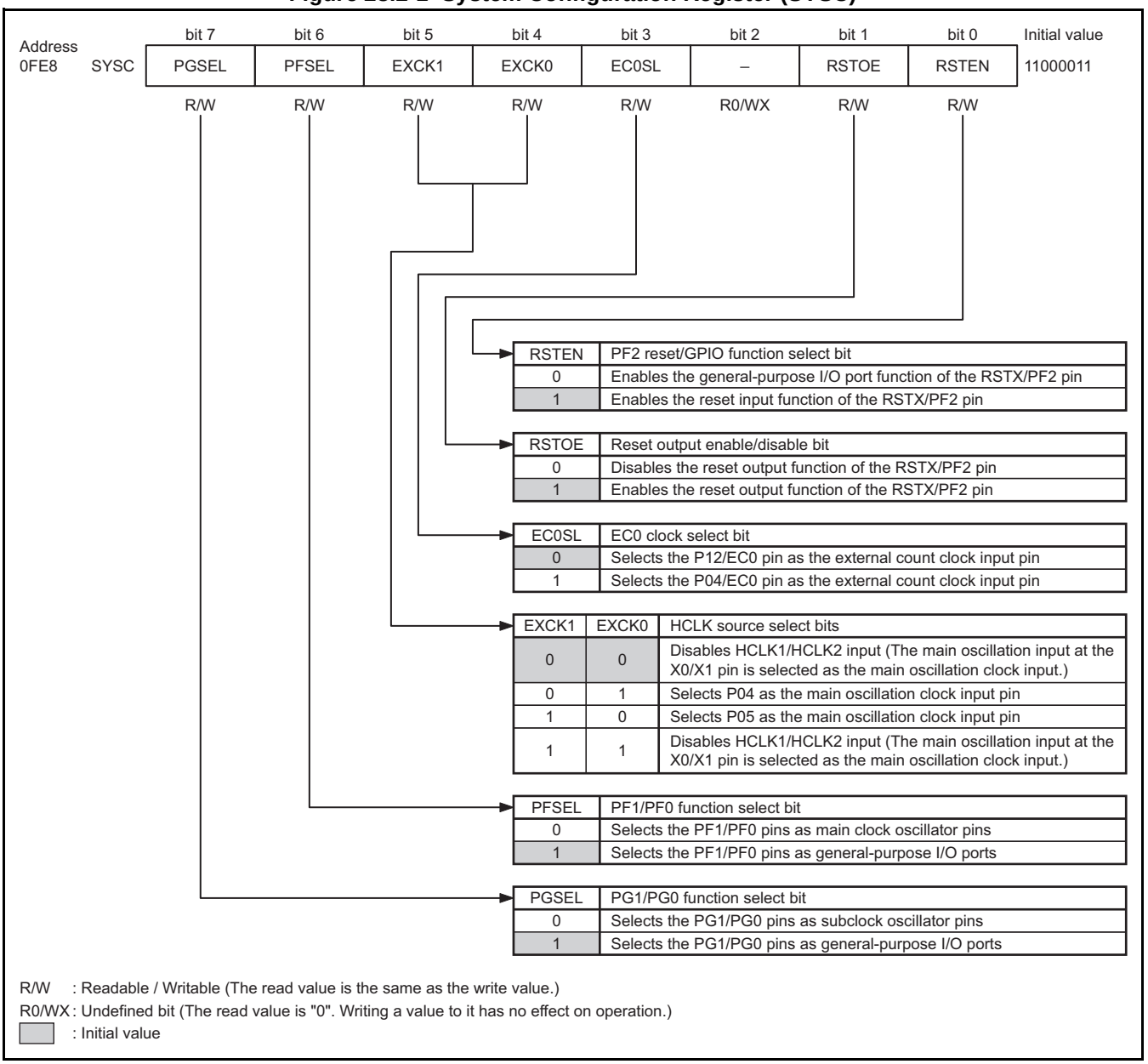


Table 23.2-1 Functions of Bits in SYSC Register

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>PGSEL:
PG1/PG0
function select bit</td><td>This bit is used to select the function of the PG1/PG0 pins.
If this bit is set to "0", the PG1/PG0 pins are selected as subclock oscillator pins, and the subclock oscillation is enabled or disabled by the subclock oscillation enable bit (SYCC2:SOSCE).
If this bit is set to "1", the PG1/PG0 pins are selected as general-purpose I/O ports.</td></tr>
<tr><td>bit6</td><td>PFSEL:
PF1/PF0
function select bit</td><td>This bit is used to select the function of the PF1/PF0 pins.
If this bit is set to "0", the PF1/PF0 pins are selected as the main clock oscillator pins, and the main clock oscillation is enabled or disabled by the main clock oscillation enable bit (SYCC2:MOSCE).
If this bit is set to "1", the PF1/PF0 pins are selected as the general-purpose I/O port.</td></tr>
<tr><td>bit5,bit4</td><td>EXCK[1:0]:
HCLK source select bits</td><td>This bit is used to select the external clock input pin to be used as the main oscillation clock.
The main oscillator clock is selected from the main oscillation input at the X0/X1 pin, HCLK1 input or HCLK2 input as shown below.

<table>
<tr><th>EXCK1</th><th>EXCK0</th><th>HCLK input pin selection</th></tr>
<tr><td>0</td><td>0</td><td>HCLK1/HCLK2 input is disabled. (The main oscillation input at the X0/X1 pin is selected as the main oscillation clock input.)</td></tr>
<tr><td>0</td><td>1</td><td>P04 is selected as the main oscillation clock input pin for HCLK1.</td></tr>
<tr><td>1</td><td>0</td><td>P05 is selected as the main oscillation clock input pin for HCLK2.</td></tr>
<tr><td>1</td><td>1</td><td>HCLK1/HCLK2 input is disabled. (The main oscillation input at the X0/X1 pin is selected as the main oscillation clock input.)</td></tr>
</table>
</td></tr>
<tr><td>bit3</td><td>EC0SL:
EC0 clock select bit</td><td>This bit is used to select the EC0 input pin to be the external count clock input pin of the 8/16-bit composite timer. (To use the EC0 input function, the corresponding register bit in the 8/16-bit composite timer must be enabled. See "CHAPTER 14 8/16-bit COMPOSITE TIMER" for details.)
If this bit is set to "0", the P12/EC0 pin is selected as the external count clock input pin.
If this bit is set to "1", the P04/EC0 pin is selected as the external count clock input pin.</td></tr>
<tr><td>bit2</td><td>Undefined bit</td><td>This bit is undefined.
When this bit is read, it always returns "0".
Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit1</td><td>RSTOE:
Reset output enable/ disable bit</td><td>This bit is used to enable and disable the reset output function of the RSTX/PF2 pin with the reset input function enabled. If the reset input function is disabled according to the setting of SYSC:RSTEN, the reset output function is disabled regardless of the setting of this bit.
See the reset input enable/disable bit (bit 0, SYSC:RSTEN) of this register.
If this bit is set to "0", the reset output function of the RSTX/PF2 pin is enabled.
If this bit is set to "1", the reset output function of the RSTX/PF2 pin is disabled.</td></tr>
<tr><td>bit0</td><td>RSTEN:
PF2 reset/GPIO
function select bit</td><td>This bit is used to enable and disable the reset input function of the RSTX/PF2 pin. The reset input function is always enabled in MB95F204H/F203H/F202H/F214H/F213H/F212H regardless of the setting of this bit.
If this bit is set to "0", the reset input function of the RSTX/PF2 pin is disabled, and the general-purpose I/O port function is enabled.
If this bit is set to "1", the reset input function of the RSTX/PF2 pin is enabled, and the general-purpose I/O port function is disabled.
Set bit 2 of the PDRF register to "1" before modifying this bit.</td></tr>
</table>

