FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM26-101xx-1E

New 8FX

8-BIT MICROCONTROLLER

MB95430H Series
HARDWARE MANUAL

New 8FX

8-BIT MICROCONTROLLER

MB95430H Series

HARDWARE MANUAL

For the information for microcontroller supports, see the following website.

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

FUJITSU SEMICONDUCTOR LIMITED

PREFACE

■ The Purpose and Intended Readership of This Manual

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

The MB95430H Series is a line of products developed as general-purpose products in the New 8FX family of proprietary 8-bit single-chip microcontrollers applicable as application-specific integrated circuits (ASICs). The MB95430H 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 MB95430H 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^{2}MC-8FX Programming Manual".

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

■ Trademark

The company names and brand names in this document are the trademarks or registered trademarks of their respective owners.

■ Sample Programs

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

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

CONTENTS

Major revisions in this edition

Page	Revisions (For details, see their respective pages.)
-	First edition

CHAPTER 1
OVERVIEW

This chapter describes the features and basic specifications of the MB95430H Series.

1.1 Features of MB95430H Series

In addition to a compact instruction set, MB95430H is a series of general-purpose single-chip microcontrollers with a variety of peripheral functions.

■ Features of MB95430H Series

● F^{2}MC-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 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-CR clock (Typ: 100 kHz, Min: 50 kHz, Max: 200 kHz)

● Timer

- 8/16-bit composite timer × 1 channel
- 16-bit PPG × 1 channel
- 16-bit free-running timer × 1 channel
- 16-bit output compare × 2 channels
- Time-base timer × 1 channel
- Watch prescaler × 1 channel

● UART/SIO

- Full duplex double buffer
- Capable of clock-asynchronous (UART) serial data transfer and clock-synchronous (SIO) serial data transfer

● I^{2}C

- Built-in wake-up function

● Voltage comparator

● Operational amplifier (OPAMP)

- Software-select programmable gain
- Software-select standalone option
- Power down function included

● 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
- Time-base timer mode

● I/O port

- MB95F432H/F433H/F434H (maximum no. of I/O ports: 28)
  - General-purpose I/O ports (N-ch open drain) : 1
  - General-purpose I/O ports (CMOS I/O) : 27
- MB95F432K/F433K/F434K (maximum no. of I/O ports: 29)
  - General-purpose I/O ports (N-ch open drain) : 2
  - General-purpose I/O ports (CMOS I/O) : 27

● On-chip debug

- 1-wire serial control
- Serial writing supported (asynchronous mode)

● Hardware/software watchdog timer

- Built-in hardware watchdog timer
- Built-in software watchdog timer

● Low-voltage detection reset circuit

- Built-in low-voltage detector

● Programmable port input voltage level

- CMOS input level / hysteresis input level

● Dual operation Flash memory

- The erase/write operation and the read operation can be executed in different banks (upper bank/lower bank) simultaneously.

● Flash memory security function

- Protects the content of the Flash memory

MB95430H Series

1.2 Product Line-up of MB95430H Series

Table 1.2-1 lists the product line-up of the MB95430H Series.

■ Product Line-up of MB95430H Series

Table 1.2-1 Product Line-up of MB95430H Series (1 / 2)

Part Number Parameter	MB95F432H	MB95F433H	MB95F434H	MB95F432K	MB95F433K	MB95F434K
Type	Flash memory product					
Clock supervisor counter	It supervises the main clock oscillation.					
Program ROM capacity	8 Kbyte	12 Kbyte	20 Kbyte	8Kbyte	12 Kbyte	20 Kbyte
RAM capacity	240 bytes	240 bytes	496 bytes	240 bytes	240 bytes	496 bytes
Low-voltage detection reset	No			Yes		
Reset input	Dedicated			Selected through software		
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 frequency = 16.25 MHz) • Interrupt processing time : 0.6 µs (with machine clock frequency = 16.25 MHz)					
General-purpose I/O	• I/O ports (Max) : 28 • CMOS I/O : 27 • N-ch open drain : 1			• I/O ports (Max) : 29 • CMOS I/O : 27 • N-ch open drain : 2		
Time-base timer	Interval time: 0.256 ms to 8.3 s (with external clock frequency = 4 MHz)					
Hardware/ software watchdog timer	• Reset generation cycle - Main oscillation clock at 10 MHz: 105 ms (Min) • The sub-CR clock can be used as the source clock of hardware watchdog timer.					
Wild register	It can be used to replace three bytes of data.					
8/10-bit A/D converter	16 channels (The 16th channel is for OPAMP output.)					
	8-bit or 10-bit resolution can be selected.					
8/16-bit composite timer	1 channel					
	• The timer can be configured as an "8-bit timer × 2 channels" or a "16-bit timer × 1 channel". • It has built-in timer function, PWC function, PWM function and capture function. • Count clock: it can be selected from internal clocks (seven types) and external clocks. • It can output square wave.					
External interrupt	8 channels					
	• Interrupt by edge detection (The rising edge, falling edge, or both edges can be selected.) • It can be used to wake up the device from the standby mode.					
On-chip debug	• 1-wire serial control • It supports serial writing (asynchronous mode).					

Table 1.2-1 Product Line-up of MB95430H Series (2 / 2)

Part Number Parameter	MB95F432H	MB95F433H	MB95F434H	MB95F432K	MB95F433K	MB95F434K
UART/SIO	1 channel					
	• Data transfer with UART/SIO is enabled. • It has a full duplex double buffer, variable data length (5/6/7/8 bits), a built-in baud rate generator and an error detection function. • It uses the NRZ type transfer format. • LSB-first data transfer and MSB-first data transfer are available to use. • Clock-asynchronous (UART) serial data transfer and clock-synchronous (SIO) serial data transfer is enabled.					
I2C	1 channel					
	• Master/slave transmission and receiving • It has a bus error function, an arbitration function, a transmission direction detection function and a wake-up function. • It also has functions of generating and detecting repeated START conditions.					
16-bit PPG	• PWM mode and one-shot mode are available to use. • The counter operating clock can be selected from eight clock sources. • It supports external trigger start.					
Output compare	• 1 channel of 16-bit free-running timer with a compare buffer • 2 channels of 16-bit output compare					
Voltage comparator	4 channels					
OPAMP	• This is an operational amplifier used in an induction heater. It contains 7 software (registers) select close loop gain selections for ground current sensing according to different sense resistor values. The OPAMP can also work as a standalone OPAMP. • It selects closed loop gain for ground current sensing according to different sense resistor values of a standalone OPAMP.					
Watch prescaler	Eight different time intervals can be selected.					
Flash memory	• It supports automatic programming, Embedded Algorithm, write/erase/erase-suspend/erase-resume commands. • It has a flag indicating the completion of the operation of Embedded Algorithm. • Number of erase/write cycles: 100000 • Data retention time: 20 years • Flash security feature for protecting the content of the Flash memory					
Standby mode	Sleep mode, stop mode, watch mode, time-base timer mode					
Package	FPT-32P-M30 DIP-32P-M06					

1.3 Differences among Products and Notes on Product Selection

The following describes differences among the products of the MB95430H Series and notes on product selection.

■ Differences among Products and Notes on Product Selection

- Current consumption

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

  For details of current consumption, refer to "■ ELECTRICAL CHARACTERISTICS" in the data sheet of the MB95430H Series.

- Package

  For details of information on each package, see "1.6 Package Dimension".

- 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 MB95430H Series.

- On-chip debug function

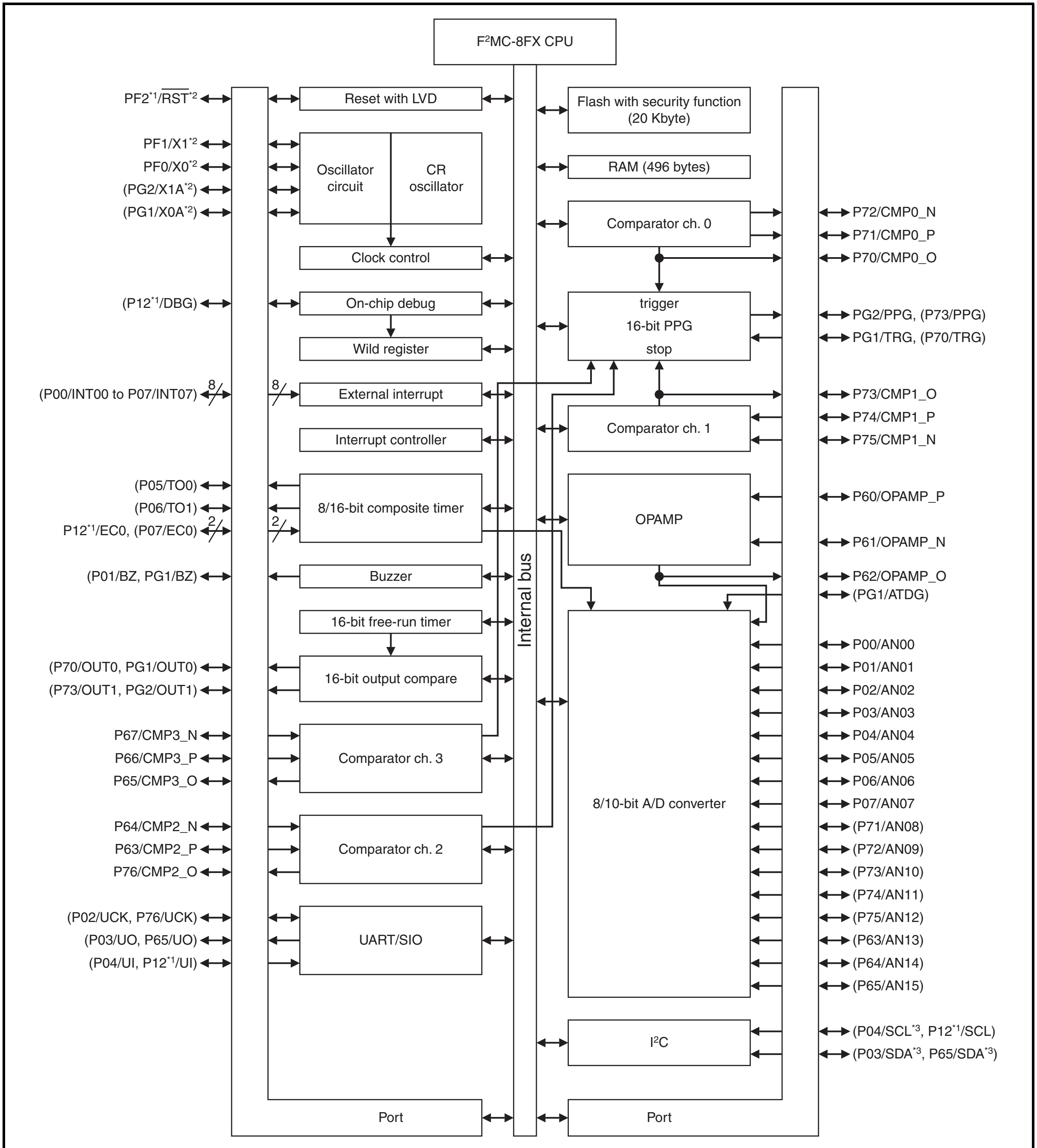
  The on-chip debug function requires that V_{CC}, V_{SS} and 1 serial-wire be connected to an evaluation tool. For details of the connection method, see "CHAPTER 29 EXAMPLE OF SERIAL PROGRAMMING CONNECTION".

1.4 Block Diagram of MB95430H Series

Figure 1.4-1 shows the block diagram of the MB95430H Series.

■ Block Diagram of MB95430H Series

Figure 1.4-1 Block Diagram of MB95430H Series



*1: PF2 and P12 are N-ch open drain pins.

*2: Software option

*3: This pin will work as an N-ch open drain pin during I²C operation.

Note: Pins in parentheses indicate that functions of those pins are shared among different resources.

MB95430H Series

1.5 Pin Assignment

Figure 1.5-1 and Figure 1.5-2 show the pin assignment of the MB95430H Series.

■ Pin Assignment of MB95430H Series

Figure 1.5-1 Pin Assignment of FPT-32P-M30

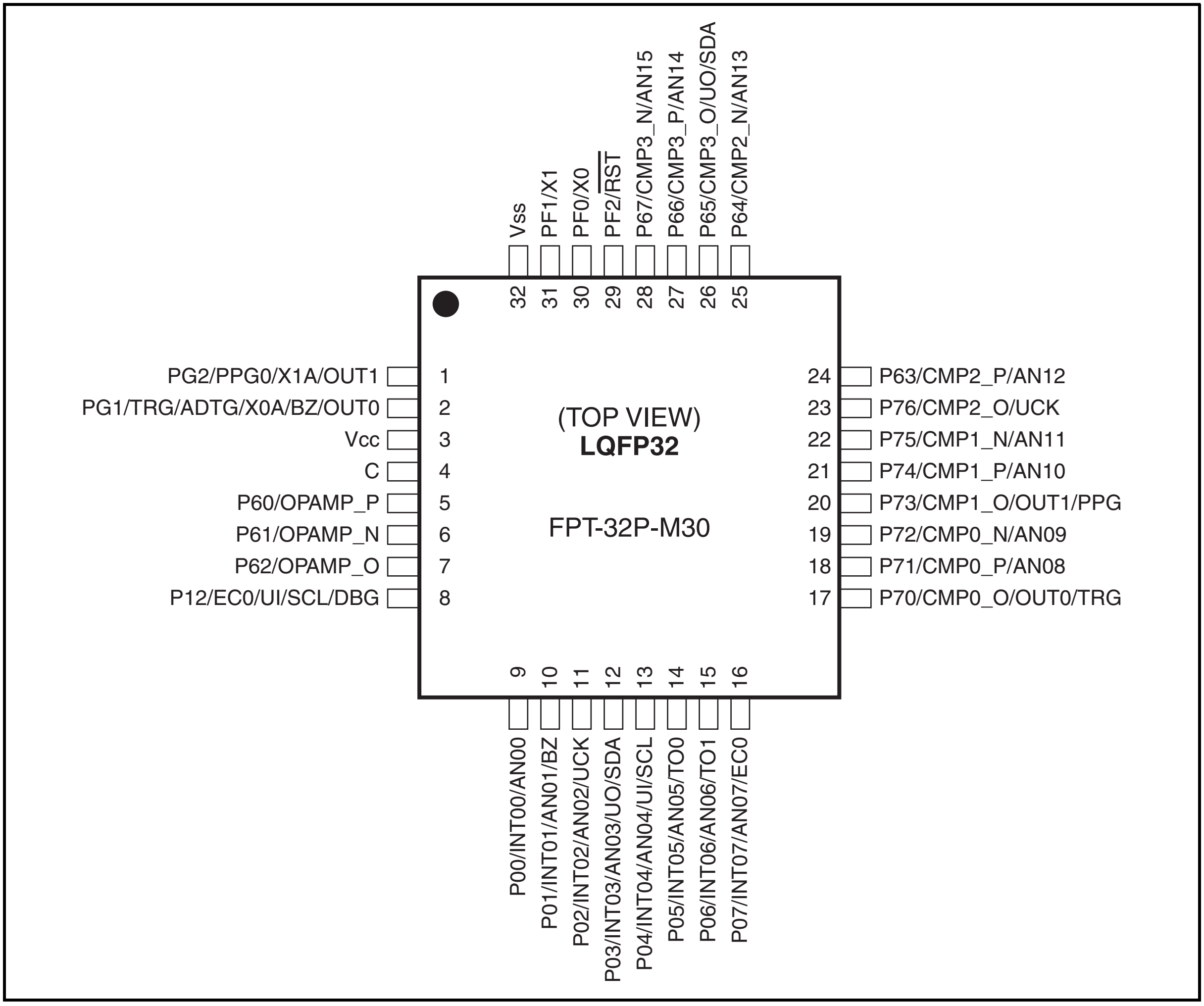
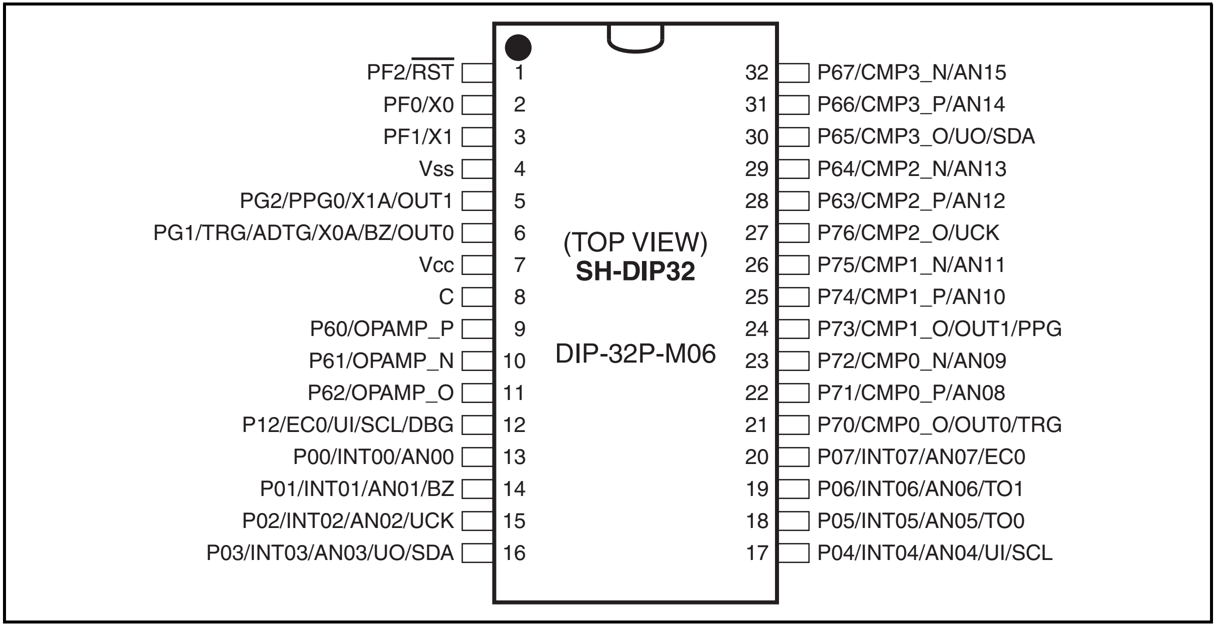


Figure 1.5-2 Pin Assignment of DIP-32P-M06



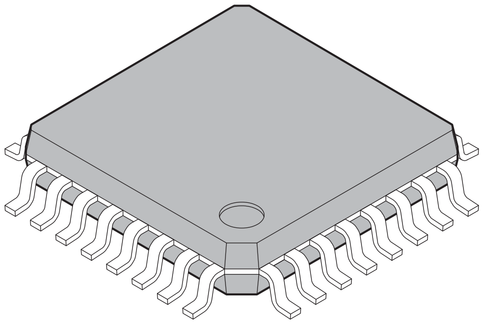
MB95430H Series

1.6 Package Dimension

The MB95430H Series is available in two types of package.