Note:

To keep the reset input/output function after the reset, RSTEN (SYSC:bit 0) and RSTOE (SYSC:bit 1) are initialized to "1" after the power is switched on. They will not be initialized by any other type of reset.

RSTEN (SYSC:bit 0) and RSTOE (SYSC:bit 1) are designed to be initialized to "1" after the power is switched on. No other types of reset can initialize the two bits above.

If the reset input/output functions have to be used in the system, it is strongly recommended that SYSC:RSTEN be initialized to "1" in the initialize program routine after a reset for stable operation. With the reset input/output functions having been enabled, all types of reset, including the watchdog reset, can be used.

23.3 Notes on Using Controller

This section provides notes on using the controller.

■ Notes on Using Controller

● Setting input pin for EC0 and HCLK

Though P04 can be selected as the input pin for EC0 and HCLK, to avoid any unexpected result, do not set P04 as the input pin for both EC0 and HCLK at the same time.

APPENDIX

This section shows the I/O map, interrupt list, memory map, pin states and mask options.

APPENDIX A I/O Map

This section shows the I/O map used in the MB95200H/210H Series.

■ II/O Map

Table A-1 I/O MAP (MB95200H Series) (1 / 3)

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	—	(Disabled)	—	—
0007_H	SYCC	System clock control register	R/W	$XXXXXX11_B$
0008_H	STBC	Standby control register	R/W	$00000XXX_B$
0009_H	RSRR	Reset source register	R	$XXXXXXXX_B$
$000A_H$	TBTC	Timebase 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
$000D_H$	SYCC2	System clock control register 2	R/W	$XX100011_B$
$000E_H$ 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 0027_H	—	(Disabled)	—	—
0028_H	PDRF	Port F data register	R/W	00000000_B
0029_H	DDRF	Port F direction register	R/W	00000000_B
$002A_H$	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
$002D_H$ to 0034_H	—	(Disabled)	—	—
0035_H	PULG	Port G pull-up register	R/W	00000000_B
0036_H	T01CR1	8/16-bit composite timer 01 status control register 1 ch. 0	R/W	00000000_B
0037_H	T00CR1	8/16-bit composite timer 00 status control register 1 ch. 0	R/W	00000000_B
0038_H	T11CR1	8/16-bit composite timer 11 status control register 1 ch. 1	R/W	00000000_B
0039_H	T10CR1	8/16-bit composite timer 10 status control register 1 ch. 1	R/W	00000000_B

Table A-1 I/O MAP (MB95200H Series) (2 / 3)

Address	Register abbreviation	Register name	R/W	Initial value
$003A_H$ to 0048_H	—	(Disabled)	—	—
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 receive/transmit 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 to $006B_H$	—	(Disabled)	—	—
$006C_H$	ADC1	8/10-bit A/D converter control register 1	R/W	00000000_B
$006D_H$	ADC2	8/10-bit A/D converter control register 2	R/W	00000000_B
$006E_H$	ADDH	8/10-bit A/D converter data register (Upper)	R/W	00000000_B
$006F_H$	ADDL	8/10-bit A/D converter data register (Lower)	R/W	00000000_B
0070_H to 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	—	Mirror of register bank pointer (RP) and 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 setting register (Upper) ch. 0	R/W	00000000_B
$0F81_H$	WRARL0	Wild register address setting register (Lower) ch. 0	R/W	00000000_B
$0F82_H$	WRDR0	Wild register data setting register ch. 0	R/W	00000000_B
$0F83_H$	WRARH1	Wild register address setting register (Upper) ch. 1	R/W	00000000_B
$0F84_H$	WRARL1	Wild register address setting register (Lower) ch. 1	R/W	00000000_B
$0F85_H$	WRDR1	Wild register data setting register ch. 1	R/W	00000000_B

Table A-1 I/O MAP (MB95200H Series) (3 / 3)

Address	Register abbreviation	Register name	R/W	Initial value
$0F86_H$	WRARH2	Wild register address setting register (Upper) ch. 2	R/W	00000000_B
$0F87_H$	WRARL2	Wild register address setting register (Lower) ch. 2	R/W	00000000_B
$0F88_H$	WRDR2	Wild register data setting register ch. 2	R/W	00000000_B
$0F89_H$ to $0F91_H$	—	(Disabled)	—	—
$0F92_H$	T01CR0	8/16-bit composite timer 01 status control register 0 ch. 0	R/W	00000000_B
$0F93_H$	T00CR0	8/16-bit composite timer 00 status control register 0 ch. 0	R/W	00000000_B
$0F94_H$	T01DR	8/16-bit composite timer 01 data register ch. 0	R/W	00000000_B
$0F95_H$	T00DR	8/16-bit composite timer 00 data register ch. 0	R/W	00000000_B
$0F96_H$	TMCR0	8/16-bit composite timer 00/01 timer mode control register ch. 0	R/W	00000000_B
$0F97_H$	T11CR0	8/16-bit composite timer 11 status control register 0 ch. 1	R/W	00000000_B
$0F98_H$	T10CR0	8/16-bit composite timer 10 status control register 0 ch. 1	R/W	00000000_B
$0F99_H$	T11DR	8/16-bit composite timer 11 data register ch. 1	R/W	00000000_B
$0F9A_H$	T10DR	8/16-bit composite timer 10 data register ch. 1	R/W	00000000_B
$0F9B_H$	TMCR1	8/16-bit composite timer 10/11 timer mode control register ch. 1	R/W	00000000_B
$0F9C_H$ 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
$0FBE_H$ to $0FC2_H$	—	(Disabled)	—	—
$0FC3_H$	AIDRL	A/D input disable register (Lower)	R/W	00000000_B
$0FC4_H$ to $0FE3_H$	—	(Disabled)	—	—
$0FE4_H$	CRTH	Main CR clock trimming register (Upper)	R/W	$1XXXXXXX_B$
$0FE5_H$	CRTL	Main CR clock trimming register (Lower)	R/W	$000XXXXX_B$
$0FE6_H$ to $0FE7_H$	—	(Disabled)	—	—
$0FE8_H$	SYSC	System configuration register	R/W	11000011_B
$0FE9_H$	CMCR	Clock monitoring control register	R/W	$XX000000_B$
$0FEA_H$	CMDR	Clock monitoring data register	R/W	00000000_B
$0FEB_H$	WDTH	Watchdog timer selection ID register (Upper)	R/W	$XXXXXXXX_B$
$0FEC_H$	WDTL	Watchdog timer selection ID register (Lower)	R/W	$XXXXXXXX_B$
$0FED_H$	—	(Disabled)	—	—
$0FEE_H$	ILSR	Input level select register	R/W	00000000_B
$0FEF_H$ 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.