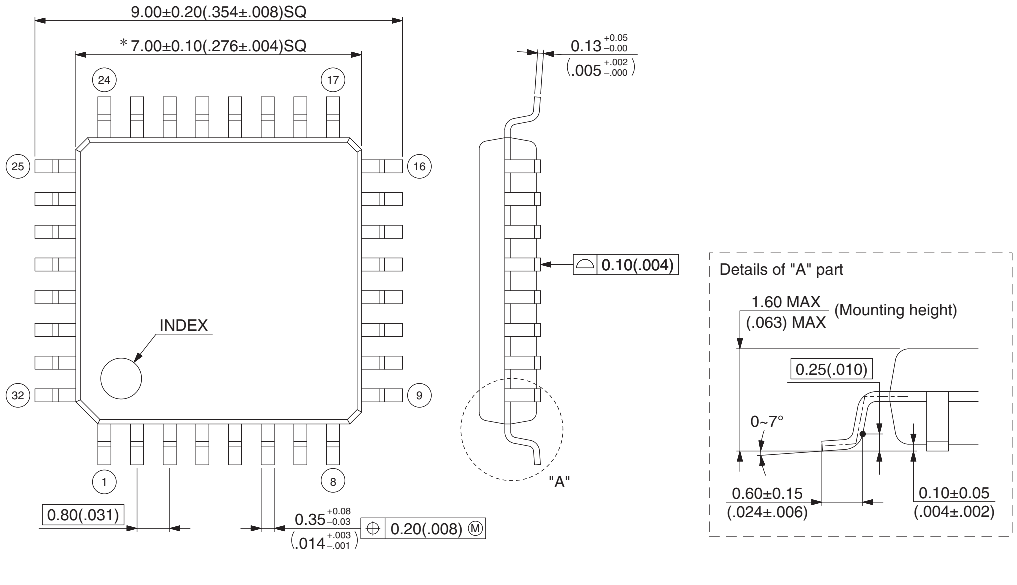
■ Package Dimension of FPT-32P-M30

Figure 1.6-1 Package Dimension of FPT32P-M30

32-pin plastic LQFP  (FPT-32P-M30)	Lead pitch	0.80 mm
	Package width × package length	7.00 mm × 7.00 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.60 mm MAX

32-pin plastic LQFP
(FPT-32P-M30)

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



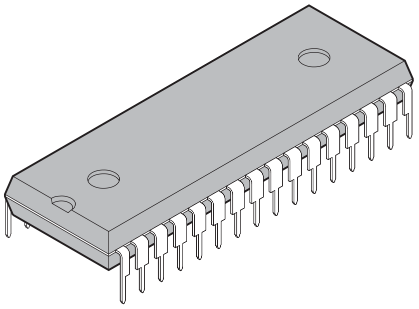
Dimensions in mm (inches).
Note: The values in parentheses are reference values.

Please check the latest package dimension at the following URL.

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

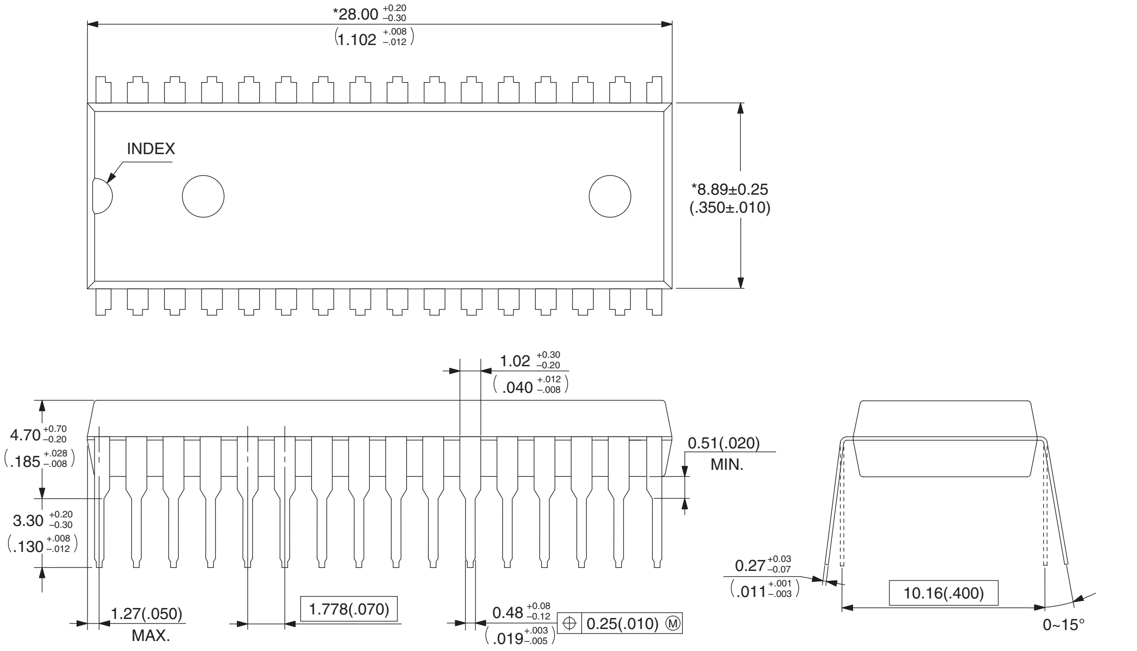
■ Package Dimension of DIP-32P-M06

Figure 1.6-2 Package Dimension of DIP-32P-M06

32-pin plastic SDIP  (DIP-32P-M06)	Lead pitch	1.778 mm
	Low space	10.16 mm
	Sealing method	Plastic mold

32-pin plastic SDIP
(DIP-32P-M06)

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



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

Please check the latest package dimension at the following URL.

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

MB95430H Series

1.7 Pin Description

Table 1.7-1 shows pin description of the MB95430H Series. The alphabets in "I/O circuit type" column of the above tables correspond to those in "Type" column of Table 1.8-1.

■ Pin Description

Table 1.7-1 Pin Description (1 / 4)

Pin no.		Pin name	I/O circuit type[*3]	Function
LQFP32[*1]	SH-DIP32[*2]			
1	5	PG2	C	General-purpose I/O port
		PPG		16-bit PPG output pin
		X1A		Subclock I/O oscillation pin
		OUT1		Output compare ch. 1 output pin
2	6	PG1	C	General-purpose I/O port
		TRG		16-bit PPG trigger input pin
		ADTG		A/D converter trigger input pin
		X0A		Subclock I/O oscillation pin
		BZ		Buzzer output pin
		OUT0		Output compare ch. 0 output pin
3	7	V_{CC}	—	Power supply pin
4	8	C	—	Capacitor connection pin
5	9	P60	K	General-purpose I/O port
		OPAMP_P		Operational amplifier input pin
6	10	P61	K	General-purpose I/O port
		OPAMP_N		Operational amplifier input pin
7	11	P60	J	General-purpose I/O port
		OPAMP_O		Operational amplifier output pin
8	12	P12	H	General-purpose I/O port
		EC0		8/16-bit composite timer external clock input pin
		UI		UART/SIO data input pin
		SCL		I^2C clock I/O pin
		DBG		DBG input pin

Table 1.7-1 Pin Description (2 / 4)

Pin no.		Pin name	I/O circuit type[*3]	Function
LQFP32[*1]	SH-DIP32[*2]			
9	13	P00	E	General-purpose I/O port
		INT00		External interrupt input pin
		AN00		A/D converter analog input pin
10	14	P01	E	General-purpose I/O port
		INT01		External interrupt input pin
		AN01		A/D converter analog input pin
		BZ		Buzzer output pin
11	15	P02	E	General-purpose I/O port
		INT02		External interrupt input pin
		AN02		A/D converter analog input pin
		UCK		UART clock I/O pin
12	16	P03	F	General-purpose I/O port
		INT03		External interrupt input pin
		AN03		A/D converter analog input pin
		UO		UART data output pin
		SDA		I^2C data I/O pin
13	17	P04	F	General-purpose I/O port
		INT04		External interrupt input pin
		AN04		A/D converter analog input pin
		UI		UART data input pin
		SCL		I^2C clock I/O pin
14	18	P05	E	General-purpose I/O port
		INT05		External interrupt input pin
		AN05		A/D converter analog input pin
		TO0		Timer output pin
15	19	P06	E	General-purpose I/O port
		INT06		External interrupt input pin
		AN06		A/D converter analog input pin
		TO1		Timer output pin

Table 1.7-1 Pin Description (3 / 4)

Pin no.		Pin name	I/O circuit type[*3]	Function
LQFP32[*1]	SH-DIP32[*2]			
16	20	P07	E	General-purpose I/O port
		INT07		External interrupt input pin
		AN07		A/D converter analog input pin
		EC0		8/16-bit composite timer external clock input pin
17	21	P70	D	General-purpose I/O port
		CMP0_O		Comparator ch. 0 output pin
		OUT0		Output compare ch. 0 output pin
		TRG		16-bit PPG trigger input pin
18	22	P71	I	General-purpose I/O port
		CMP0_P		Comparator ch. 0 positive input pin
		OUT0		Output compare ch. 0 output pin
19	23	P72	I	General-purpose I/O port
		CMP0_N		Comparator ch. 0 negative input pin
		AN09		A/D converter analog input pin
20	24	P73	D	General-purpose I/O port
		CMP1_O		Comparator ch. 1 output pin
		OUT1		Output compare ch. 1 output pin
		PPG		16-bit PPG output pin
21	25	P74	I	General-purpose I/O port
		CMP1_P		Comparator ch. 1 positive input pin
		AN10		A/D converter analog input pin
22	26	P75	I	General-purpose I/O port
		CMP1_N		Comparator ch. 1 negative input pin
		AN10		A/D converter analog input pin
23	27	P76	D	General-purpose I/O port
		CMP2_O		Comparator ch. 2 output pin
		UCK		UART/SIO clock I/O pin
24	28	P63	I	General-purpose I/O port
		CMP2_P		Comparator ch. 2 positive input pin
		AN12		A/D converter analog input pin

Table 1.7-1 Pin Description (4 / 4)

Pin no.		Pin name	I/O circuit type*3	Function
LQFP32*1	SH-DIP32*2			
25	29	P64	I	General-purpose I/O port
		CMP2_N		Comparator ch. 2 negative input pin
		AN13		A/D converter analog input pin
26	30	P65	L	General-purpose I/O port
		CMP3_O		Comparator ch. 3 output pin
		UO		UART/SIO data output pin
		SDA		I2C data I/O pin
27	31	P66	I	General-purpose I/O port
		CMP3_P		Comparator ch. 3 positive input pin
		AN14		A/D converter analog input pin
28	32	P67	I	General-purpose I/O port
		CMP3_N		Comparator ch. 3 negative input pin
		AN15		A/D converter analog input pin
29	1	PF2	A	General-purpose I/O port
		$\overline{\text{RST}}$		Reset pin Dedicated reset pin in MB95F432H/F433H/F434H
30	2	PF0	B	General-purpose I/O port
		X0		Main clock input oscillation pin
31	3	PF1	B	General-purpose I/O port
		X1		Main clock I/O oscillation pin
32	4	V_{SS}	—	Power supply pin (GND)

*1: Package code: FPT-32P-M30
*2: Package code: DIP-32P-M06
*3: 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.

■ I/O Circuit Types

Table 1.8-1 I/O Circuit Types (1 / 4)

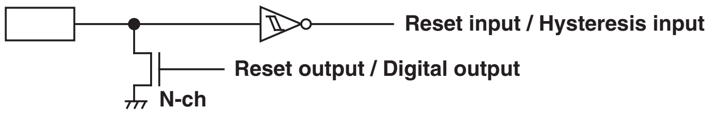
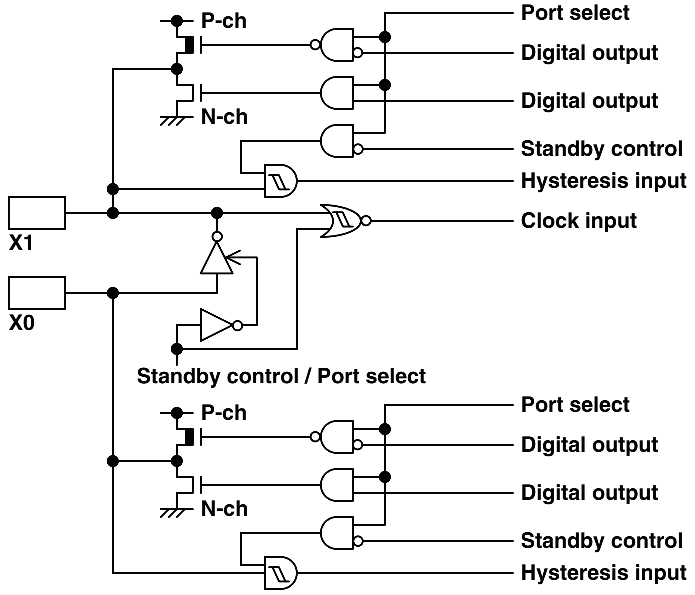
Type	Circuit	Remarks
A	 Reset input / Hysteresis input Reset output / Digital output N-ch	• N-ch open drain output • Hysteresis input • Reset output
B	 P-ch N-ch Port select Digital output Digital output Standby control Hysteresis input Clock input X1 X0 Standby control / Port select P-ch N-ch Port select Digital output Digital output Standby control Hysteresis input	• Oscillation circuit • High-speed side Feedback resistance: approx. 1 MΩ • CMOS output • Hysteresis input

Table 1.8-1 I/O Circuit Types (2 / 4)

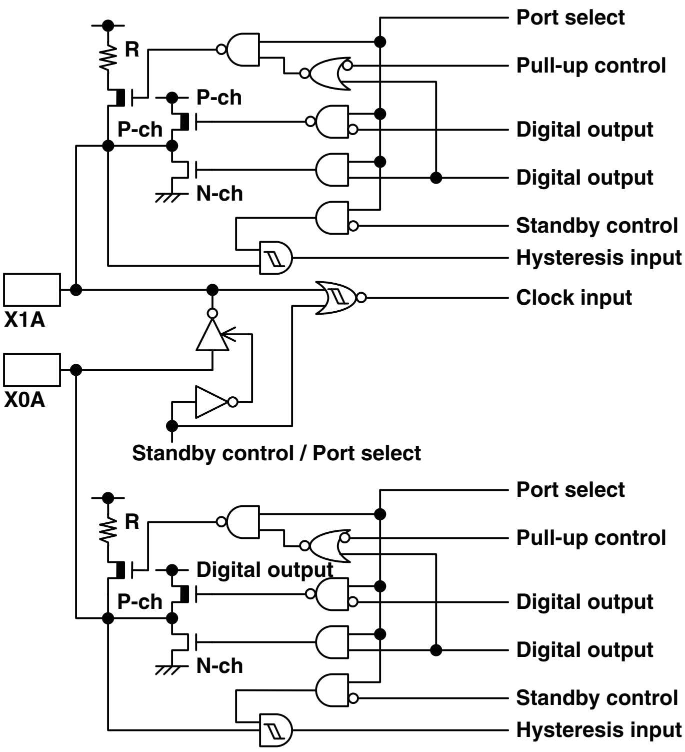
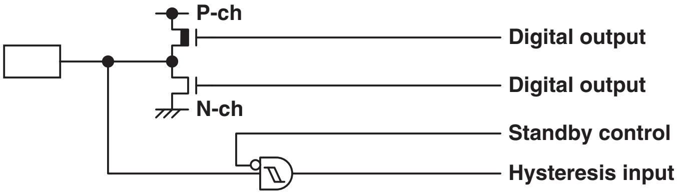
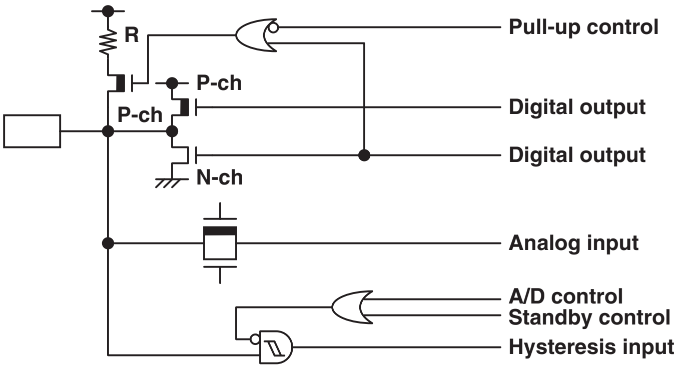
Type	Circuit	Remarks
C	 Port select Pull-up control R P-ch P-ch Digital output Digital output N-ch Standby control Hysteresis input Clock input X1A X0A Standby control / Port select Port select Pull-up control R Digital output P-ch Digital output Digital output N-ch Standby control Hysteresis input	• Oscillation circuit • Low-speed side Feedback resistance: approx. 10 MΩ • CMOS output • Hysteresis input • Pull-up control available
D	 P-ch Digital output Digital output N-ch Standby control Hysteresis input	• CMOS output • Hysteresis input
E	 Pull-up control R P-ch P-ch Digital output Digital output N-ch Analog input A/D control Standby control Hysteresis input	• CMOS output • Hysteresis input • Pull-up control available • Analog input

Table 1.8-1 I/O Circuit Types (3 / 4)

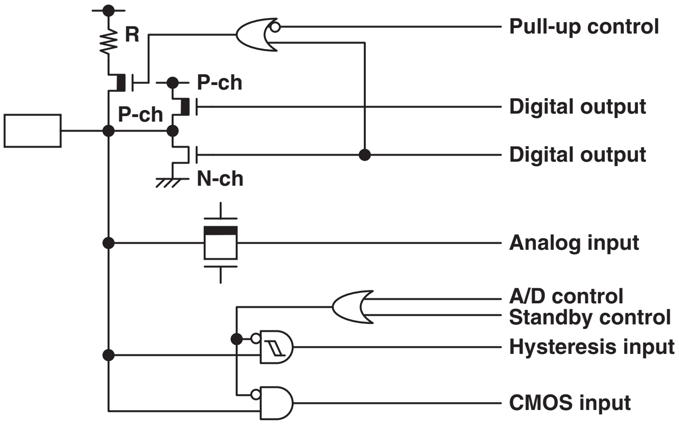
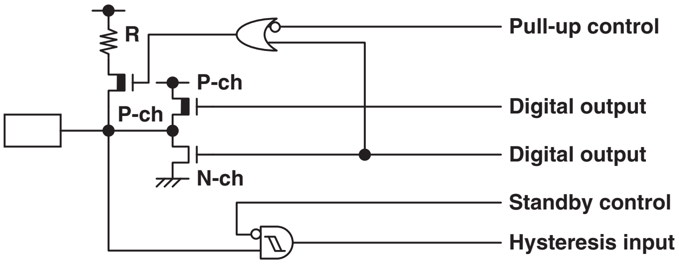
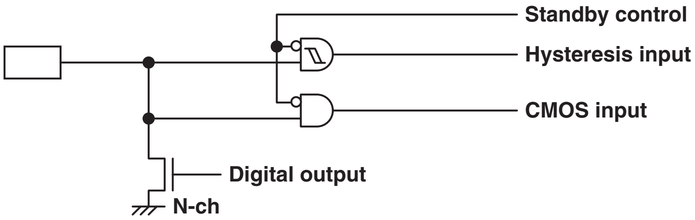
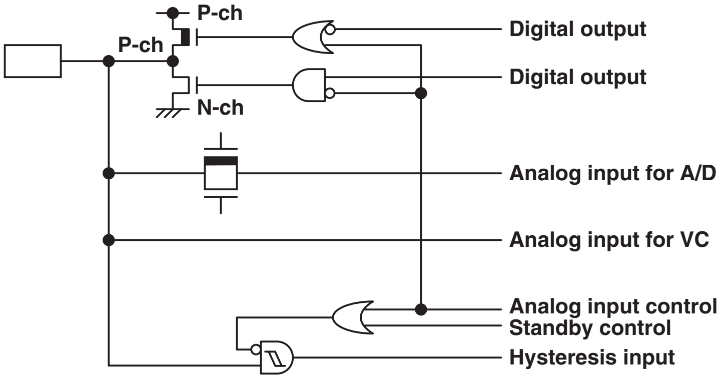
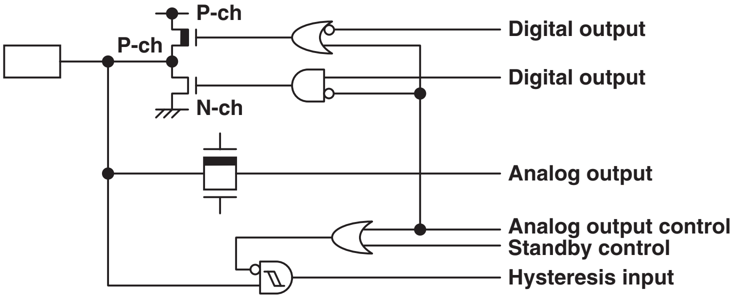
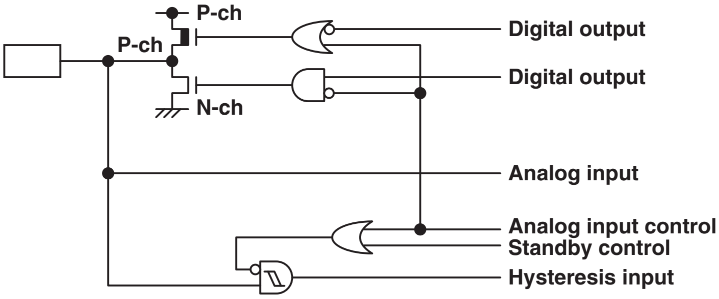
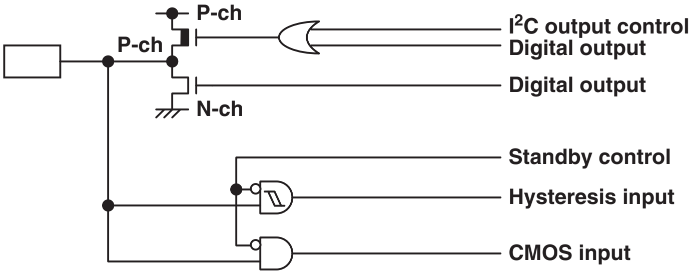
Type	Circuit	Remarks
F	 R Pull-up control P-ch P-ch Digital output Digital output N-ch Analog input A/D control Standby control Hysteresis input CMOS input	• CMOS output • Hysteresis input • CMOS input • Pull-up control available • Analog input • N-ch open drain output (as I^2C output)
G	 R Pull-up control P-ch P-ch Digital output Digital output N-ch Standby control Hysteresis input	• CMOS output • Hysteresis input • Pull-up control available
H	 Standby control Hysteresis input CMOS input Digital output N-ch	• N-ch open drain output • Hysteresis input • CMOS input
I	 P-ch P-ch Digital output Digital output N-ch Analog input for A/D Analog input for VC Analog input control Standby control Hysteresis input	• CMOS output • Hysteresis input
J	 P-ch P-ch Digital output Digital output N-ch Analog output Analog output control Standby control Hysteresis input	• CMOS output • Hysteresis input

Table 1.8-1 I/O Circuit Types (4 / 4)

Type	Circuit	Remarks
K	 P-ch P-ch N-ch Digital output Digital output Analog input Analog input control Standby control Hysteresis input	• CMOS output • Hysteresis input
L	 P-ch P-ch N-ch I²C output control Digital output Digital output Standby control Hysteresis input CMOS input	• CMOS output • Hysteresis input • CMOS input • N-ch open drain output (as I^2C output)

CHAPTER 2

NOTES ON DEVICE HANDLING

This chapter provides notes on using the MB95430H Series.

2.1 Notes on Device Handling

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

■ DEVICE HANDLING

- 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 MB95430H 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.

- Note 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.

- $\overline{\text{RST}}$ pin

  Connect the $\overline{\text{RST}}$ pin directly to an external pull-up resistor.

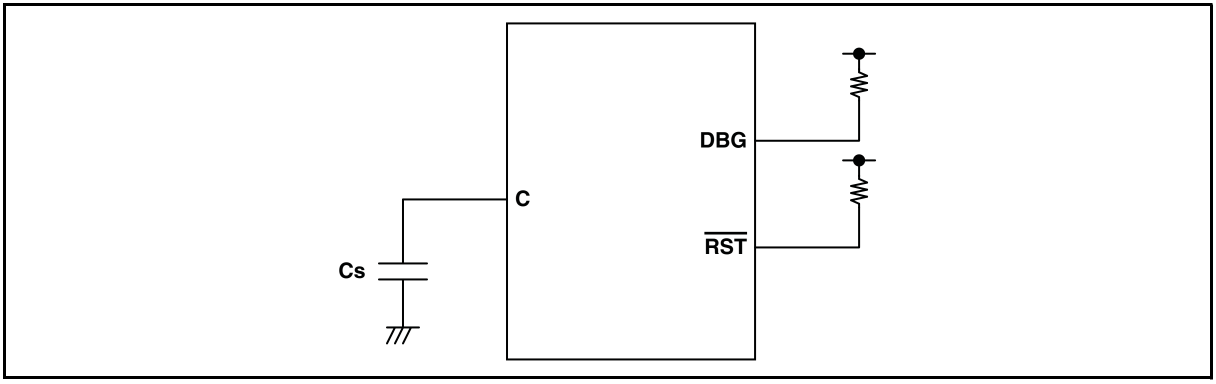
  To prevent the device from unintentionally entering the reset mode due to noise, minimize the distance between the $\overline{\text{RST}}$ pin and the V_{CC} or V_{SS} pin when designing the layout of the printed circuit board.

  The $\overline{\text{RST}}$/PF2 pin functions as the reset input/output pin after power-on. In addition, the reset output can be enabled by the RSTOE bit in the SYSC1 register, and the reset input function or the general purpose I/O function can be selected by the RSTEN bit in the SYSC1 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/$\overline{\text{RST}}$/C Pin Connection



- Note on serial communication

  In serial communication, reception of wrong data may occur due to noise or other causes. Therefore, design a printed circuit board to prevent noise from occurring. Taking account of the reception of wrong data, take measures such as adding a checksum to the end of data in order to detect errors. If an error is detected, retransmit the data.

CHAPTER 3
MEMORY SPACE

This chapter describes the memory space.

3.1 Memory Space

The memory space of the MB95430H Series is 64 Kbyte 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

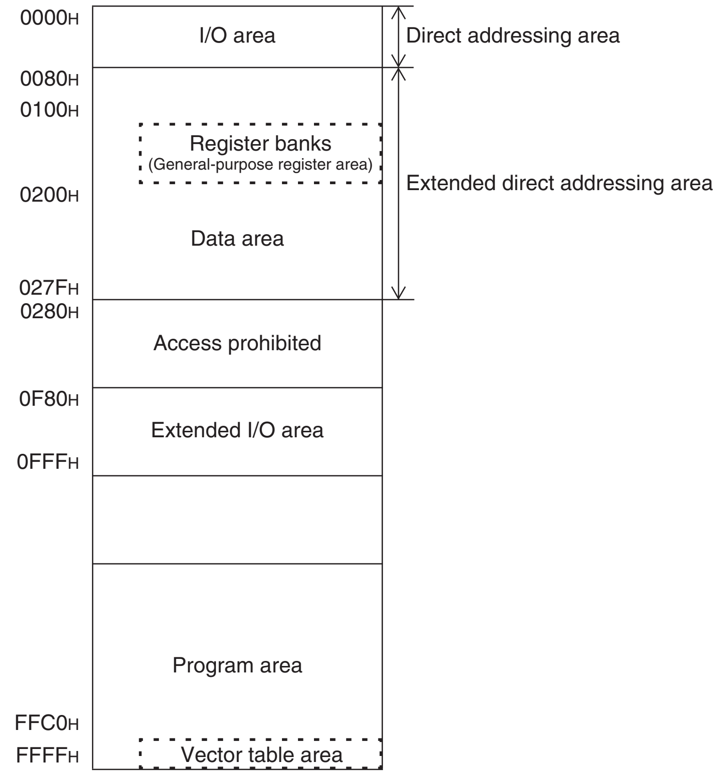
- Static RAM is incorporated in the data area as the internal data area.
- The internal RAM size varies according to product.
- The RAM area from 0090_H to $00FF_H$ can be accessed at high-speed by using direct addressing instructions.
- The area from 0100_H to $027F_H$ is an extended direct addressing area. It can be accessed at high-speed by direct addressing instructions with a direct bank pointer set. (MB95F434H/F434K)
- 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. (MB95F432H/F432K/F433H/F434K)
- The area from 0100_H to $01FF_H$ can be used as a general-purpose register area. (MB95F434H/F434K)
- The area from 0100_H to $017F_H$ can be used as a general-purpose register area. (MB95F432H/F432K/F433H/F433K)

● 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 MB95F434H/F434K) (Addresses: 0100$_H$ to 017F$_H$ in MB95F432H/F432K/F433H/F433K)

- 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 "5.1.1 Register Bank Pointer (RP)" and "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 30 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 "E.2 Special Instruction" in APPENDIX.

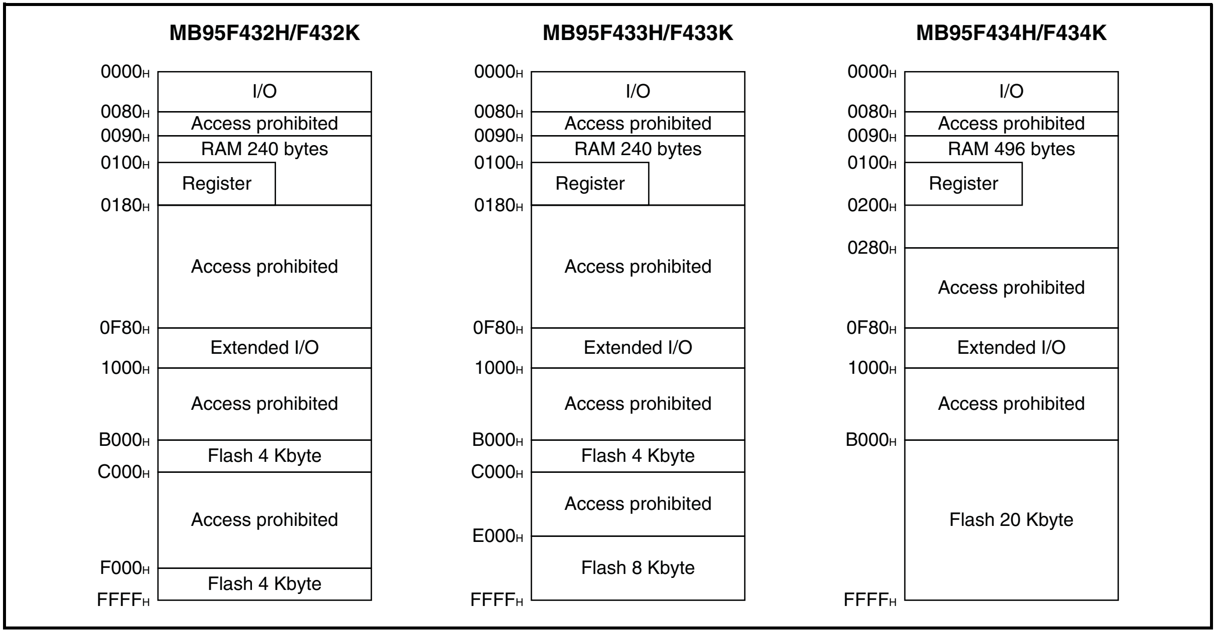
MB95430H Series

3.2 Memory Map

This section shows a memory map of the MB95430H Series.

■ Memory Map

Figure 3.2-1 Memory Map



Parameter / Part number	Flash memory	RAM
MB95F432H/F432K	8 Kbyte	240 bytes
MB95F433H/F433K	12 Kbyte	240 bytes
MB95F434H/F434K	20 Kbyte	496 bytes

CHAPTER 4
MEMORY ACCESS MODE

This chapter describes the memory access mode.

4.1 Memory Access Mode

The MB95430H Series support only one memory access mode: single-chip mode.

■ Single-chip Mode

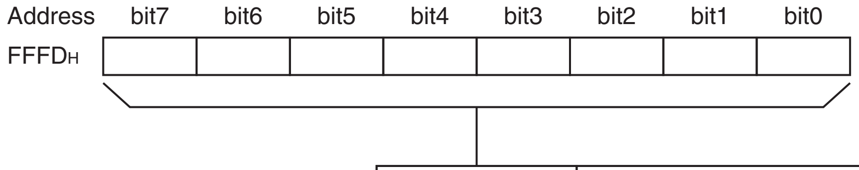
In single-chip mode, only the internal RAM and ROM are used, and no external bus access is executed.

● 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
00H	Selects single-chip mode.
Other than 00H	Reserved. Do not set mode data to any value other than 00H.

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

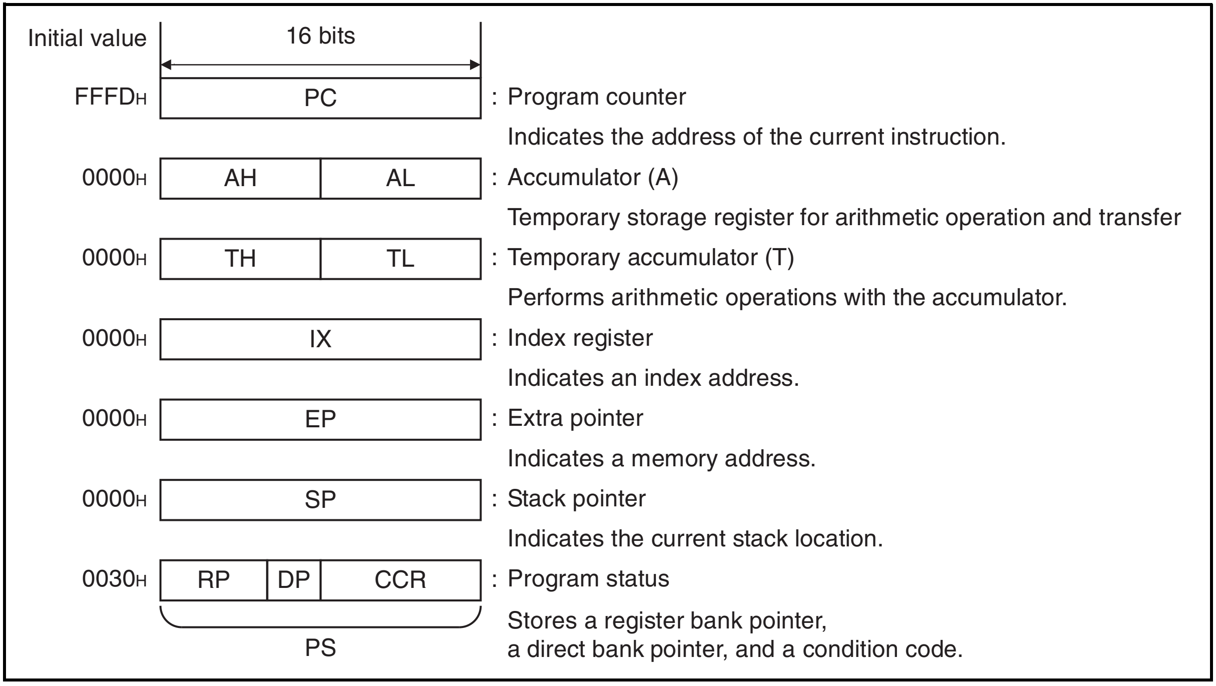
The CPU has dedicated registers: a program counter (PC), two registers for arithmetic operations (A and T), three address pointers (IX, EP, and SP), and the program status (PS) register. Each of the registers is 16 bits long. The PS register consists of the register bank pointer (RP), direct pointer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


Initial value
16 bits
FFFDH
PC
: Program counter
Indicates the address of the current instruction.
0000H
AH
AL
: Accumulator (A)
Temporary storage register for arithmetic operation and transfer
0000H
TH
TL
: Temporary accumulator (T)
Performs arithmetic operations with the accumulator.
0000H
IX
: Index register
Indicates an index address.
0000H
EP
: Extra pointer
Indicates a memory address.
0000H
SP
: Stack pointer
Indicates the current stack location.
0030H
RP
DP
CCR
: Program status
Stores a register bank pointer,
a direct bank pointer, and a condition code.
PS

■ 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.

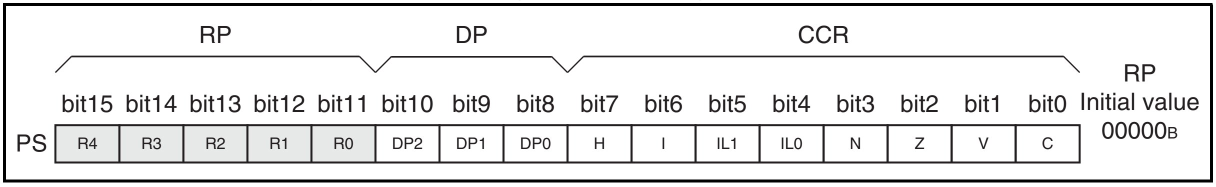
5.1.1 Register Bank Pointer (RP)

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.