Note :

Do not write to an address that is "(Disabled)". If a "(Disabled)" address is read, an undefined value is returned.

Table A-2 I/O MAP (MB95210H Series) (1 / 3)

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	—	(Disabled)	—	—
0007_H	SYCC	System clock control register	R/W	$XXXXXX11_B$
0008_H	STBC	Standby control register	R/W	$00000XXX_B$
0009_H	RSRR	Reset source register	R	$XXXXXXXX_B$
$000A_H$	TBTC	Timebase 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
$000D_H$	SYCC2	System clock control register 2	R/W	$XX100011_B$
$000E_H$ to 0015_H	—	(Disabled)	—	—
0016_H	—	(Disabled)	—	—
0017_H	—	(Disabled)	—	—
0018_H to 0027_H	—	(Disabled)	—	—
0028_H	PDRF	Port F data register	R/W	00000000_B
0029_H	DDRF	Port F direction register	R/W	00000000_B
$002A_H$	—	(Disabled)	—	—
$002B_H$	—	(Disabled)	—	—
$002C_H$	PUL0	Port 0 pull-up register	R/W	00000000_B
$002D_H$ to 0034_H	—	(Disabled)	—	—
0035_H	—	(Disabled)	—	—
0036_H	T01CR1	8/16-bit composite timer 01 status control register 1 ch. 0	R/W	00000000_B
0037_H	T00CR1	8/16-bit composite timer 00 status control register 1 ch. 0	R/W	00000000_B
0038_H	—	(Disabled)	—	—
0039_H	—	(Disabled)	—	—
$003A_H$ to 0048_H	—	(Disabled)	—	—
0049_H	—	(Disabled)	—	—
$004A_H$	EIC20	External interrupt circuit control register ch. 4	R/W	00000000_B
$004B_H$	EIC30	External interrupt circuit control register ch. 6	R/W	00000000_B
$004C_H$ to $004F_H$	—	(Disabled)	—	—
0050_H	—	(Disabled)	—	—

Table A-2 I/O MAP (MB95210H Series) (2 / 3)

Address	Register abbreviation	Register name	R/W	Initial value
0051_H	—	(Disabled)	—	—
0052_H	—	(Disabled)	—	—
0053_H	—	(Disabled)	—	—
0054_H	—	(Disabled)	—	—
0055_H	—	(Disabled)	—	—
0056_H to $006B_H$	—	(Disabled)	—	—
$006C_H$	ADC1	8/10-bit A/D converter control register 1	R/W	00000000_B
$006D_H$	ADC2	8/10-bit A/D converter control register 2	R/W	00000000_B
$006E_H$	ADDH	8/10-bit A/D converter data register (Upper)	R/W	00000000_B
$006F_H$	ADDL	8/10-bit A/D converter data register (Lower)	R/W	00000000_B
0070_H to 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	—	Mirror of register bank pointer (RP) and 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
$007B_H$	—	(Disabled)	—	—
$007C_H$	—	(Disabled)	—	—
$007D_H$	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 setting register (Upper) ch. 0	R/W	00000000_B
$0F81_H$	WRARL0	Wild register address setting register (Lower) ch. 0	R/W	00000000_B
$0F82_H$	WRDR0	Wild register data setting register ch. 0	R/W	00000000_B
$0F83_H$	WRARH1	Wild register address setting register (Upper) ch. 1	R/W	00000000_B
$0F84_H$	WRARL1	Wild register address setting register (Lower) ch. 1	R/W	00000000_B
$0F85_H$	WRDR1	Wild register data setting register ch. 1	R/W	00000000_B
$0F86_H$	WRARH2	Wild register address setting register (Upper) ch. 2	R/W	00000000_B
$0F87_H$	WRARL2	Wild register address setting register (Lower) ch. 2	R/W	00000000_B
$0F88_H$	WRDR2	Wild register data setting register ch. 2	R/W	00000000_B
$0F89_H$ to $0F91_H$	—	(Disabled)	—	—
$0F92_H$	T01CR0	8/16-bit composite timer 01 status control register 0 ch. 0	R/W	00000000_B
$0F93_H$	T00CR0	8/16-bit composite timer 00 status control register 0 ch. 0	R/W	00000000_B
$0F94_H$	T01DR	8/16-bit composite timer 01 data register ch. 0	R/W	00000000_B
$0F95_H$	T00DR	8/16-bit composite timer 00 data register ch. 0	R/W	00000000_B

Table A-2 I/O MAP (MB95210H Series) (3 / 3)

Address	Register abbreviation	Register name	R/W	Initial value
$0F96_H$	TMCR0	8/16-bit composite timer 00/01 timer mode control register ch. 0	R/W	00000000_B
$0F97_H$	—	(Disabled)	—	—
$0F98_H$	—	(Disabled)	—	—
$0F99_H$	—	(Disabled)	—	—
$0F9A_H$	—	(Disabled)	—	—
$0F9B_H$	—	(Disabled)	—	—
$0F9C_H$ to $0FBB_H$	—	(Disabled)	—	—
$0FBC_H$	—	(Disabled)	—	—
$0FBD_H$	—	(Disabled)	—	—
$0FBE_H$ to $0FC2_H$	—	(Disabled)	—	—
$0FC3_H$	AIDRL	A/D input disable register (Lower)	R/W	00000000_B
$0FC4_H$ to $0FE3_H$	—	(Disabled)	—	—
$0FE4_H$	CRTH	Main CR clock trimming register (Upper)	R/W	$1XXXXXXX_B$
$0FE5_H$	CRTL	Main CR clock trimming register (Lower)	R/W	$000XXXXX_B$
$0FE6_H$ to $0FE7_H$	—	(Disabled)	—	—
$0FE8_H$	SYSC	System configuration register	R/W	11000011_B
$0FE9_H$	CMCR	Clock monitoring control register	R/W	$XX000000_B$
$0FEA_H$	CMDR	Clock monitoring data register	R/W	00000000_B
$0FEB_H$	WDTH	Watchdog timer selection ID register (Upper)	R/W	$XXXXXXXX_B$
$0FEC_H$	WDTL	Watchdog timer selection ID register (Lower)	R/W	$XXXXXXXX_B$
$0FED_H$	—	(Disabled)	—	—
$0FEE_H$	ILSR	Input level select register	R/W	00000000_B
$0FEF_H$ 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.