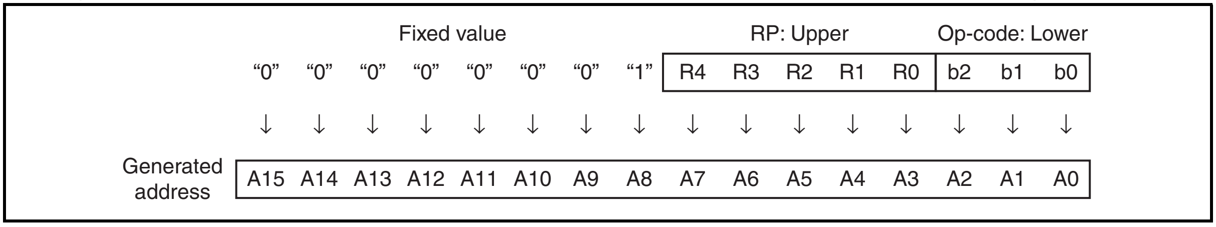
Figure 5.1-2 Configuration of Register Bank Pointer



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

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



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

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

The register bank pointer allows the space from "0100$_H$" to "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.

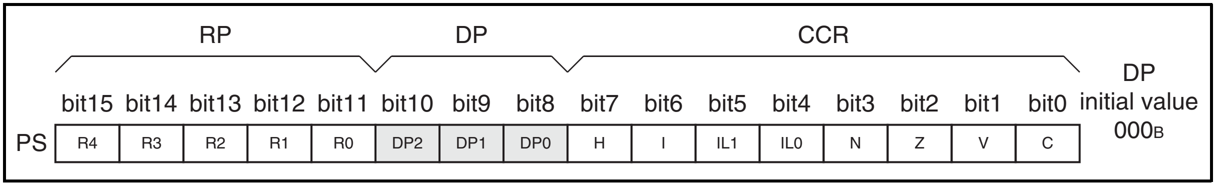
5.1.2 Direct Bank Pointer (DP)

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



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

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, the access area is "0090_H to $00FF_H$" in the MB95430H Series.

*2: The available access area is up to "0180_H" in MB95F432H/F432K/F433H/F433K.

*3: The available access area is up to "0280_H" in MB95F434H/F434K.

Table 5.1-2 Direct Address Instruction List

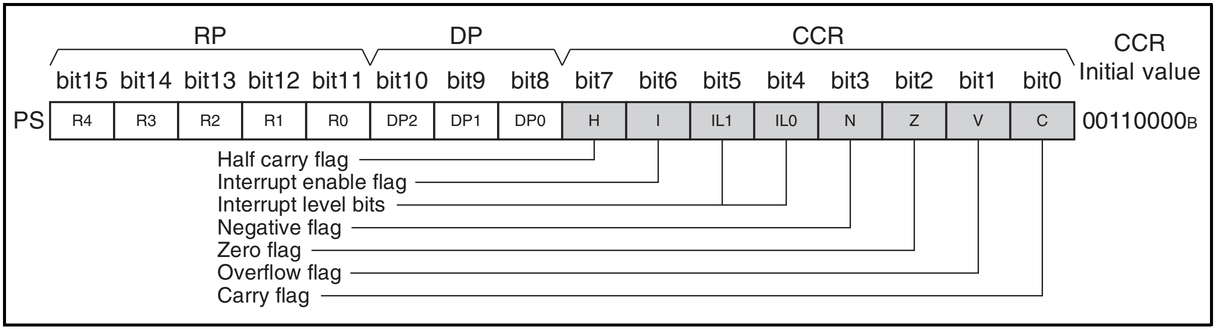
Applicable instructions
CLRB dir:bit
SETB dir:bit
BBC dir:bit,rel
BBS dir:bit,rel
MOV A,dir
CMP A,dir
ADDC A,dir
SUBC A,dir
MOV dir,A
XOR A,dir
AND A,dir
OR A,dir
MOV dir,#imm
CMP dir,#imm
MOVW A,dir
MOVW dir,A

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 (CCR)



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)

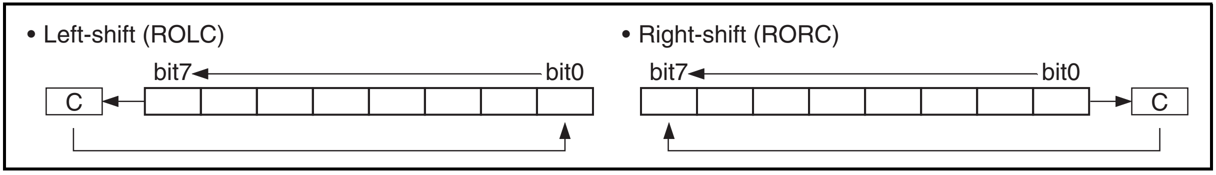
This flag indicates whether the result of an operation has caused an overflow, with the operand used in the operation being regarded as an integer expressed as a complement of two. If an overflow occurs, the overflow flag is set to "1"; otherwise, it is set to "0".

● 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

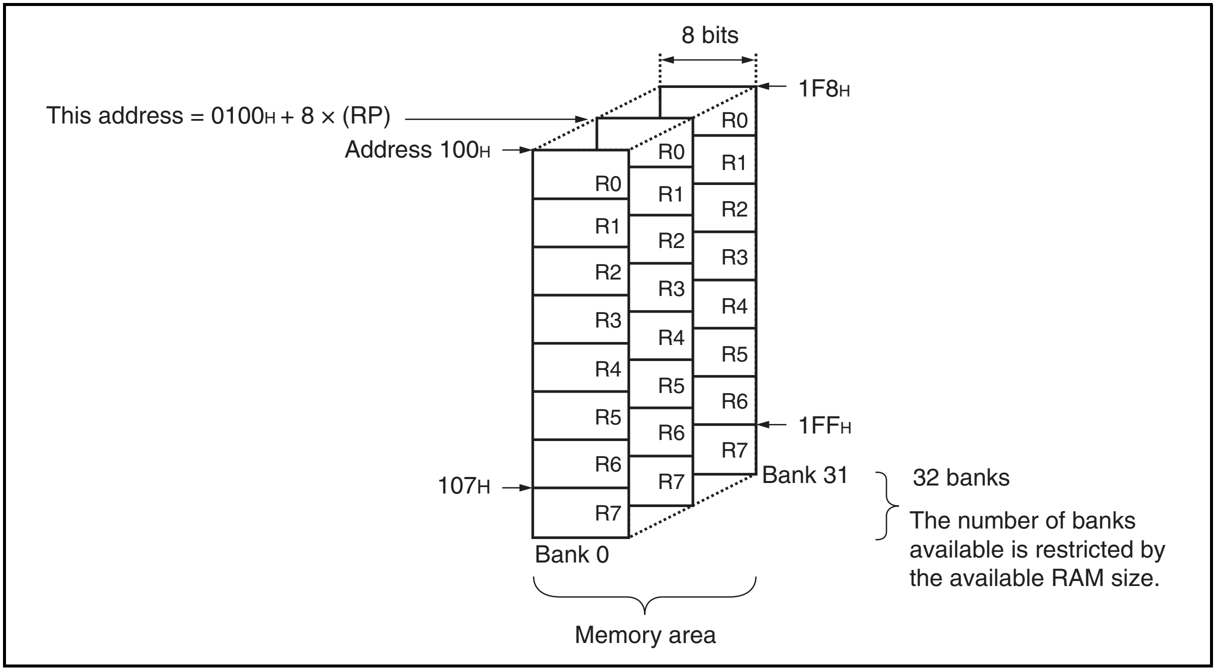
The general-purpose registers are a memory block in which each bank consists of eight 8-bit registers. Up to 32 register banks can be used in total. The register bank pointer (RP) is used to specify a register bank.
Register banks are useful for interrupt handling, vector call processing, and sub-routine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


8 bits
1F8H
This address = 0100H + 8 × (RP)
Address 100H
R0
R1
R2
R3
R4
R5
R6
R7
1FFH
107H
Bank 31
Bank 0
32 banks
The number of banks available is restricted by the available RAM size.
Memory area

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.

Note:

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 bytes, 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 bytes. 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.

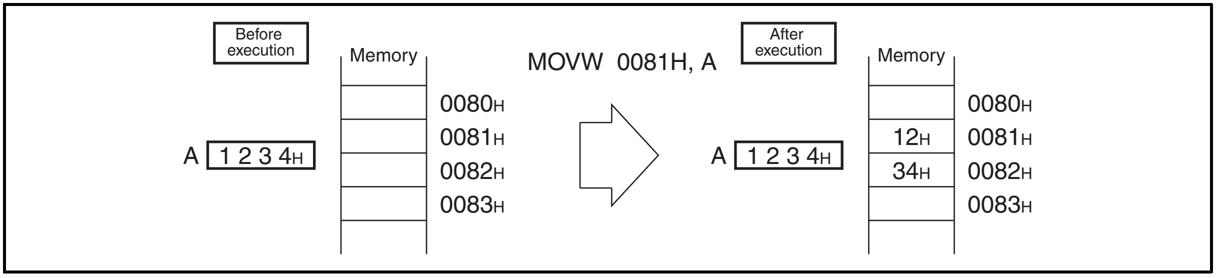
■ Placement of 16-bit Data in Memory

● 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

```
[Example] MOV  A,  5678H    ; Extended address
          MOVW A,  #1234H   ; 16-bit immediate data

                    Assemble
              :
          XXX0H XX XX
          XXX2H 60 56 78    ; Extended address
          XXX5H E4 12 34    ; 16-bit immediate data
          XXX8H XX
              :
```

● 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^2MC-8FX family has a built-in clock controller that optimizes its power consumption. It supports both the external main clock and the external subclock.
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 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 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.

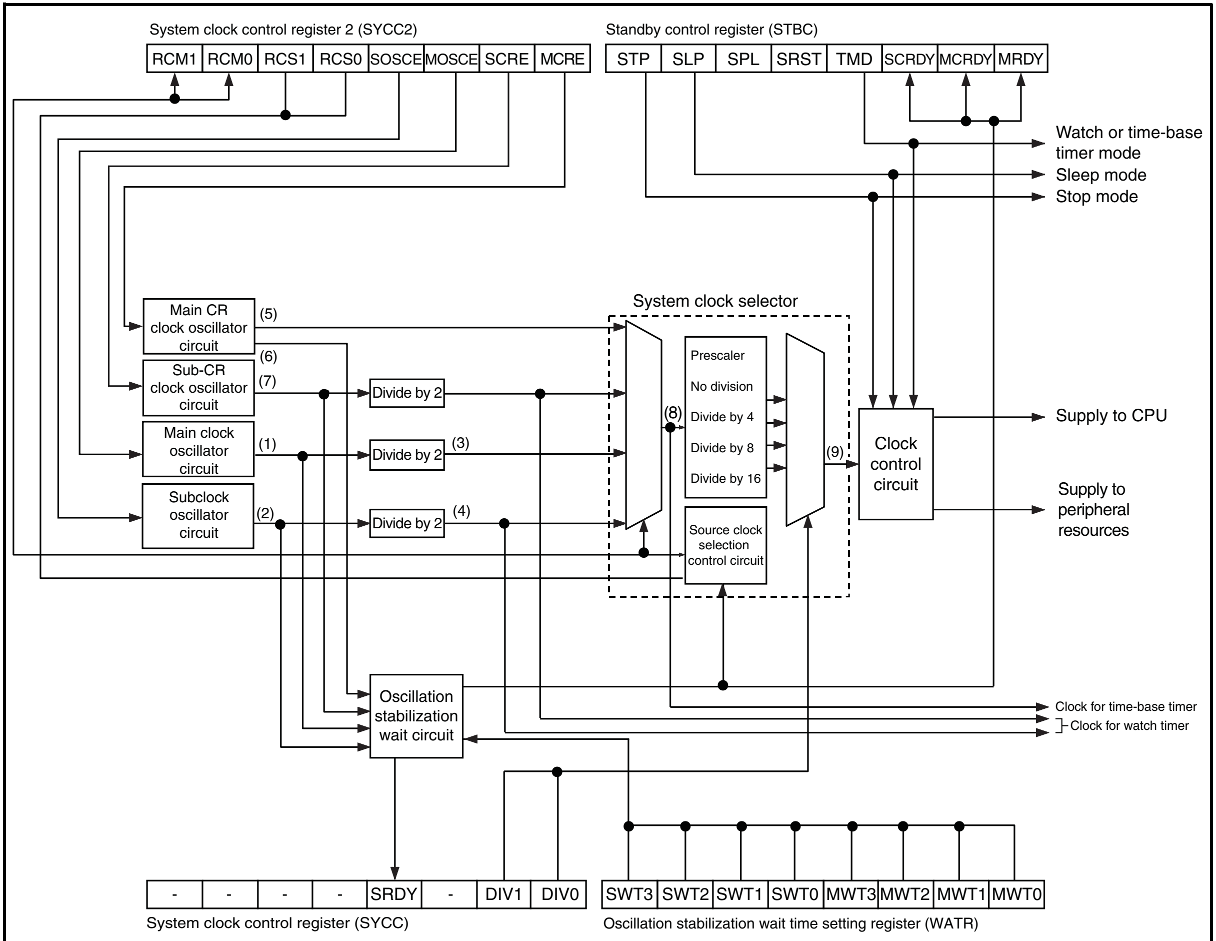
The clock controller selects the optimum power consumption and functions depending on the combination of clock mode and standby mode.

This device has four source clocks: a main clock formed by dividing the main oscillation clock by 2, a subclock formed by dividing the sub-oscillation clock by 2, a main CR clock, and a sub-CR clock formed by dividing the sub-CR oscillation clock by 2.

■ 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 Controller


System clock control register 2 (SYCC2)
RCM1
RCM0
RCS1
RCS0
SOSCE
MOSCE
SCRE
MCRE
Standby control register (STBC)
STP
SLP
SPL
SRST
TMD
SCRDY
MCRDY
MRDY
Watch or time-base timer mode
Sleep mode
Stop mode
Main CR clock oscillator circuit
Sub-CR clock oscillator circuit
Main clock oscillator circuit
Subclock oscillator circuit
(5)
(6)
(7)
(1)
(2)
(3)
(4)
Divide by 2
Divide by 2
Divide by 2
System clock selector
Prescaler
No division
Divide by 4
Divide by 8
Divide by 16
(8)
(9)
Source clock selection control circuit
Clock control circuit
Supply to CPU
Supply to peripheral resources
Oscillation stabilization wait circuit
Clock for time-base timer
Clock for watch timer
-
-
-
-
SRDY
-
DIV1
DIV0
System clock control register (SYCC)
SWT3
SWT2
SWT1
SWT0
MWT3
MWT2
MWT1
MWT0
Oscillation stabilization wait time setting register (WATR)

(1): Main clock (F_{CH})
(2): Subclock (F_{CL})
(3): Main clock
(4): Subclock
(5): Main CR clock (F_{CRH})
(6): Main CR reference clock (F_{CRHS})
(7): Sub-CR clock (F_{CRL})
(8): Source clock
(9): Machine clock (MCLK)

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

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 a dedicated timer in the oscillation stabilization wait circuit as the oscillation stabilization signal for the main clock, or one of the 15 types of oscillation stabilization signals created by the same dedicated timer 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, time-base 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 by dividing the main oscillation clock by 2.
Main CR clock mode	The machine clock is generated from the main CR clock.
Subclock mode	The machine clock is generated by dividing the sub-oscillation clock by 2.
Sub-CR clock mode	The machine clock is generated by dividing the sub-CR oscillation clock by 2.

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 time-base timer output selected) Subclock (with watch prescaler output selected)

For some peripheral functions other than the one listed above, the time-base 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 time-base timer mode and watch mode, the standby mode can be set independently of the clock mode.

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

Table 6.1-3 Standby 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.
Time-base timer mode	Clock signals are only supplied to the time-base timer and the watch prescaler, while the clock supply to other circuits is stopped. As a result, all the functions other than the time-base timer, watch prescaler, external interrupt, and low-voltage detection reset (option) are stopped. The time-base timer mode can be used in main clock mode and main CR clock mode.
Watch mode	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 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.

Note:

Clocks that are not mentioned in Table 6.1-3 are supplied under particular settings.

For example, with main clock mode being used in stop mode, when SYCC2:SOSCE and SYCC2:SCRE have been set to "1", the watch prescaler operates.

In addition, with the hardware watchdog timer already started, the watchdog timer operates also in standby mode.

■ Combinations of Clock Mode and Standby Mode

Table 6.1-4 and Table 6.1-5 list 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 (1)

Function	RUN				Sleep			
	Main clock mode	Main CR clock mode	Subclock mode	Sub-CR clock mode	Main clock mode	Main CR clock mode	Subclock mode	Sub-CR clock mode
Main clock	Operating	Stopped*1	Stopped		Operating	Stopped*1	Stopped	
Main CR clock	Stopped*2	Operating	Stopped		Stopped*2	Operating	Stopped	
Subclock	Operating*3		Operating	Operating*3	Operating*3		Operating	Operating*3
Sub-CR clock	Operating*4		Operating*4	Operating	Operating*4		Operating*4	Operating
CPU	Operating		Operating		Stopped		Stopped	
ROM	Operating		Operating		Value held		Value held	
RAM								
I/O ports	Operating		Operating		Output held		Output held	
Time-base timer	Operating		Stopped		Operating		Stopped	
Watch prescaler	Operating*3, *4		Operating		Operating*3, *4		Operating	
External interrupt	Operating		Operating		Operating		Operating	
Hardware watchdog timer	Operating		Operating		Operating*5		Operating*5	
Software watchdog timer	Operating		Operating		Stopped		Stopped	
Low-voltage detection reset	Operating		Operating		Operating		Operating	
Other peripheral functions	Operating		Operating		Operating		Operating	

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

*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.

Table 6.1-5 Combinations of Standby Mode and Clock Mode and Internal Operating States (2)

Function	Time-base timer		Watch prescaler		Stop			
	Main clock mode	Main CR clock mode	Subclock mode	Sub-CR clock mode	Main clock mode	Main CR clock mode	Subclock mode	Sub-CR clock mode
Main clock	Operating	Stopped*1	Stopped		Stopped			
Main CR clock	Stopped*2	Operating	Stopped		Stopped			
Subclock	Operating*3		Operating	Operating*3	Operating*3		Stopped	
Sub-CR clock	Operating*4		Operating*4	Operating	Operating*4		Stopped	
CPU	Stopped		Stopped		Stopped			
ROM	Value held		Value held		Value held			
RAM								
I/O ports	Output held / Hi-Z		Output held		Output held/Hi-Z			
Time-base timer	Operating		Stopped		Stopped			
Watch prescaler	Operating*3, *4		Operating		Operating*3, 4		Stopped	
External interrupt	Operating		Operating		Operating			
Hardware watchdog timer	Operating*5		Operating*5		Operating*5			
Software watchdog timer	Stopped		Stopped		Stopped			
Low-voltage detection reset	Operating		Operating		Operating			
Other peripheral functions	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 operates when the subclock oscillation enable bit in the system clock control register 2 (SYCC2:SOSCE) is set to "1".

*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.

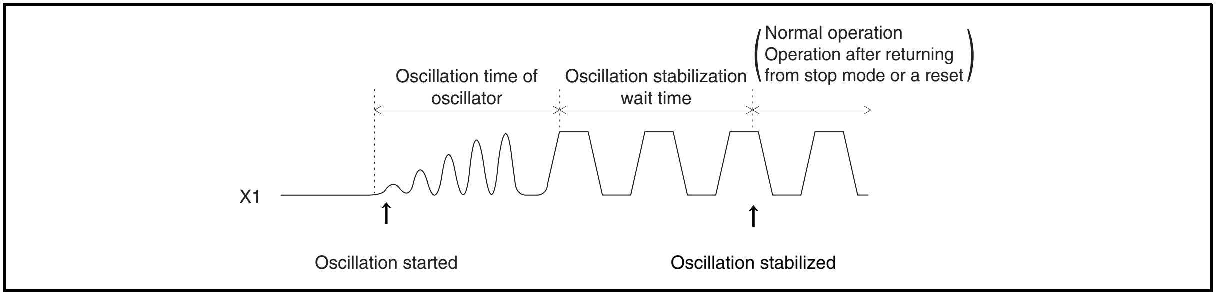
■ Oscillation Stabilization Wait Time

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


Normal operation
Operation after returning from stop mode or a reset
Oscillation time of oscillator
Oscillation stabilization wait time
X1
Oscillation started
Oscillation stabilized

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.
	Other than power-on reset	Register settings (WATR:MWT3, MWT2, MWT1, MWT0)
Subclock	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 CR oscillation stabilization wait time to elapse.

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	$2^5/F_{CRL}$

*: F_{CRHS}: 1 MHz

■ 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.

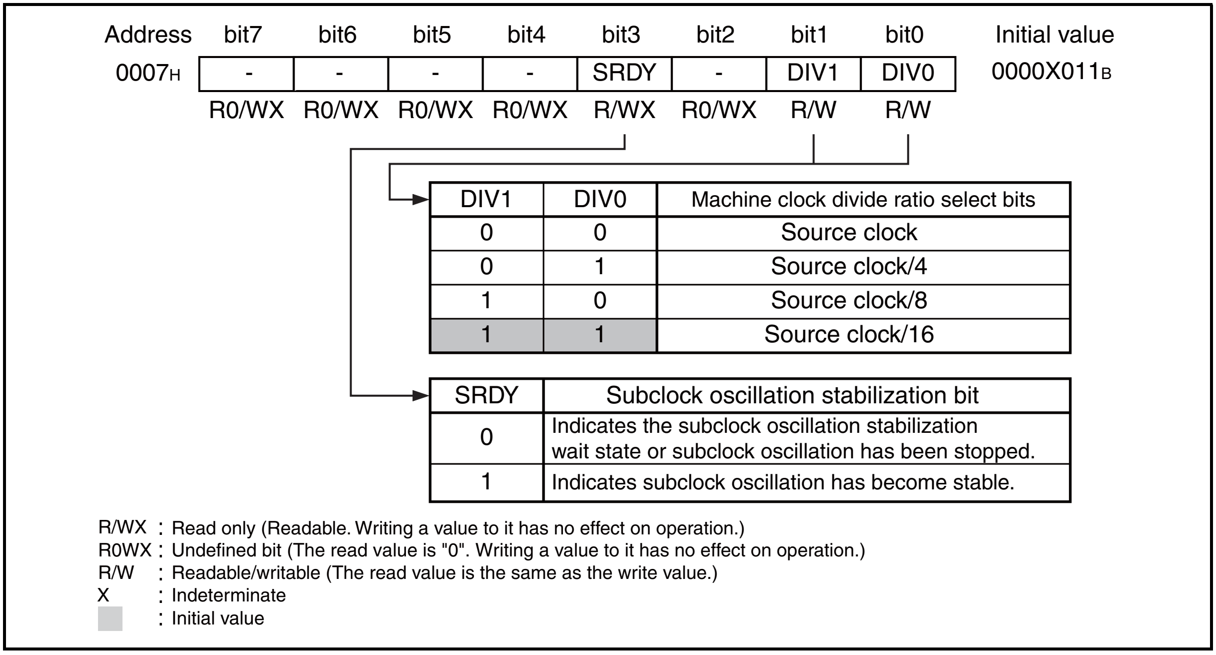
For details on state transition, see "6.7 Clock Modes" and "6.8 Operations in Low-power Consumption Mode (Standby Mode)".

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
0007$_H$	-	-	-	-	SRDY	-	DIV1	DIV0	0000X011$_B$
	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R0/WX	R/W	R/W	

DIV1	DIV0	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>The read value is always "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit3</td><td>SRDY:
Subclock oscillation stabilization bit</td><td>This bit indicates whether subclock oscillation has become stable.
• When the SRDY 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.</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</th></tr>
<tr><td>0</td><td>0</td><td>Source clock (No division)</td></tr>
<tr><td>0</td><td>1</td><td>Source clock/4</td></tr>
<tr><td>1</td><td>0</td><td>Source clock/8</td></tr>
<tr><td>1</td><td>1</td><td>Source clock/16</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	SWT2	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

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>
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>$(2^{14}\text{-}2)/F_{CH}$</td><td>About 4.10 ms</td></tr>
<tr><td>1110_B</td><td>2^{13}-2</td><td>$(2^{13}\text{-}2)/F_{CH}$</td><td>About 2.05 ms</td></tr>
<tr><td>1101_B</td><td>2^{12}-2</td><td>$(2^{12}\text{-}2)/F_{CH}$</td><td>About 1.02 ms</td></tr>
<tr><td>1100_B</td><td>2^{11}-2</td><td>$(2^{11}\text{-}2)/F_{CH}$</td><td>511.5 µs</td></tr>
<tr><td>1011_B</td><td>2^{10}-2</td><td>$(2^{10}\text{-}2)/F_{CH}$</td><td>255.5 µs</td></tr>
<tr><td>1010_B</td><td>2^{9}-2</td><td>$(2^{9}\text{-}2)/F_{CH}$</td><td>127.5 µs</td></tr>
<tr><td>1001_B</td><td>2^{8}-2</td><td>$(2^{8}\text{-}2)/F_{CH}$</td><td>63.5 µs</td></tr>
<tr><td>1000_B</td><td>2^{7}-2</td><td>$(2^{7}\text{-}2)/F_{CH}$</td><td>31.5 µs</td></tr>
<tr><td>0111_B</td><td>2^{6}-2</td><td>$(2^{6}\text{-}2)/F_{CH}$</td><td>15.5 µs</td></tr>
<tr><td>0110_B</td><td>2^{5}-2</td><td>$(2^{5}\text{-}2)/F_{CH}$</td><td>7.5 µs</td></tr>
<tr><td>0101_B</td><td>2^{4}-2</td><td>$(2^{4}\text{-}2)/F_{CH}$</td><td>3.5 µs</td></tr>
<tr><td>0100_B</td><td>2^{3}-2</td><td>$(2^{3}\text{-}2)/F_{CH}$</td><td>1.5 µs</td></tr>
<tr><td>0011_B</td><td>2^{2}-2</td><td>$(2^{2}\text{-}2)/F_{CH}$</td><td>0.5 µs</td></tr>
<tr><td>0010_B</td><td>2^{1}-2</td><td>$(2^{1}\text{-}2)/F_{CH}$</td><td>0.0 µs</td></tr>
<tr><td>0001_B</td><td>2^{1}-2</td><td>$(2^{1}\text{-}2)/F_{CH}$</td><td>0.0 µs</td></tr>
<tr><td>0000_B</td><td>2^{1}-2</td><td>$(2^{1}\text{-}2)/F_{CH}$</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 standby control register (STBC: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>

■ Note on Setting WATR Register

When using the dual operation Flash function of a device not equipped with the low-voltage detection reset, always set the main clock oscillation stabilization wait time to 90 µs or above (set WATR:MWT[3:0] to "1010_B" or above with the main clock frequency F_{CH} being 4 MHz).

The above setting requirement applies to the following products:

MB95F432H/F433H/F434H

When a flash write/erase operation occurs with the main clock oscillation stabilization wait time having ended within 90 µs, the operation may fail.

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, time-base timer mode, or watch mode, to set the pin state in stop mode, time-base timer mode, and watch mode, and to control the generation of software resets.

■ Standby Control Register (STBC)

Figure 6.5-1 Standby Control Register (STBC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0008_H	STP	SLP	SPL	SRST	TMD	SCRDY	MCRDY	MRDY	$00000XXX_B$
	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 time-base 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, time-base timer mode, or watch mode.
1	Places external pins in a high impedance state in stop mode, time-base timer mode, or watch mode.

SLP	Sleep bit	
	Read	Write
0	"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

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. The read value is “0”.)
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":** Has no effect on operation. **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":** Has no effect on operation. **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, time-base timer mode, and watch mode. **Writing "0":** The state (level) of an external pin is kept in stop mode, time-base timer mode and watch mode. **Writing "1":** An external pin becomes high impedance in stop mode, time-base 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	This bit sets transition to time-base timer mode or watch mode. • Writing "1" to this bit in main clock mode or main CR clock mode causes the device to transit to time-base 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 SCRDY 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.
bit1	MCRDY: Main CR clock oscillation stabilization bit	This bit indicates whether main CR clock oscillation has become stable. • When the MCRDY bit is set to "1", that indicates the oscillation stabilization wait time for the main CR clock has elapsed. • When the MCRDY 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.

Note:

- 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 select 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)

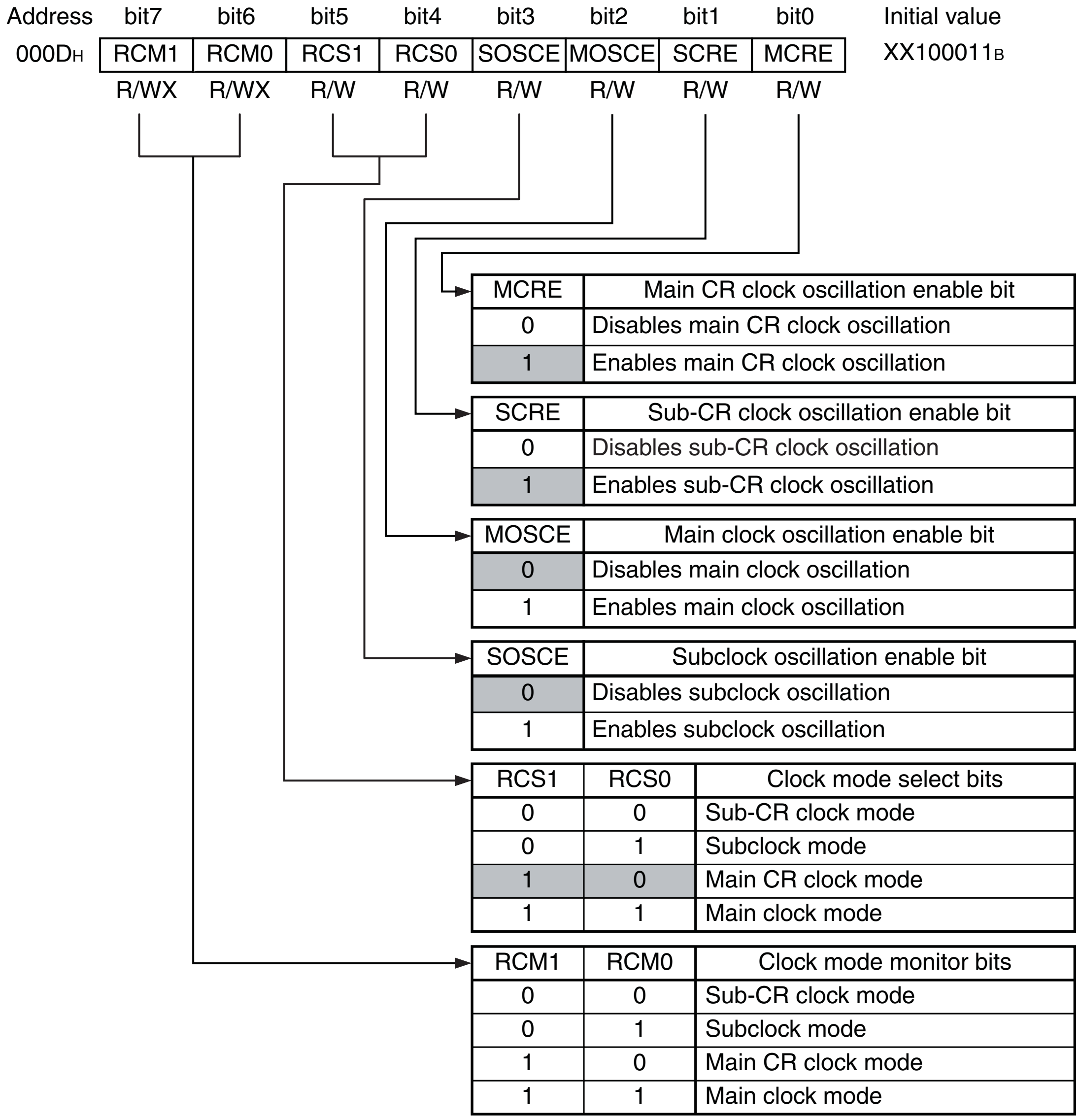
  When released from standby mode, the device returns to the normal operating state.

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	RCM1	RCM0	RCS1	RCS0	SOSCE	MOSCE	SCRE	MCRE	XX100011B
	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/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.)
X	:	Indeterminate
▨	:	Initial value

Note: Read also CHAPTER 31 “SYSTEM CONFIGURATION CONTROLLER” before enabling main clock oscillation or subclock oscillation.

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"**: Indicates sub-CR clock mode. **"01_B"**: Indicates subclock mode. **"10_B"**: Indicates main CR clock mode. **"11_B"**: Indicates 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 configuration register, writing "11_B" to these bits is ignored, and their values remain unchanged. • If subclock oscillation has been disabled by the system configuration register, 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 configuration register, 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 configuration register, writing "1" to this bit is ignored, and its value remains 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 "00_B", this bit is set to "1". • If the RCS or RCM bits are "00_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 time-base timer operates using the main clock.

The watch prescaler can operate using the subclock or the sub-CR clock.

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 time-base 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

In subclock mode, main clock oscillation is stopped* and the subclock is used as the machine clock for the CPU and peripheral functions. In this mode, the time-base 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 time-base timer and the watchdog timer operate using the main clock.

The watch prescaler can operate using the subclock or the sub-CR clock.

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 time-base timer mode.

■ Operations in Sub-CR Clock Mode

In sub-CR clock mode, main clock oscillation is stopped* and the sub-CR clock is used as the machine clock for the CPU and peripheral functions. In this mode, the time-base timer stops as it requires the main clock for operation. The watch prescaler operates using the sub-CR clock.

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.

* 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 clock and the main CR clock cannot be enabled by writing "1" to SYCC2:MOSCE and "1" to SYCC2:MCRE respectively.

■ 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

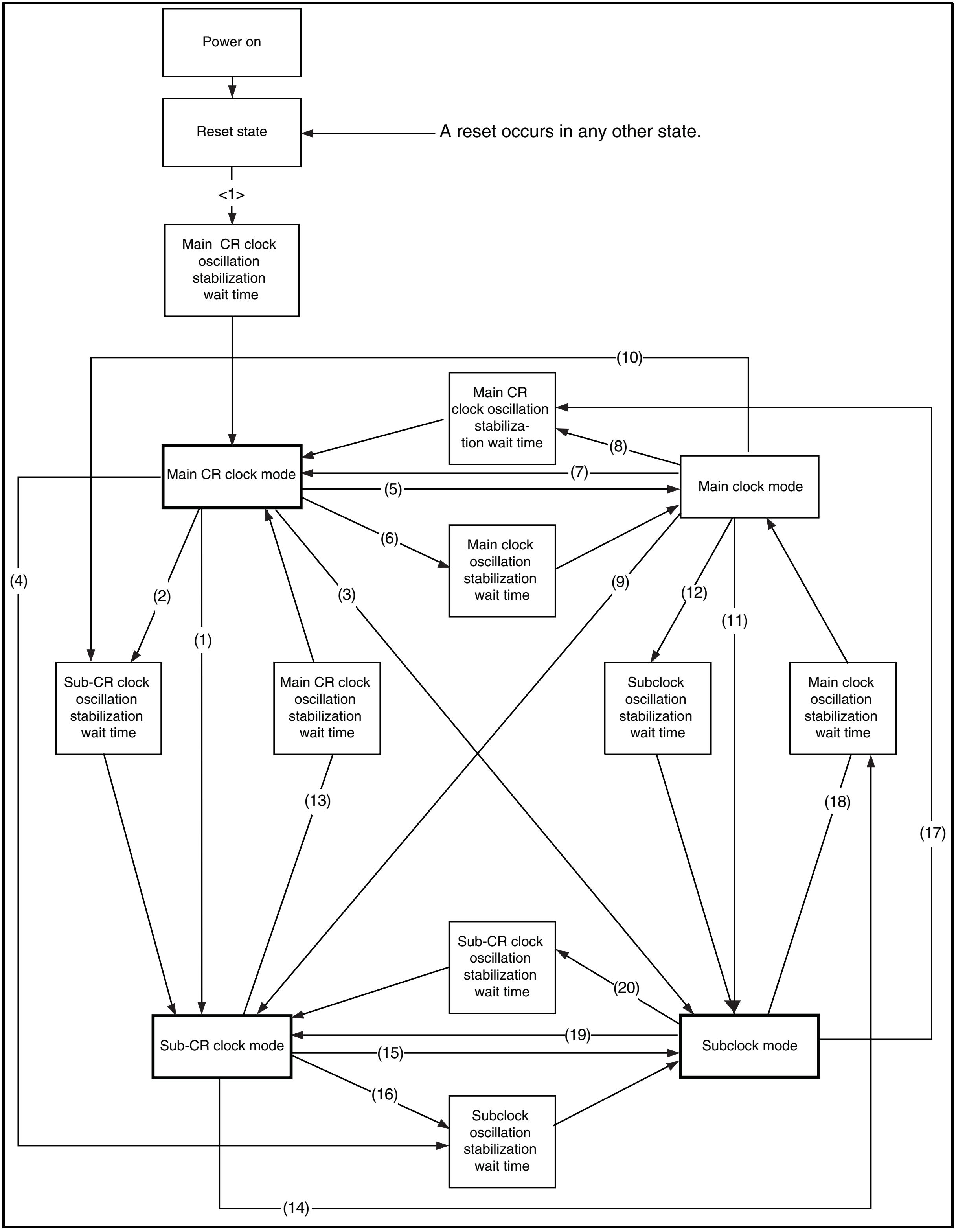

Power on
Reset state
A reset occurs in any other state.
<1>
Main CR clock oscillation stabilization wait time
(10)
Main CR clock oscillation stabiliza-tion wait time
(8)
Main CR clock mode
(7)
(5)
Main clock mode
(6)
Main clock oscillation stabilization wait time
(4)
(2)
(3)
(9)
(12)
(11)
(1)
Sub-CR clock oscillation stabilization wait time
Main CR clock oscillation stabilization wait time
Subclock oscillation stabilization wait time
Main clock oscillation stabilization wait time
(13)
(18)
(17)
Sub-CR clock oscillation stabilization wait time
(20)
(19)
Sub-CR clock mode
(15)
Subclock mode
(16)
Subclock oscillation stabilization wait time
(14)

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 sub-CR clock oscillation stabilization wait time and 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 clock mode select 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 sub-CR clock 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 clock mode select bits (SYCC2:RCS1, RCS0) are set to "00_B".
(2)			
(3)		Subclock	When the clock mode select 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 system clock control register (SYCC:SRDY) is "1_B", the device transits to subclock mode immediately after the clock mode select bits (SYCC2:RCS1, RCS0) are set to "01_B".
(4)			
(5)		Main clock	When the clock mode select 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 (STBC:MRDY) is "1_B", the device transits to main clock mode immediately after the clock mode select bits (SYCC2:RCS1, RCS0) are set to "11_B".
(6)			
(7)	Main clock	Main CR clock	When the clock mode select 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 (STBC:MCRDY) is "1_B", the device transits to main CR clock mode immediately after the clock mode select 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 clock mode select 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 clock mode select 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.
(15)		Subclock	Same as (3) and (4)
(16)			
(17)	Subclock	Main CR clock	Same as (13)
(18)		Main clock	Same as (14)
(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, time-base timer mode and watch mode.