Note :

Do not write to an address that is "(Disabled)". If a "(Disabled)" address is read, an undefined value is returned.

APPENDIX B Table of Interrupt Sources

This section shows the table of interrupt sources used in the MB95200H/210H Series.

■ Table of Interrupt Sources

See "CHAPTER 5 CPU" for interrupt operation.

Table B-1 MB95200H Series

Interrupt source	Interrupt request number	Vector table address		Bit name of interrupt level setting register	Priority order of interrupt sources of the same level (occurring simultaneously)
		Upper	Lower		
External interrupt ch.4	IRQ0	$FFFA_H$	$FFFB_H$	L00 [1 : 0]	High
External interrupt ch.5	IRQ1	$FFF8_H$	$FFF9_H$	L01 [1 : 0]	↑
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					↑
—	IRQ4	$FFF2_H$	$FFF3_H$	L04 [1 : 0]	↑
8/16-bit composite timer ch.0 (Lower)	IRQ5	$FFF0_H$	$FFF1_H$	L05 [1 : 0]	↑
8/16-bit composite 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]	↑
—	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]	↑
8/16-bit composite timer ch.1 (Upper)	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]	↑
8/10-bit A/D converter	IRQ18	$FFD6_H$	$FFD7_H$	L18 [1 : 0]	↑
Timebase timer	IRQ19	$FFD4_H$	$FFD5_H$	L19 [1 : 0]	↑
Watch prescaler	IRQ20	$FFD2_H$	$FFD3_H$	L20 [1 : 0]	↑
—	IRQ21	$FFD0_H$	$FFD1_H$	L21 [1 : 0]	↑
8/16-bit composite timer ch.1 (Lower)	IRQ22	$FFCE_H$	$FFCF_H$	L22 [1 : 0]	↑
Flash memory	IRQ23	$FFCC_H$	$FFCD_H$	L23 [1 : 0]	Low

Table B-2 MB95210H Series

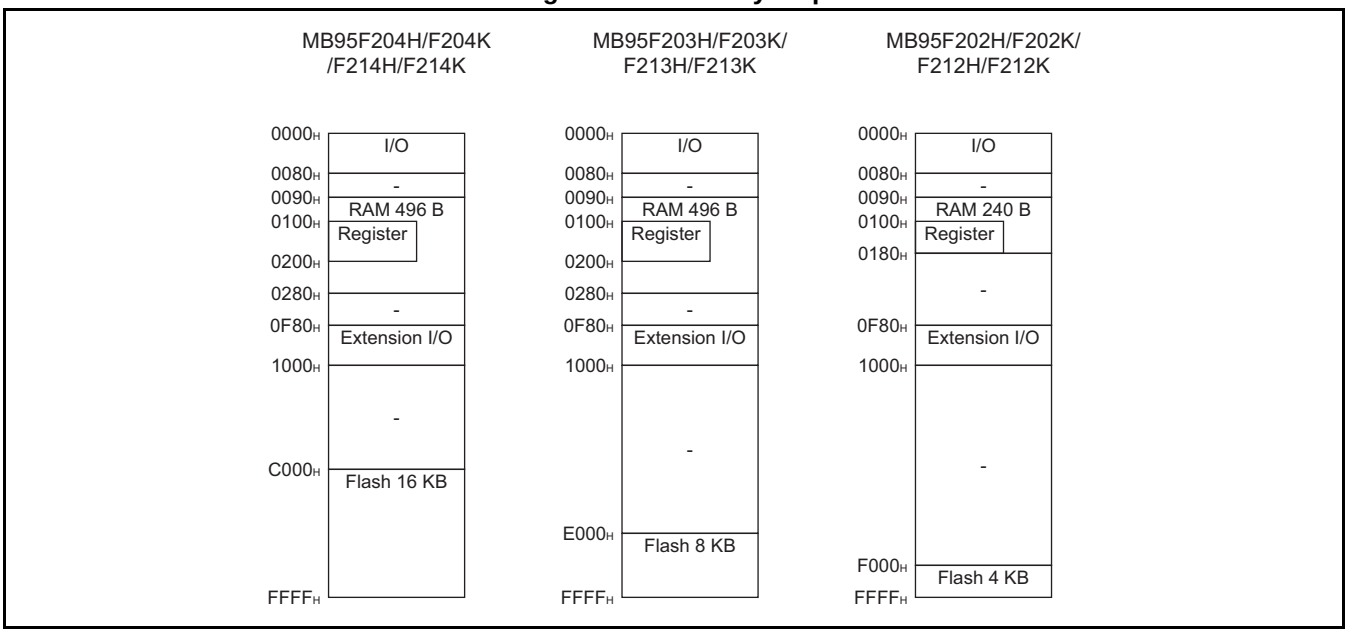
Interrupt source	Interrupt request number	Vector table address		Bit name of interrupt level setting register	Priority order of interrupt sources of the same level (occurring simultaneously)
		Upper	Lower		
External interrupt ch. 4	IRQ0	$FFFA_H$	$FFFB_H$	L00 [1 : 0]	High
—	IRQ1	$FFF8_H$	$FFF9_H$	L01 [1 : 0]	
—	IRQ2	$FFF6_H$	$FFF7_H$	L02 [1 : 0]	
External interrupt ch. 6					
—	IRQ3	$FFF4_H$	$FFF5_H$	L03 [1 : 0]	
—					
—	IRQ4	$FFF2_H$	$FFF3_H$	L04 [1 : 0]	
8/16-bit composite timer ch. 0 (Lower)	IRQ5	$FFF0_H$	$FFF1_H$	L05 [1 : 0]	
8/16-bit composite timer ch. 0 (Upper)	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]	
8/10-bit A/D converter	IRQ18	$FFD6_H$	$FFD7_H$	L18 [1 : 0]	
Timebase timer	IRQ19	$FFD4_H$	$FFD5_H$	L19 [1 : 0]	
Watch prescaler	IRQ20	$FFD2_H$	$FFD3_H$	L20 [1 : 0]	
—	IRQ21	$FFD0_H$	$FFD1_H$	L21 [1 : 0]	
—	IRQ22	$FFCE_H$	$FFCF_H$	L22 [1 : 0]	
Flash memory	IRQ23	$FFCC_H$	$FFCD_H$	L23 [1 : 0]	Low

APPENDIX C Memory Map

This section shows the memory map of the MB95200H/210H Series.