■ Overview of Transiting to and Returning from Standby Mode

There are four standby modes: sleep mode, stop mode, time-base 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 CR 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 operating.

■ 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, time-base timer mode or watch mode, and to set an I/O port or a peripheral resource pin to high impedance in stop mode, time-base timer mode or watch mode.

See "APPENDIX D Pin States of MB95430H Series" in APPENDIX 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 select 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 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 Diagram

Figure 6.8-1 shows a standby mode state transition diagram.

Figure 6.8-1 Standby Mode State Transition Diagram

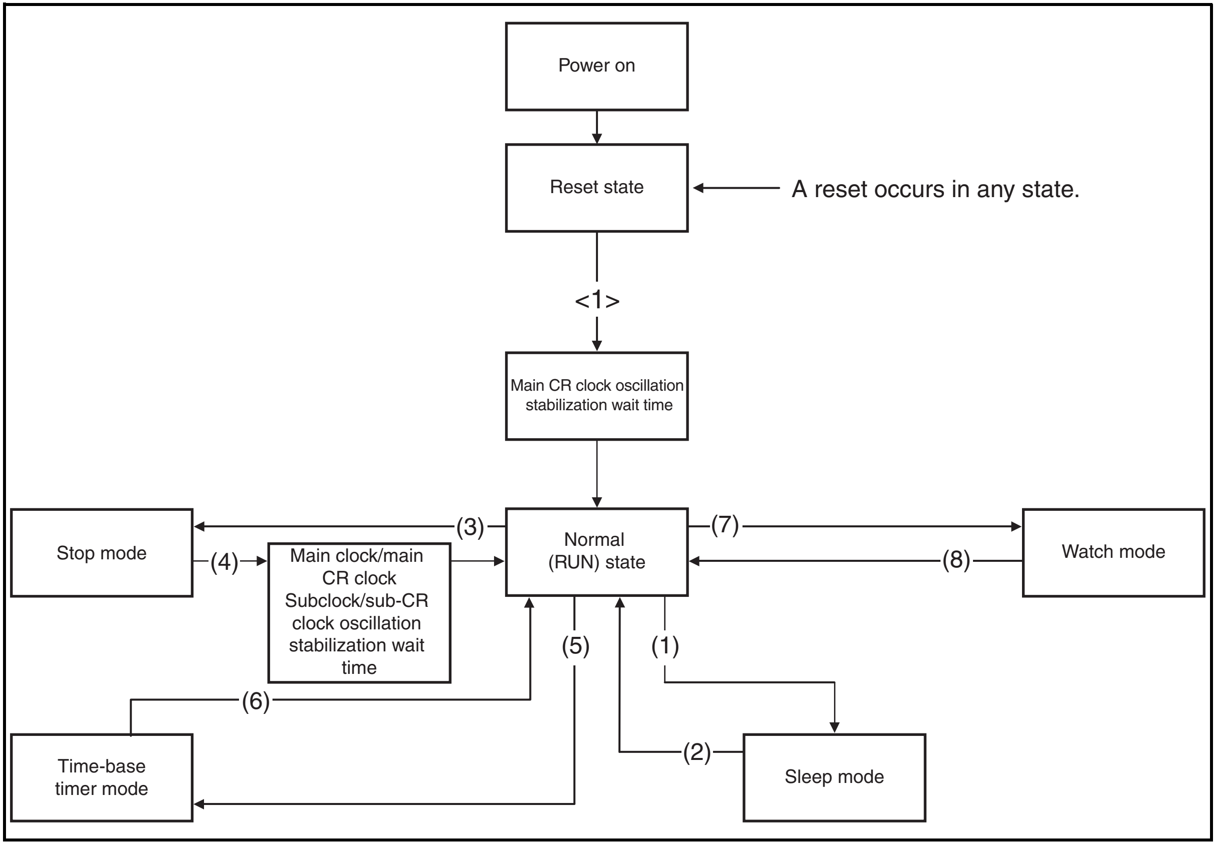

Power on
Reset state
A reset occurs in any state.
<1>
Main CR clock oscillation stabilization wait time
Stop mode
(3)
Normal (RUN) state
(7)
Watch mode
(4)
Main clock/main CR clock Subclock/sub-CR clock oscillation stabilization wait time
(8)
(5)
(1)
(6)
(2)
Time-base timer mode
Sleep mode

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 and the sub-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.
(5)	Time-base timer mode	The device transits to time-base timer mode when "1" is written to the watch bit in the standby control register (STBC:TMD) in main clock mode or main 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 30 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 30 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 time-base timer interrupt request may be generated. If the interrupt interval time of the time-base 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 time-base 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 30 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 Time-base Timer Mode

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

■ Operations in Time-base Timer Mode

The time-base timer mode is a mode in which main clock supply is stopped except the clock supply to the time-base timer. In this mode, while retaining the contents of registers and RAM existing at the point immediately before the device transits to time-base timer mode, the device stops all functions except the time-base 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 time-base timer mode, the sub-CR clock does not stop and the hardware watchdog timer operates. For details, see "CHAPTER 30 NON-VOLATILE REGISTER FUNCTION (NVR)".

● Transition to time-base timer mode

If the clock mode 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 time-base timer mode.

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

After the device transits to time-base 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 time-base timer mode

The device is released from time-base timer mode by a reset, a time-base 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 time-base timer mode by an interrupt from the watch prescaler.

Note:

If the device is released from time-base timer mode by an interrupt, a peripheral function having transited to time-base timer mode during operation resumes operating from the point at which it transited to time-base 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 time-base 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 30 NON-VOLATILE REGISTER FUNCTION (NVR)".

● Transition to watch mode

If the clock mode 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 time-base timer mode during operation resumes operating from the point at which it transited to time-base 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 watch mode.

MB95430H Series

6.9 Clock Oscillator Circuit

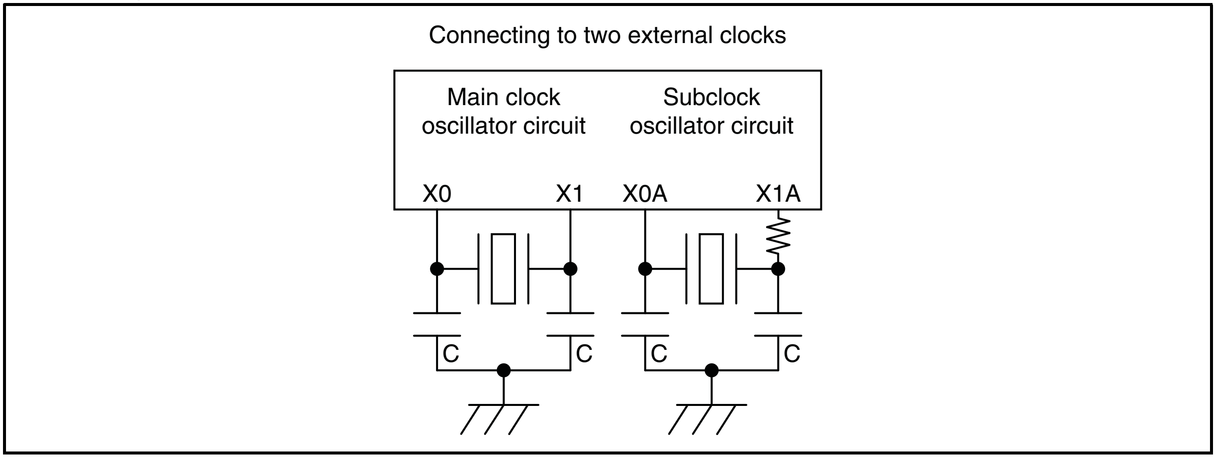
The clock oscillator circuit generates an internal clock with an oscillator connected to the clock oscillation pin or by inputting clock signals to the clock oscillation pin.

■ Clock Oscillator Circuit

● Using crystal oscillators and ceramic oscillators

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

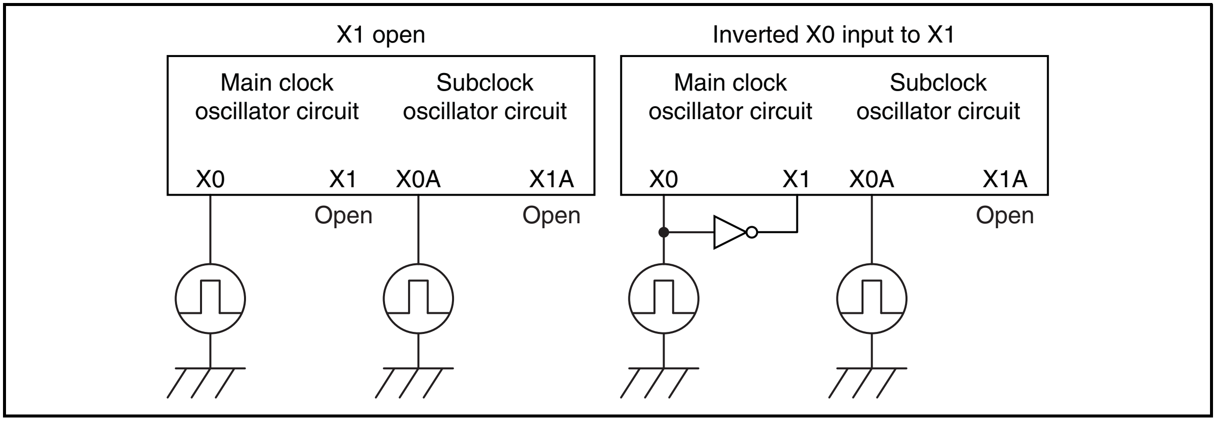
Figure 6.9-1 Sample Connection of Crystal Oscillators and Ceramic Oscillators


Connecting to two external clocks
Main clock oscillator circuit
Subclock oscillator circuit
X0
X1
X0A
X1A
C
C
C
C

● 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 the MB95430H 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


X1 open
Main clock oscillator circuit
Subclock oscillator circuit
X0
X1
X0A
X1A
Open
Open
Inverted X0 input to X1
Main clock oscillator circuit
Subclock oscillator circuit
X0
X1
X0A
X1A
Open

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 time-base 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 ($F_{CH}/2^7$, $F_{CH}/2^8$, $F_{CRH}/2^6$ or $F_{CRH}/2^7$)output from the time-base 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 ($F_{CH}/2^7$, $F_{CH}/2^8$, $F_{CRH}/2^6$ or $F_{CRH}/2^7$) of the time-base timer.

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

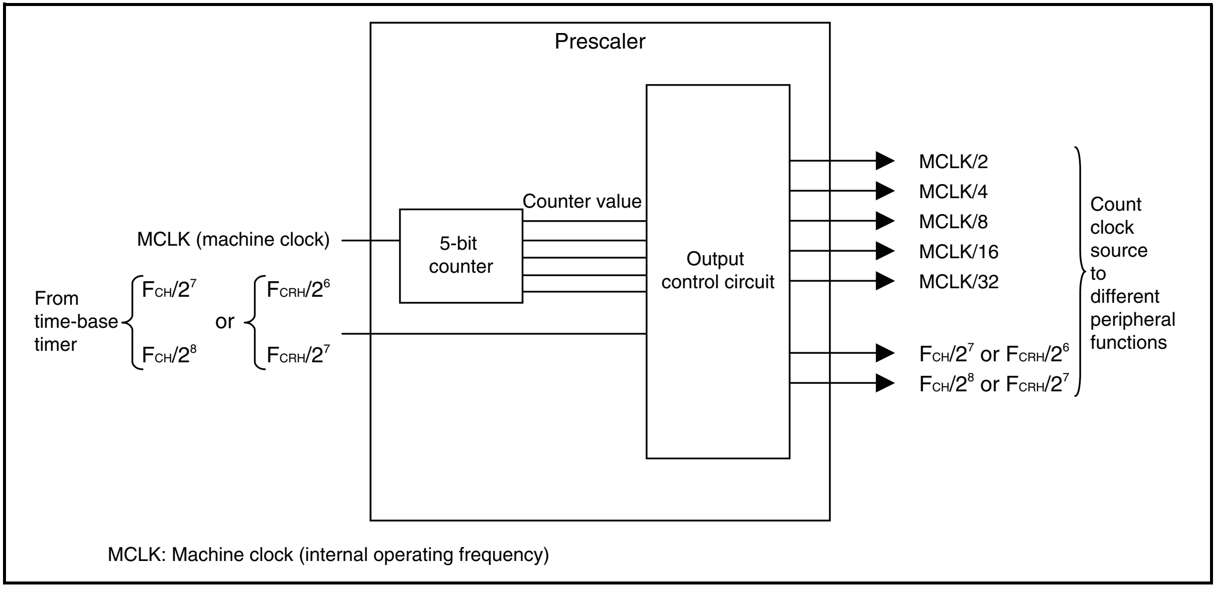
MB95430H Series

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 time-base timer ($F_{CH}/2^7$, $F_{CH}/2^8$, $F_{CRH}/2^6$ or $F_{CRH}/2^7$) and supplies it to peripheral functions.

■ Input Clock

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

■ Output Clock

The prescaler supplies clocks to the 8/16-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 time-base timer ($F_{CH}/2^7$, $F_{CH}/2^8$, $F_{CRH}/2^6$ or $F_{CRH}/2^7$), and supplies them to different peripheral functions. The prescaler keeps operating while the machine clock and the clocks from the time-base timer are being supplied.

Table 6.12-1 and Table 6.12-2 list the count clock sources generated by the prescaler.

Table 6.12-1 Count Clock Sources Generated by Prescaler (F_{CH})

Count clock source frequency	Frequency (F_{CH} = 20 MHz, MCLK = 10 MHz)	Frequency (F_{CH} = 32 MHz, MCLK = 16 MHz)	Frequency (F_{CH} = 32.5 MHz, MCLK = 16.25 MHz)
MCLK/2	5 MHz	8 MHz	8.125 MHz
MCLK/4	2.5 MHz	4 MHz	4.0625 MHz
MCLK/8	1.25 MHz	2 MHz	2.0313 MHz
MCLK/16	0.625 MHz	1 MHz	1.0156 MHz
MCLK/32	0.3125 MHz	0.5 MHz	0.5078 MHz
$F_{CH}/2^7$	156.25 kHz	250 kHz	253.9 kHz
$F_{CH}/2^8$	78.128 kHz	125 kHz	126.95 kHz

Table 6.12-2 Count Clock Sources Generated by Prescaler (F_{CRH})

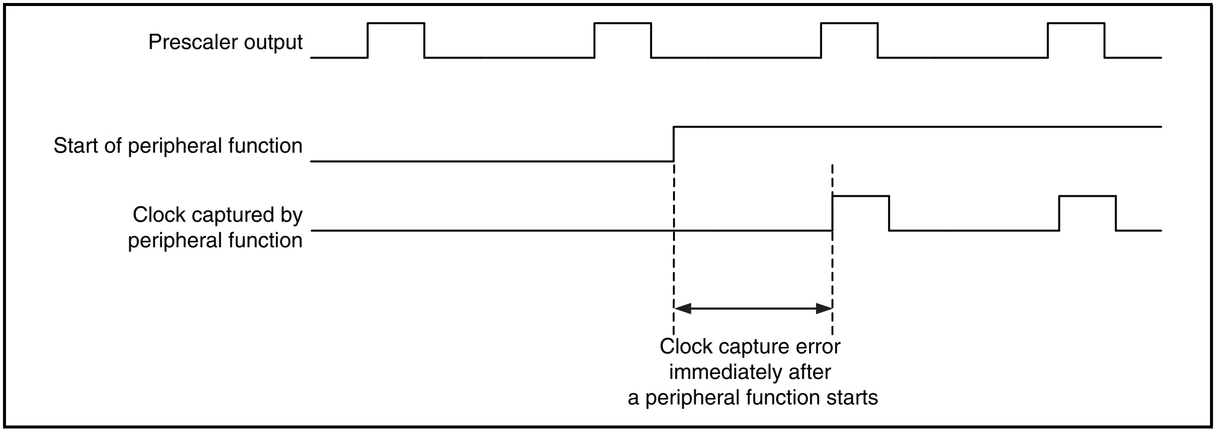
Count clock source frequency	Frequency (F_{CRH} = 1 MHz, MCLK = 1 MHz)	Frequency (F_{CRH} = 8 MHz, MCLK = 8 MHz)	Frequency (F_{CRH} = 10 MHz, MCLK = 10 MHz)	Frequency (F_{CRH} = 12.5 MHz, MCLK = 12.5 MHz)
MCLK/2	500 kHz	4 MHz	5 MHz	6.25 MHz
MCLK/4	250 kHz	2 MHz	2.5 MHz	3.125 MHz
MCLK/8	125 kHz	1 MHz	1.25 MHz	1.5625 MHz
MCLK/16	62.5 kHz	0.5 MHz	0.625 MHz	0.78125 MHz
MCLK/32	31.25 kHz	0.25 MHz	0.3125 MHz	0.390625 MHz
$F_{CH}/2^7$	15.625 kHz	125 kHz	156.25 kHz	195.3125 kHz
$F_{CH}/2^8$	7.8125 kHz	62.5 kHz	78.125 kHz	97.65625 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 time-base 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 will occur, depending on the output value of the prescaler.