■ Memory Map

Figure C-1 Memory Map



	Flash memory	RAM
MB95F204H/F204K/F214H/F214K	16 KB	496 B
MB95F203H/F203K/F213H/F213K	8 KB	496 B
MB95F202H/F202K/F212H/F212K	4 KB	240 B

APPENDIX D Pin States of MB95200H/210H Series

Table D-1 belows shows the pin states of the MB95200H/210H Series in each mode.

■ Pin States in Each Mode

Table D-1 Pin States in Each Mode (1 / 2)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		In reset
			SPL=0	SPL=1	SPL=0	SPL=1	
PF0/X0	Oscillation circuit input	Oscillation circuit input	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Oscillation circuit input*3
PF1/X1	Oscillation circuit output	Oscillation circuit input	"H"	"H"	"H"	"H"	Oscillation circuit output*3
PG0/X0A	Oscillation circuit input	Oscillation circuit input	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Oscillation circuit input*5
PG1/X1A	Oscillation circuit input	Oscillation circuit input	"H"	"H"	"H"	"H"	Oscillation circuit input*5
PF2/$\overline{\text{RSTX}}$	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input*4
P00/AN00	I/O port/ peripheral function I/O/ Analog input	I/O port/ peripheral function I/O/ Analog input	I/O port/ peripheral function I/O/ Analog input	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 enabled.)	I/O port/ peripheral function I/O/ Analog input	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 enabled.)	Hi-Z Input disabled*2
P01/AN01							
P02/ INT02/ AN02/SCK							
P03/ INT03/ AN03/SOT							
P04/ INT04/ AN04/SIN/ HCLK1/ EC0							
P05/ INT05/ AN05/ TO00/ HCLK2							
P06/ INT06/ TO01	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O		I/O port/ peripheral function I/O		
P07/INT07							

Table D-1 Pin States in Each Mode (2 / 2)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		In reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P12/EC0/ DBG	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	"H"	I/O port/ peripheral function I/O	"H"	"H"
P62/TO10				Hi-Z (However, the setting of the pull-up is effective.) Input cutoff		Hi-Z (However, the setting of the pull-up is effective.) Input cutoff	Hi-Z Input enabled*1 (However, it does not function.)
P63/TO11							
P64/EC1							
PF0/X0	I/O port	I/O port	I/O port	Hi-Z Input cutoff	I/O port	Hi-Z Input cutoff	Hi-Z Input enabled*1*6 (However, it does not function.)

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

Hi-Z: High impedance

*1: "Input enabled" means that the input function is enabled. While the input function is enabled, pull-up or pull-down operation has to be performed in order to prevent leaks due to external input. If a pin is used as an output port, the pin state is the same as that of other ports.

*2: "Input disabled" means direct input gate operation from the pin is disabled.

*3: The pin state when PF0/X0 and PF1/X1 are configured as main osc pins

*4: The pin state when PF2/RSTX is configured as reset pin

*5: The pin state when PG0/X0A and PG1/X1A are configured as sub-osc pins

*6: The pin state when these pins are configured as GPIOs

APPENDIX E Instruction Overview

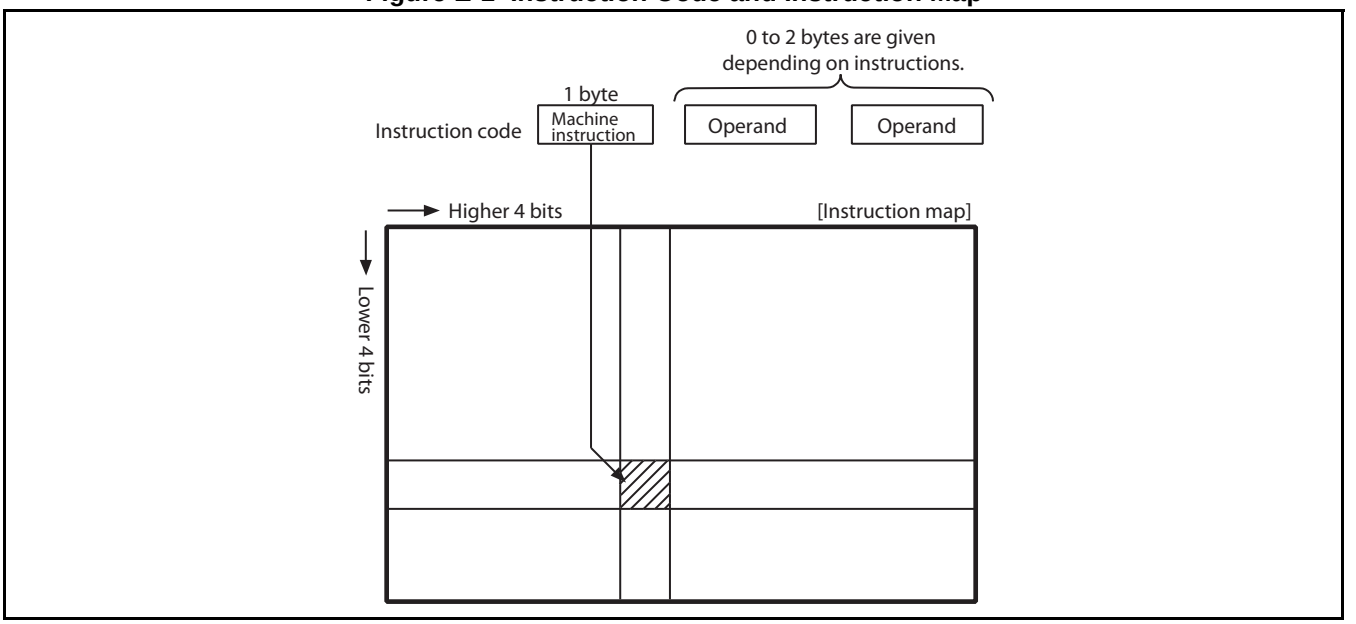
This section shows instructions used in F^{2}MC-8FX.

■ 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 the following four types: transfer, arithmetic operation, branch and other.
- There are various methods of addressing, and ten types of addressing can be selected according to the combination of the instruction and the operand selected.
- It has bit operation instructions, and can operate the read-modify-write operation.
- There is an instruction that orders special operation.

■ Meanings of Symbols Used in Instruction Tables

Table E-1 shows the meanings of symbols used in instruction tables in APPENDIX E.

Table E-1 Meaning of Symbols Used in Instruction Tables

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.)

■ Meanings of Items in Instruction Tables

Table E-2 Meanings of Items in Instruction Tables

Item	Meaning
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.

E.1 Addressing

F²MC-8FX has the following ten types of addressing:

- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

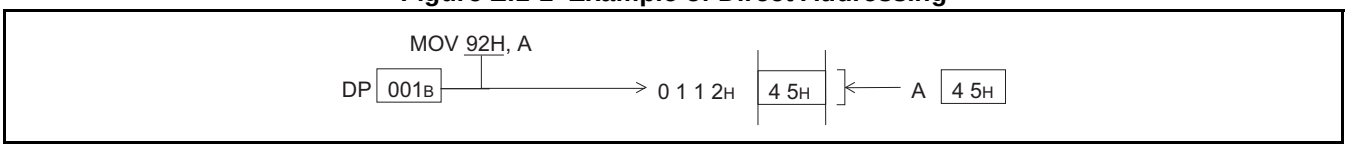
■ Explanation of Addressing

● 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

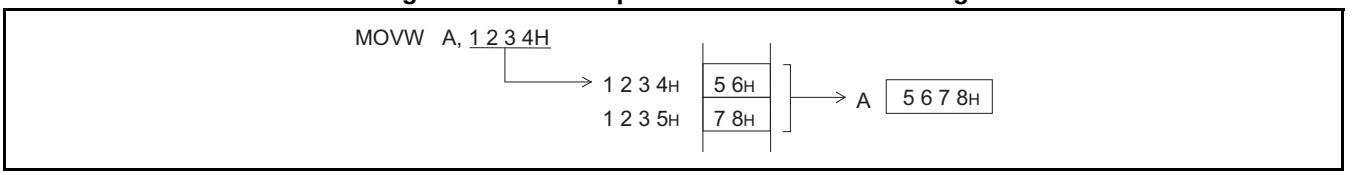


● Extended addressing

This is used when the area of the entire 64 KB 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

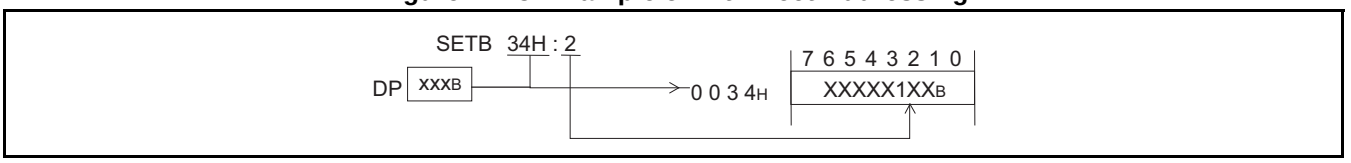


● Bit direct addressing

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

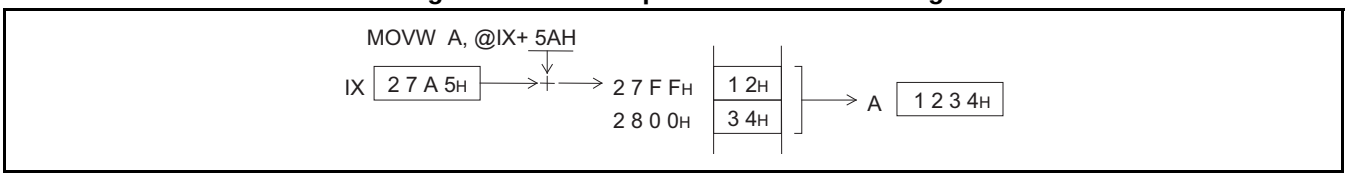


● Index addressing

This is used when the area of the entire 64 KB 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

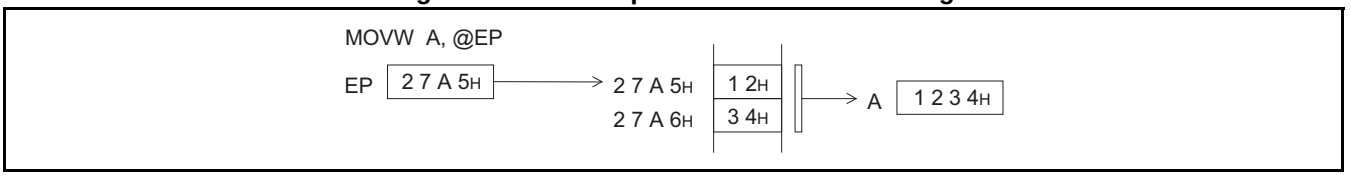


● Pointer addressing

This is used when the area of the entire 64 K bytes 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

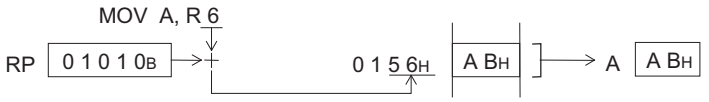


● General-purpose register addressing

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.

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

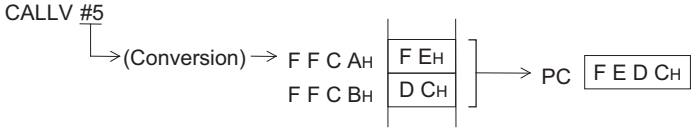
This is used when branching to the subroutine address registered in the table with the addressing shown "#vct" in the instruction table. In this addressing, information on "#vct" is contained in the operation code, and the address of the table is created using the combinations shown in Table E.1-1.

Table E.1-1 Vector Table Address Corresponding to Respective "#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

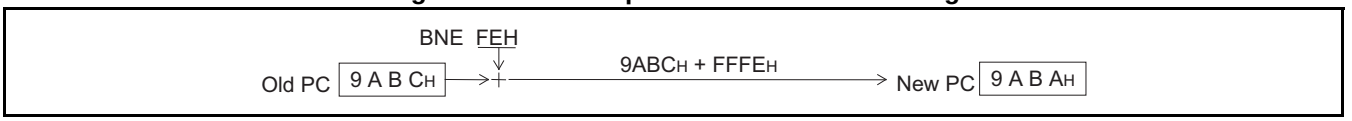


● Relative addressing

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



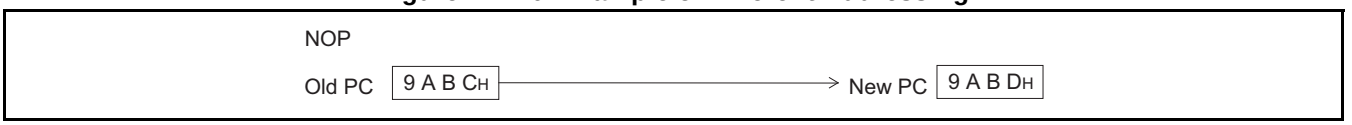
In this example, by jumping to the address where the operation code of BNE is stored, it results in an infinite loop.

● 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



E.2 Special Instruction

This section explains special instructions other than the addressing.