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


Prescaler output
Start of peripheral function
Clock captured by peripheral function
Clock capture error 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 ($\overline{\text{RST}}$).

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.

The low-voltage detection reset circuit is only available in certain products. For details, see "1.2 Product Line-up of MB95430H Series".

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 21 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.

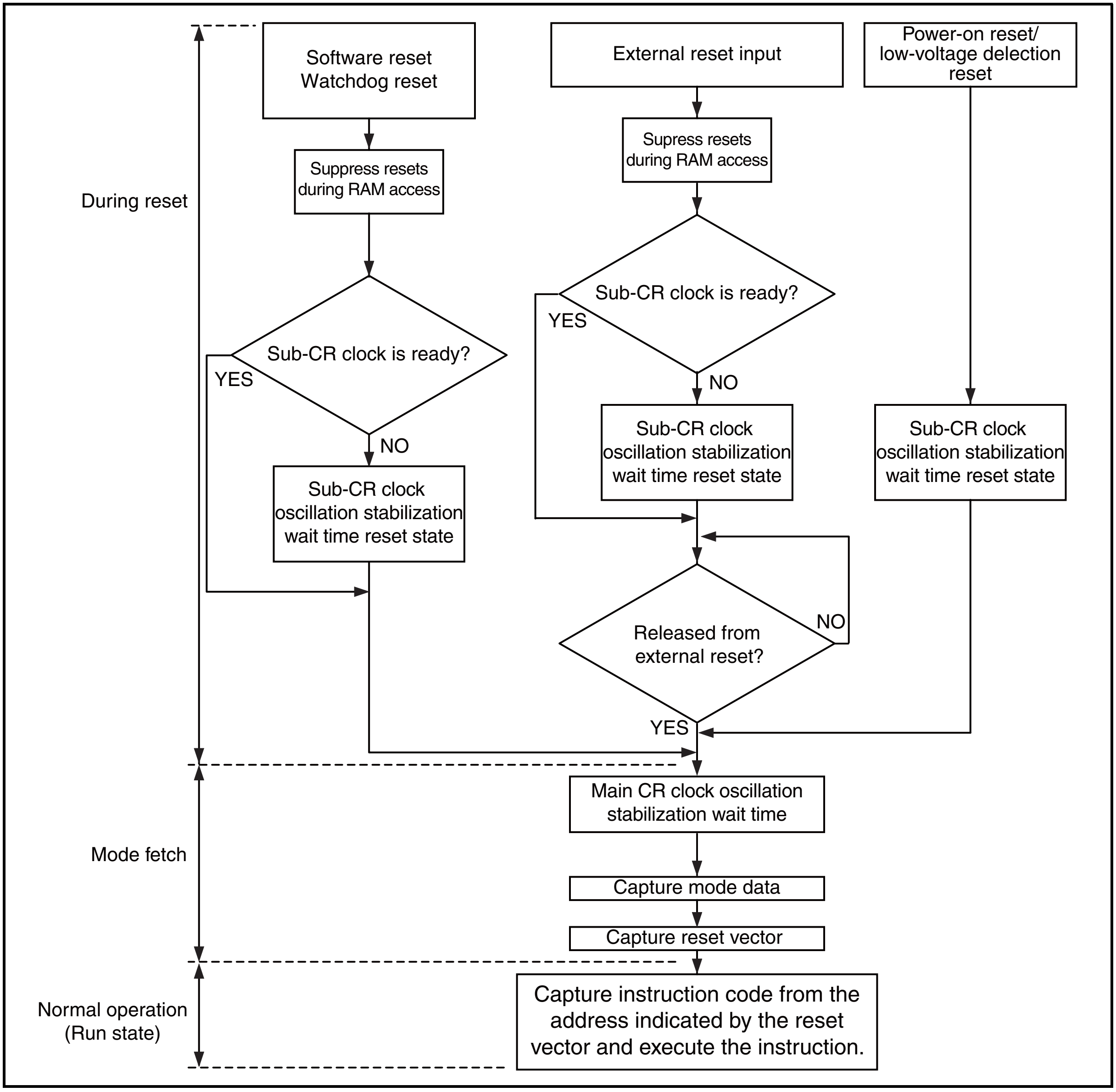
■ Reset Output

When the reset input function is effective and the reset output function is effective, the $\overline{\text{RST}}$ pin outputs "L" level while resetting it. However, the function to output "L" level is not provided for external reset in the reset pin.

The reset input function and the reset output function setting Please refer to "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

■ 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.

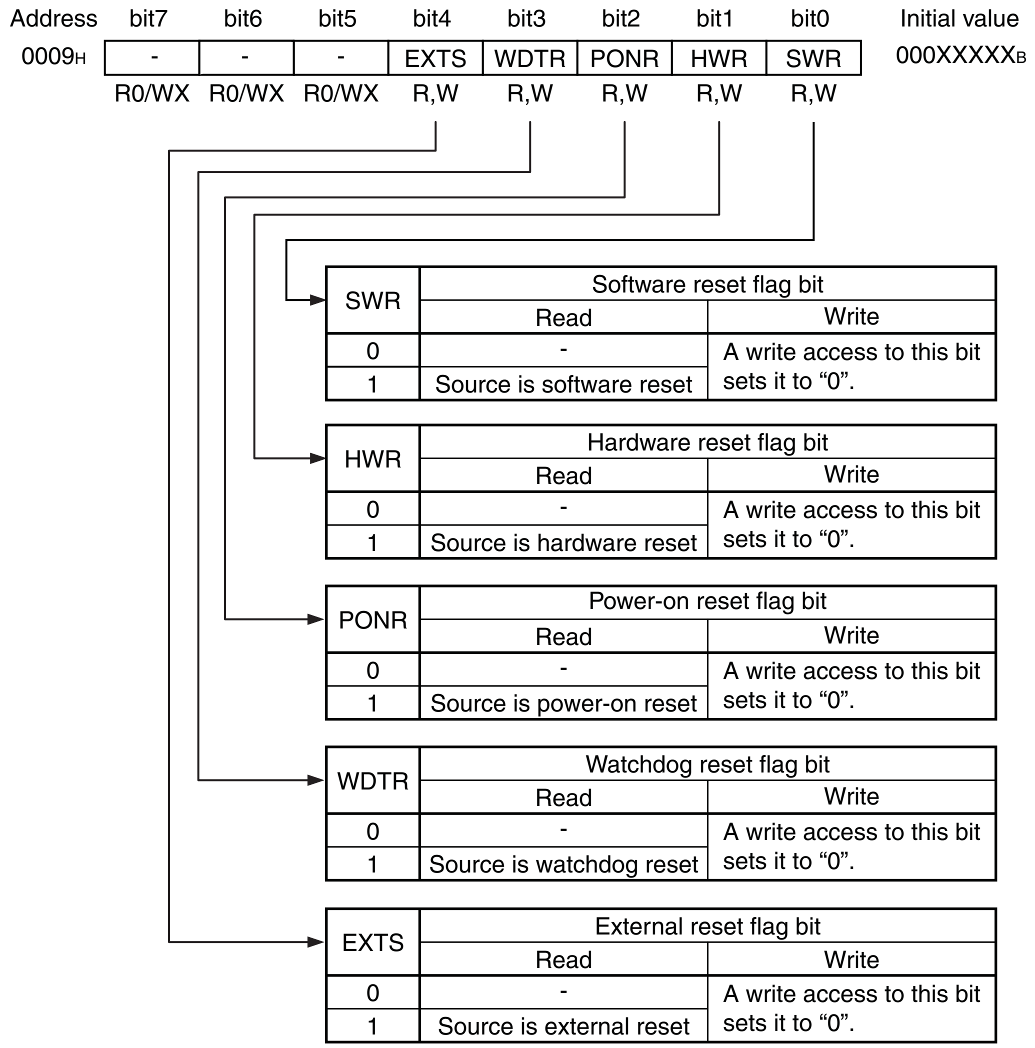
For details of the states of all pins during a reset, see "APPENDIX D Pin States of MB95430H Series" in APPENDIX.

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)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0009_H	-	-	-	EXTS	WDTR	PONR	HWR	SWR	$000XXXXX_B$
	R0/WX	R0/WX	R0/WX	R,W	R,W	R,W	R,W	R,W	

SWR	Software reset flag bit	
	Read	Write
0	-	A write access to this bit sets it to “0”.
1	Source is software reset	

HWR	Hardware reset flag bit	
	Read	Write
0	-	A write access to this bit sets it to “0”.
1	Source is hardware reset	

PONR	Power-on reset flag bit	
	Read	Write
0	-	A write access to this bit sets it to “0”.
1	Source is power-on reset	

WDTR	Watchdog reset flag bit	
	Read	Write
0	-	A write access to this bit sets it to “0”.
1	Source is watchdog reset	

EXTS	External reset flag bit	
	Read	Write
0	-	A write access to this bit sets it to “0”.
1	Source is external reset	

R,W	:	Readable/writable (The read value is different from the write value.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
X	:	Indeterminate

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

Bit name		Function
bit7 to bit5	Undefined bits	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 or a write access (writing 0 or 1) to this bit clears it to "0".
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 or a write access (writing 0 or 1) to this bit clears it to "0".
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 or a write access (writing 0 or 1) to this bit clears it to "0".
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 bit2 to bit4 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 or a write access (writing 0 or 1) to this bit clears it to "0".
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 or a write access (writing 0 or 1) to this bit or a power-on reset clears it to "0".

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^2MC-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 interrupt requests of the same level (generated simultaneously)
	Upper	Lower		
IRQ00	FFFA$_H$	FFFB$_H$	L00 [1:0]	Highest
IRQ01	FFF8$_H$	FFF9$_H$	L01 [1:0]	
IRQ02	FFF6$_H$	FFF7$_H$	L02 [1:0]	
IRQ03	FFF4$_H$	FFF5$_H$	L03 [1:0]	
IRQ04	FFF2$_H$	FFF3$_H$	L04 [1:0]	
IRQ05	FFF0$_H$	FFF1$_H$	L05 [1:0]	
IRQ06	FFEE$_H$	FFEF$_H$	L06 [1:0]	
IRQ07	FFEC$_H$	FFED$_H$	L07 [1:0]	
IRQ08	FFEA$_H$	FFEB$_H$	L08 [1:0]	
IRQ09	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	00079$_H$	L03[1:0]	L02[1:0]	L01[1:0]	L00[1:0]	R/W	11111111$_B$
ILR1	0007A$_H$	L07[1:0]	L06[1:0]	L05[1:0]	L04[1:0]	R/W	11111111$_B$
ILR2	0007B$_H$	L11[1:0]	L10[1:0]	L09[1:0]	L08[1:0]	R/W	11111111$_B$
ILR3	0007C$_H$	L15[1:0]	L14[1:0]	L13[1:0]	L12[1:0]	R/W	11111111$_B$
ILR4	0007D$_H$	L19[1:0]	L18[1:0]	L17[1:0]	L16[1:0]	R/W	11111111$_B$
ILR5	0007E$_H$	L23[1:0]	L22[1:0]	L21[1:0]	L20[1:0]	R/W	11111111$_B$

The interrupt level setting registers assign 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

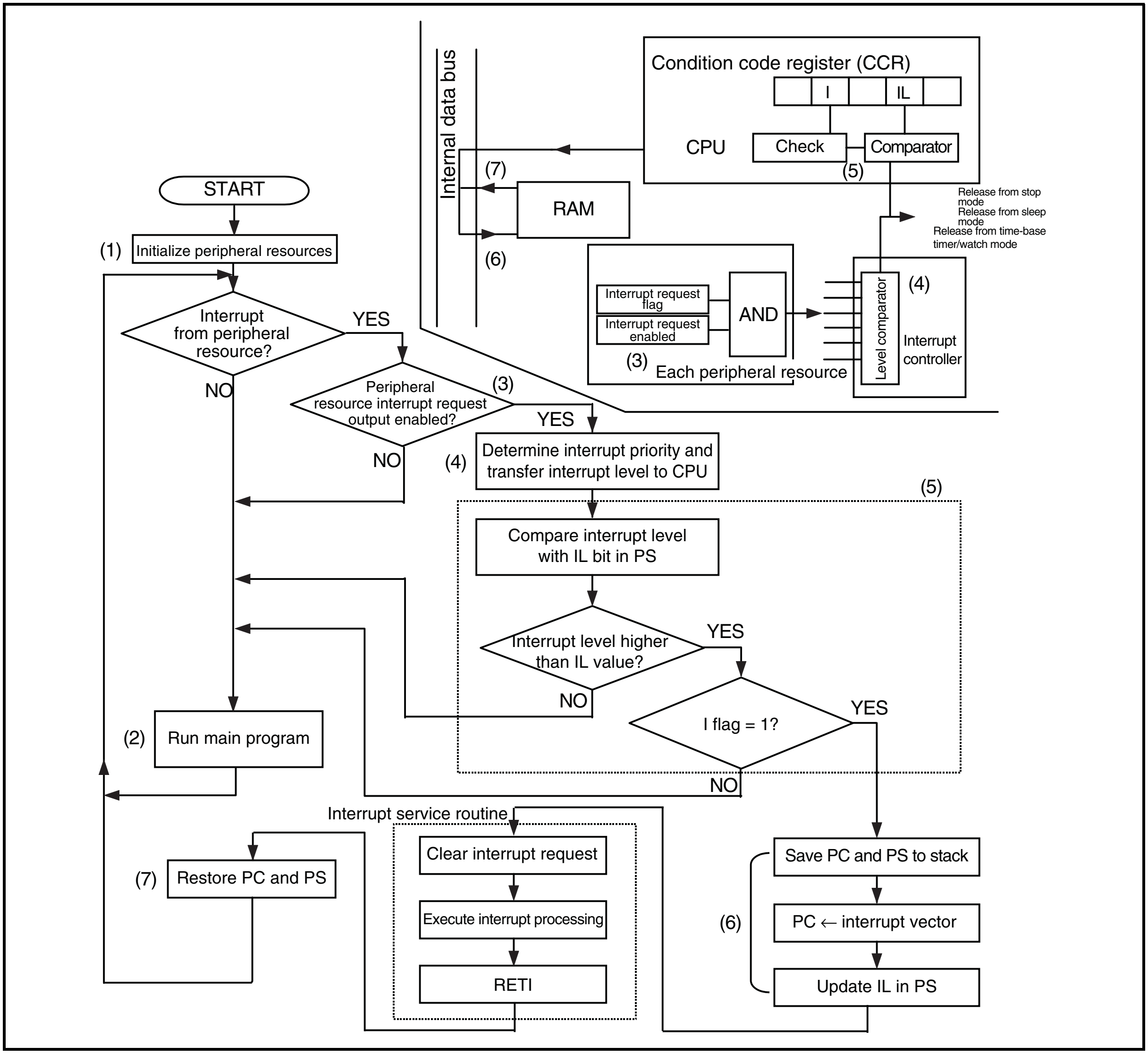
When an interrupt request is made by a peripheral resource, the interrupt controller notifies the CPU of the interrupt level of that interrupt request. When the CPU is ready to accept interrupts, it halts the program it is executing and executes an interrupt service routine.

■ 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


Internal data bus
Condition code register (CCR)
I
IL
CPU
Check
Comparator
(5)
(7)
RAM
(6)
Release from stop mode
Release from sleep mode
Release from time-base timer/watch mode
START
(1)
Initialize peripheral resources
Interrupt request flag
Interrupt request enabled
AND
(3)
Each peripheral resource
Level comparator
(4)
Interrupt controller
Interrupt from peripheral resource?
YES
NO
Peripheral resource interrupt request output enabled?
(3)
YES
NO
(4)
Determine interrupt priority and transfer interrupt level to CPU
(5)
Compare interrupt level with IL bit in PS
Interrupt level higher than IL value?
YES
NO
I flag = 1?
YES
NO
(2)
Run main program
Interrupt service routine
Clear interrupt request
Execute interrupt processing
RETI
(7)
Restore PC and PS
Save PC and PS to stack
(6)
PC ← interrupt vector
Update IL in PS

(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.