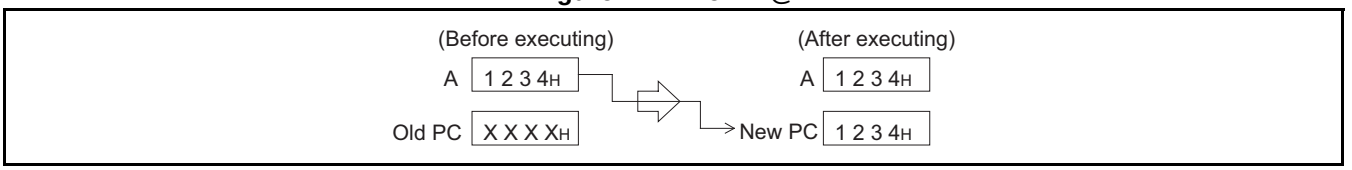
■ Special Instruction

● 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

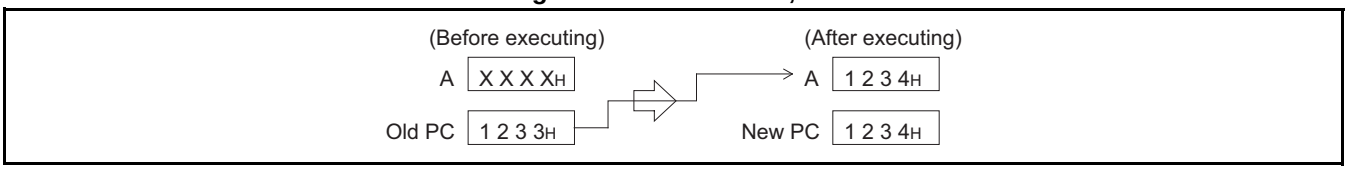


● MOVW A, PC

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



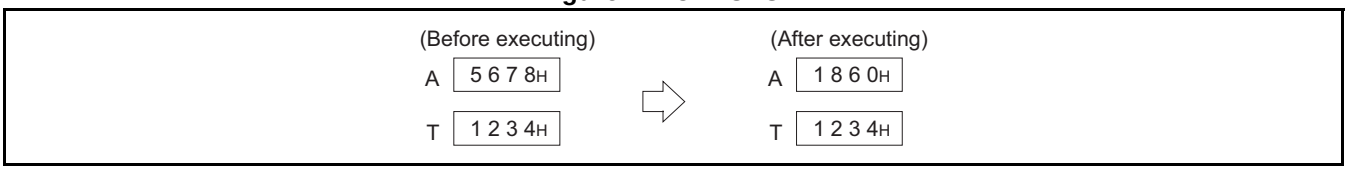
When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-2, the value "1234_H" stored in A corresponds to the address where 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.

Figure E.2-3 shows a summary of the instruction.

Figure E.2-3 MULU A

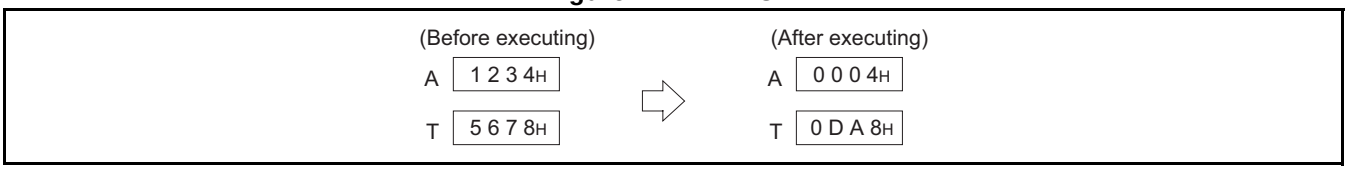


● DIVU A

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

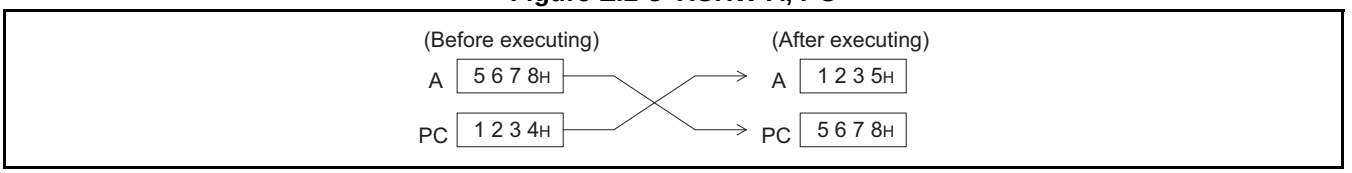


● XCHW A, PC

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.

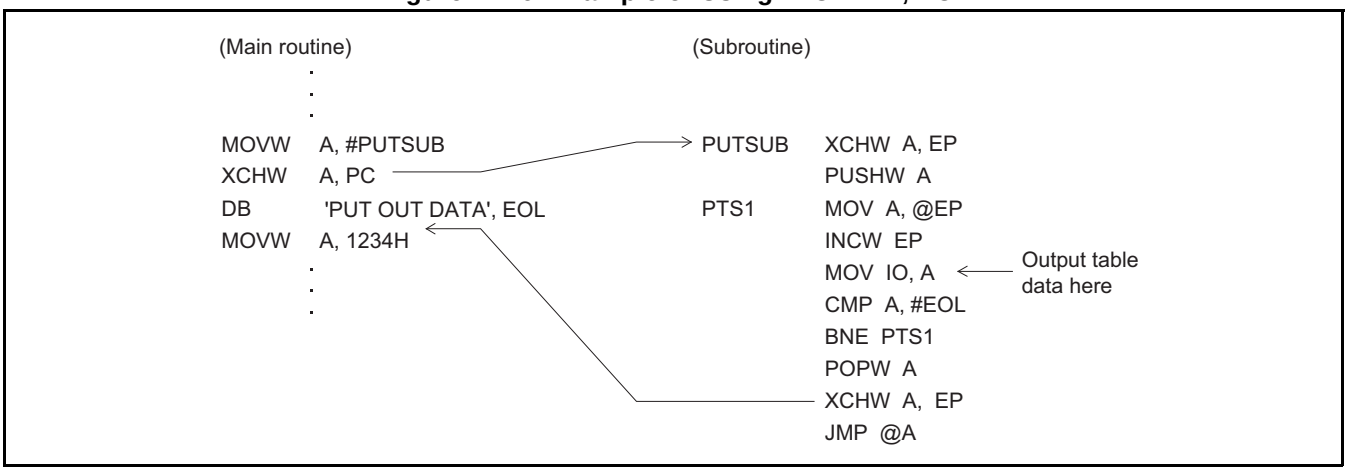
Figure E.2-5 XCHW A, PC



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".

Figure E.2-6 shows an assembler language example.

Figure E.2-6 Example of Using "XCHW A, PC"

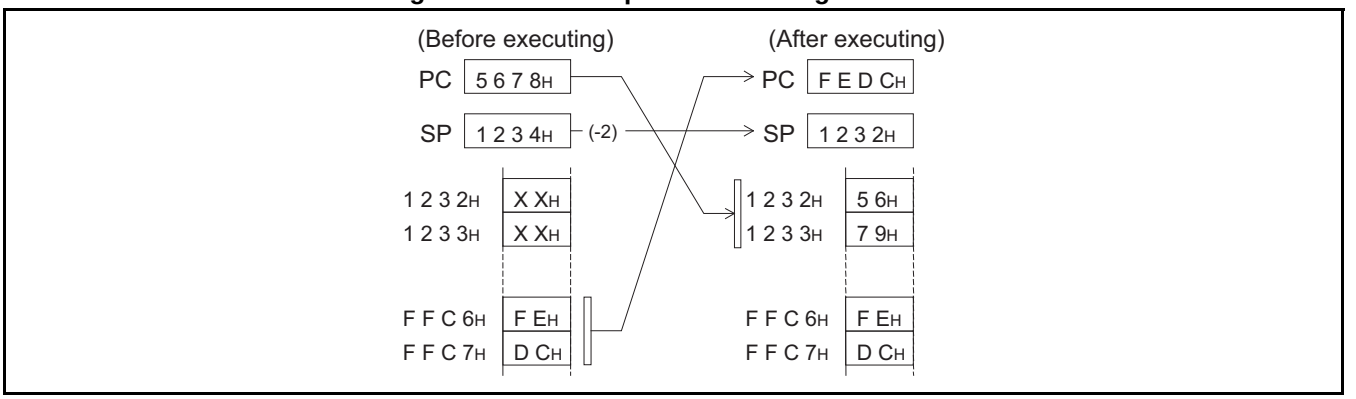


● CALLV #vct

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).

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	$FFC0_H$	$FFC1_H$

E.3 Bit Manipulation Instructions (SETB, CLRB)

Some peripheral function registers include bits whose read values during a normal read operation are different those in an operation of the bit manipulation instruction.

■ Read-modify-write Operation

With the bit manipulation instructions (SETB and CLRB), specific bits in a register or RAM can be set to ’1" (by SETB) and cleared to "0" (by CLRB). In actual operation, since the CPU handles data by the eight bits, it performs a sequence of operations (read-modify-write operation) in a bit manipulation instruction. In the read-modify-write operation, the CPU reads 8-bit data, modifies a bit specified, and writes the data it has read back to its original address.

Table E.3-1 shows bus operation for bit manipulation instructions.

Table E.3-1 Bus Operation for Bit Manipulation Instructions

CODE	MNENONIC	~	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 in Execution of Bit Manipulation Instructions

For some I/O ports and interrupt request flag bits, the read destination in a normal read operation differs from the one in a read-modify-write operation.

- I/O ports (bit manipulation instruction)

From some I/O ports, while an I/O pin value is read during a normal read operation, a port data register value is read during bit manipulation. This mechanism prevents other port data register bits from being modified accidentally, regardless of the I/O directions and states of pins.

- Interrupt request flag bits (bit manipulation instruction)

During a normal read operation, an interrupt request flag bit functions as a flag bit indicating whether an interrupt request occurs. During bit manipulation, the bit always returns "1". This mechanism prevents the interrupt request flag bit from being cleared accidentally by writing "0" to it during the manipulation of another bit.

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

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 Instructions (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	decimaladjustforaddition	-	-	-	+	+	+	+	84
38	DAS		1	1	decimaladjustforsubtraction	-	-	-	+	+	+	+	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
46	AND	A, #d8	2	2	(A) ← (AL) ∧ d8	-	-	-	+	+	R	-	64
47	AND	A, dir	3	2	(A) ← (AL) ∧ (dir)	-	-	-	+	+	R	-	65

Table E.4-2 Arithmetic Operation Instructions (2 / 2)

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
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	ifZ = 1thenPC←PC + rel	-	-	-	-	-	-	-	FD
	BZ/BEQ	rel(at no branch)	2										
2	BNZ/BNE	rel(at branch)	4	2	ifZ = 0thenPC←PC + rel	-	-	-	-	-	-	-	FC
	BNZ/BNE	rel(at no branch)	2										
3	BC/BLO	rel(at branch)	4	2	ifC = 1thenPC←PC + rel	-	-	-	-	-	-	-	F9
	BC/BLO	rel(at no branch)	2										
4	BNC/BHS	rel(at branch)	4	2	ifC = 0thenPC←PC + rel	-	-	-	-	-	-	-	F8
	BNC/BHS	rel(at no branch)	2										
5	BN	rel(at branch)	4	2	ifN = 1thenPC←PC + rel	-	-	-	-	-	-	-	FB
	BN	rel(at no branch)	2										
6	BP	rel(at branch)	4	2	ifN = 0thenPC←PC + rel	-	-	-	-	-	-	-	FA
	BP	rel(at no branch)	2										
7	BLT	rel(at branch)	4	2	ifV∀N = 1thenPC←PC + rel	-	-	-	-	-	-	-	FF
	BLT	rel(at no branch)	2										
8	BGE	rel(at branch)	4	2	ifV∀N = 0thenPC←PC + rel	-	-	-	-	-	-	-	FE
	BGE	rel(at no branch)	2										
9	BBC	dir : b, rel	5	3	if (dir : b) = 0thenPC←PC + rel	-	-	-	-	+	-	-	B0 to B7
10	BBS	dir : b, rel	5	3	if (dir : b) = 1thenPC←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	vectorcall	-	-	-	-	-	-	-	E8 to EF
14	CALL	ext	6	3	subroutinecall	-	-	-	-	-	-	-	31
15	XCHW	A, PC	3	1	(PC) ← (A) , (A) ← (PC) + 1	-	-	dH	-	-	-	-	F4
16	RET		6	1	returnfromsubroutine	-	-	-	-	-	-	-	20
17	RETI		8	1	returnfrominterrupt	-	-	-		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

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

H / L	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 Options

The mask option list of the MB95200H/210H Series is shown in Table F-1.

■ Mask Option List

Table F-1 Mask Option List

No.	Part Number	MB95F204H MB95F203H MB95F202H MB95F214H MB95F213H MB95F212H	MB95F204K MB95F203K MB95F202K MB95F214K MB95F213K MB95F212K
	Selectable/Fixed	Fixed	Fixed
1	Low voltage detection reset • With low voltage detection reset • Without low voltage detection reset	Without low-voltage detection reset	With low-voltage detection reset
2	Reset • With dedicated reset input • Without dedicated reset input	With dedicated reset input	Without dedicated reset input

INDEX

The index starts from the next page.

It is arranged in alphabetical order.

Index

D

E

F

G

H

I

L

R

S

T

V

W

Register Index

W

Pin Function Index

Interrupt Vector Index

I

CM26-10120-1E

FUJITSU MICROELETRONICS • CONTROLLER MANUAL

F^2MC-8FX

8-BIT MICROCONTROLLER

MB95200H/210H Series

HARDWARE MANUAL

April 2008 The first edition

Published **FUJITSU MICROELECTRONICS LIMITED**

Edited Strategic Business Development Dept.