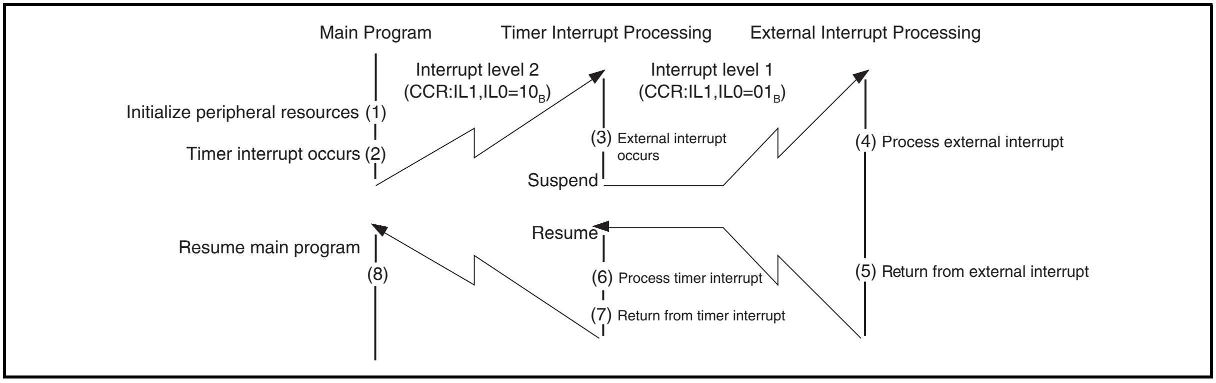
■ Nested Interrupts

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


Main Program
Timer Interrupt Processing
External Interrupt Processing
Interrupt level 2
(CCR:IL1,IL0=10$_B$)
Interrupt level 1
(CCR:IL1,IL0=01$_B$)
Initialize peripheral resources (1)
Timer interrupt occurs (2)
(3) External interrupt occurs
Suspend
(4) Process external interrupt
Resume
Resume main program
(8)
(6) Process timer interrupt
(7) Return from timer interrupt
(5) Return from external interrupt

- 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

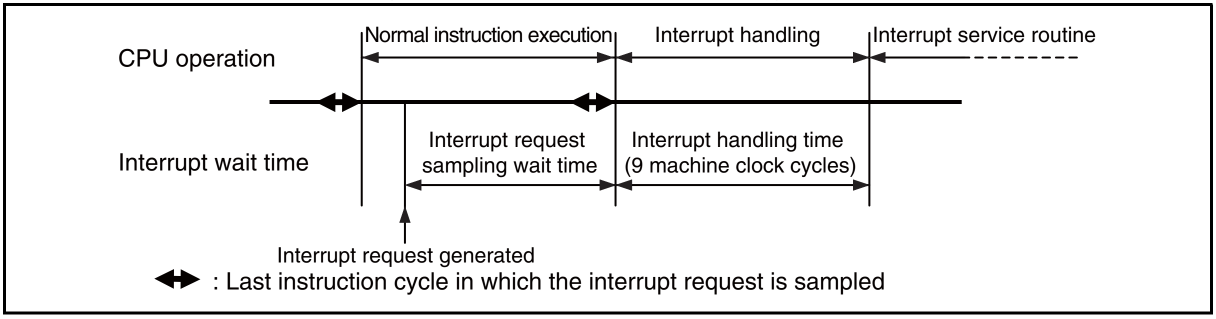
The CPU decides whether an interrupt request has occurred by sampling the interrupt request during the last cycle of an instruction. Therefore, the CPU cannot recognize interrupt requests while executing an instruction. This sampling wait time reaches its maximum when an interrupt request occurs immediately 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


CPU operation
Normal instruction execution
Interrupt handling
Interrupt service routine
Interrupt wait time
Interrupt request sampling wait time
Interrupt handling time (9 machine clock cycles)
Interrupt request generated
◆◆ : Last instruction cycle in which the interrupt request is sampled

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

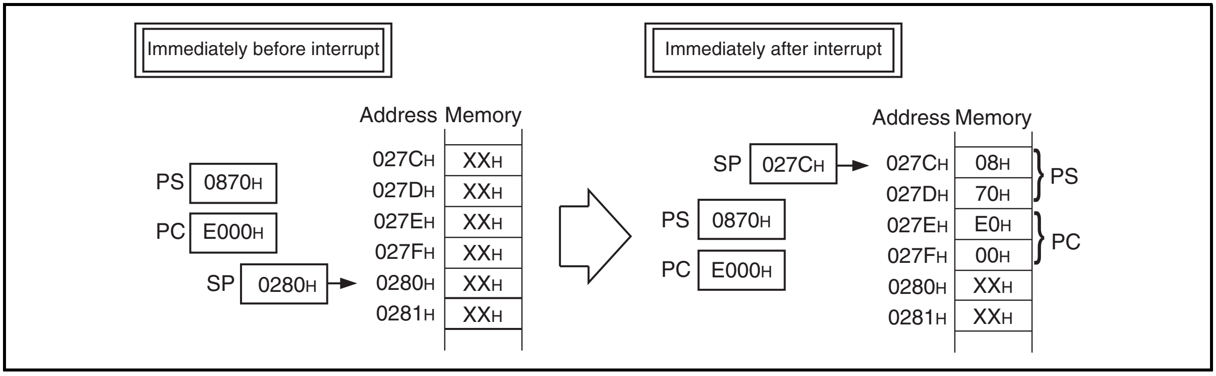
This section describes how the contents of a register are saved and restored during interrupt processing.

■ 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.

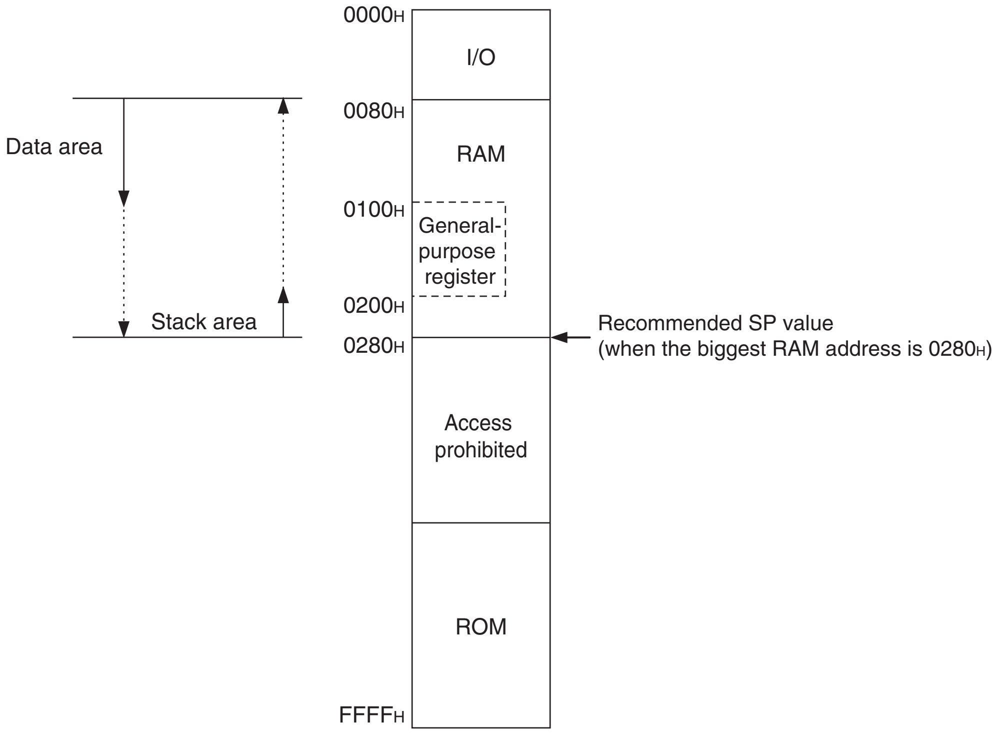
■ Interrupt Processing Stack Area

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 lists the registers for each port.

Table 9.1-1 List of Port Registers

Register name		Read/Write	Initial value
Port 0 data register	(PDR0)	R, RM/W	00000000_B
Port 0 direction register	(DDR0)	R/W	00000000_B
Port 1 data register	(PDR1)	R, RM/W	00000000_B
Port 1 direction register	(DDR1)	R/W	00000000_B
Port 6 data register	(PDR6)	R, RM/W	00000000_B
Port 6 direction register	(DDR6)	R/W	00000000_B
Port 7 data register	(PDR7)	R, RM/W	00000000_B
Port 7 direction register	(DDR7)	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
A/D input disable register (upper)	(AIDRH)	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 lists the port 0 pins.

Table 9.2-1 Port 0 Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P00/INT00/ AN00	P00: General-purpose I/O	INT00: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN00: Analog input				
P01/INT01/ AN01/BZ	P01: General-purpose I/O	INT01: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN01: Analog input				
		BZ: Buzzer output				
P02/INT02/ AN02/UCK	P02: General-purpose I/O	INT02: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN02: Analog input				
		UCK: UART/SIO clock I/O				
P03/INT03/ AN03/UO/ SDA	P03: General-purpose I/O	INT03: External interrupt input	Hysteresis/ CMOS/ Analog	CMOS	-	❍
		AN03: Analog input				
		UO: UART/SIO data output				
		SDA: I^{2}C data I/O			❍	
P04/INT04/ AN04/UI// SCL	P04: General-purpose I/O	INT04: External interrupt input	Hysteresis/ CMOS/ Analog	CMOS	-	❍
		AN04: Analog input				
		UI: UART/SIO data output				
		SCL: I^{2}C clock I/O			❍	
P05/INT05/ AN05/TO0	P05: General-purpose I/O	INT05: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN05: Analog input				
		TO0: 8/16-bit composite timer ch. 0 output				
P06/INT06/ AN06/TO1	P06: General-purpose I/O	INT06: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN06: Analog input				
		TO1: 8/16-bit composite timer ch. 1 output				
P07/INT07/ AN07/EC0	P07: General-purpose I/O	INT07: External interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN07: Analog input				
		EC0: 8/16-bit composite timer ch. 0 clock input				

OD: Open drain, PU: Pull-up

■ Block Diagrams of Port 0

Figure 9.2-1 Block Diagram of P00 and P07

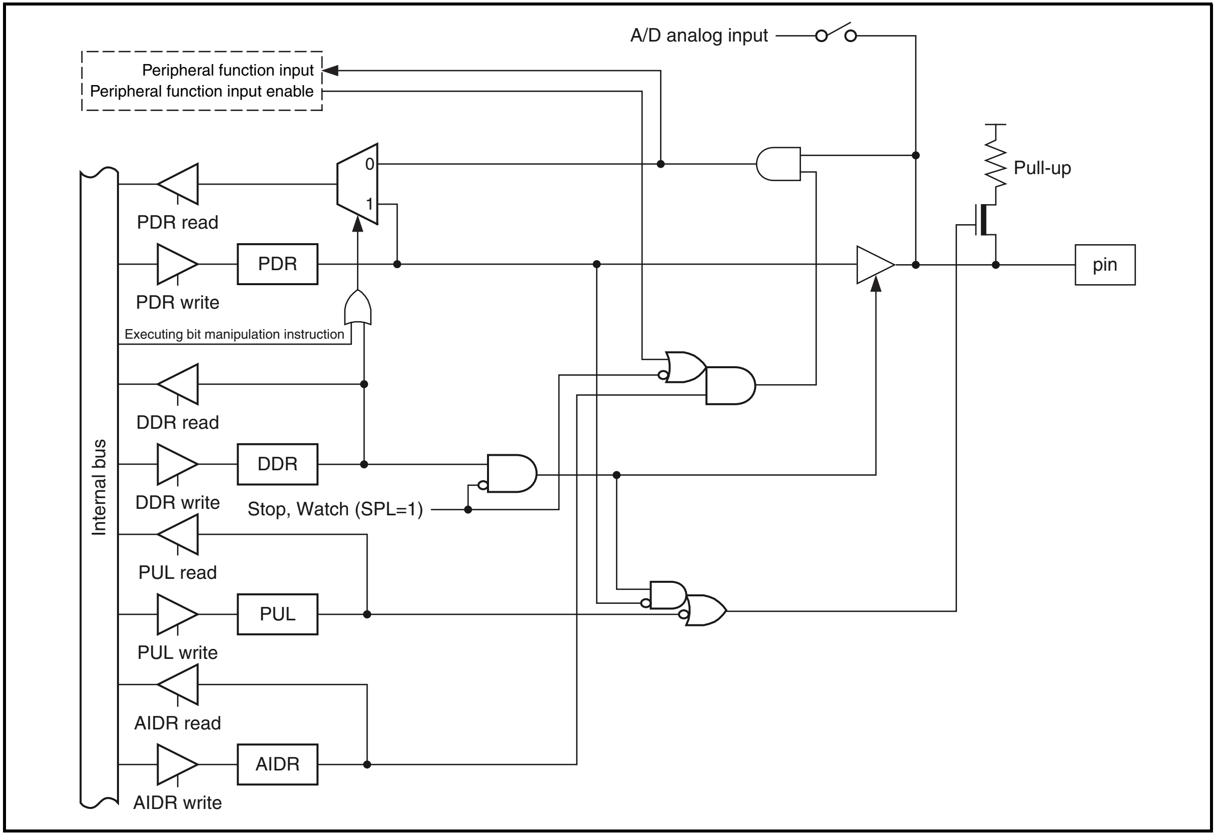


Figure 9.2-2 Block Diagram of P01, P02, P05 and P06

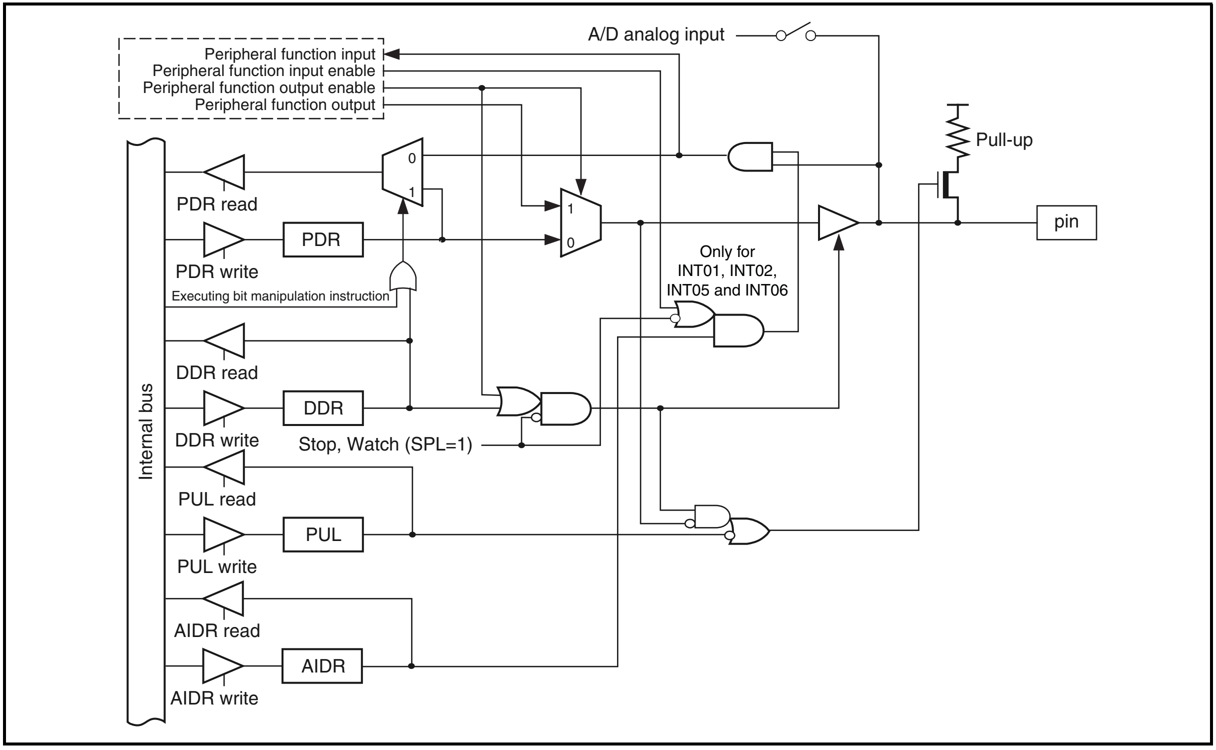
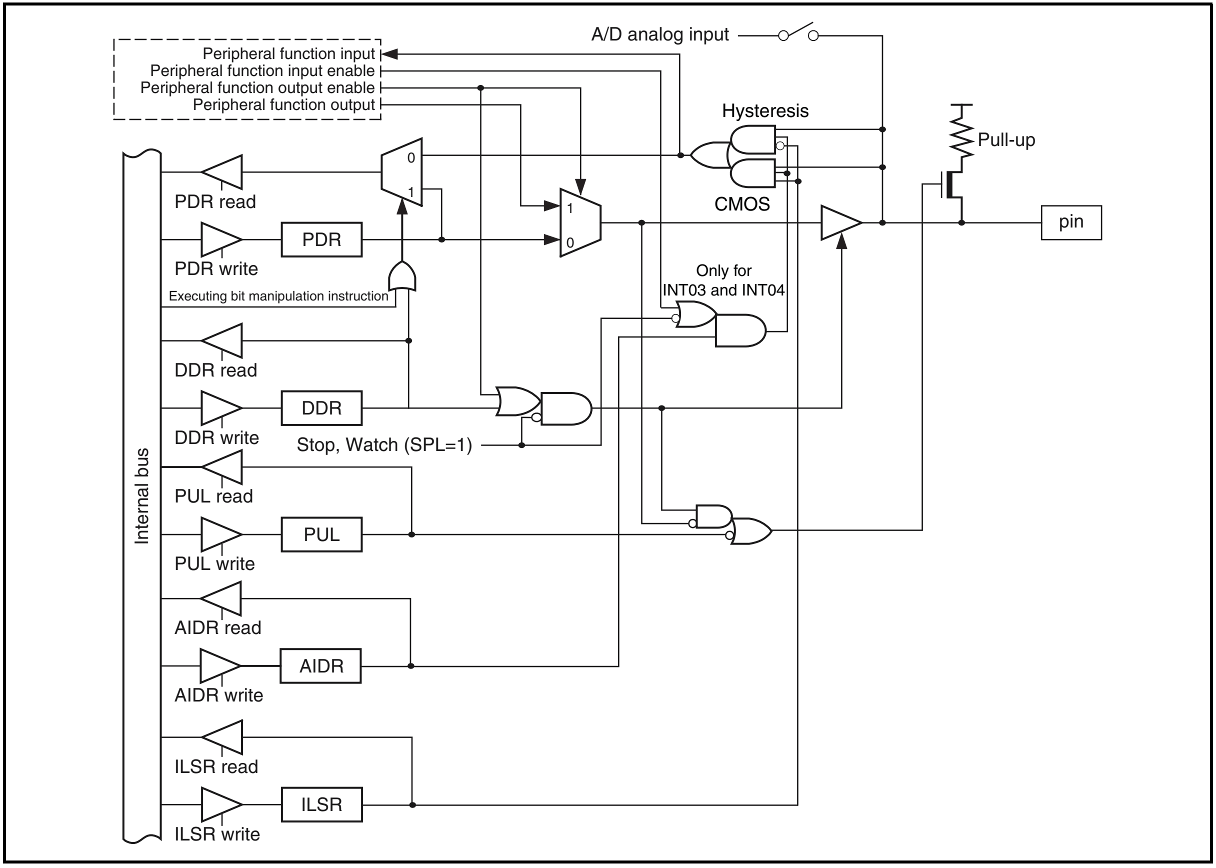


Figure 9.2-3 Block Diagram of P03 and P04



Note:

When P03 and P04 are used as SDA and SCL respectively, they become N-ch open drain pins.

9.2.1 Port 0 Registers

This section describes the registers of port 0.

■ Port 0 Register Functions

Table 9.2-2 lists the functions of the port 0 register.

Table 9.2-2 Port 0 Register Functions

Register abbr.	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 9.2-3 shows the correspondence between port 0 pins and each register bit.

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

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

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 (RMW) type of instruction 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 (RMW) type of instruction 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 (RMW) type of instruction 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 INT00), 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 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 P03 or P04 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P03 or P04 should become the hysteresis input level.
- For pins other than P03 or P04, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.
- When changing the input level of P03 or P04, ensure that the peripheral function (UART/ I^{2}C/External interrupt) has been stopped.

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

Table 9.2-4 Pin State of Port 0

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

MB95430H Series

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)
- Input level select register (ILSR)

■ Port 1 Pin

Port 1 has one I/O pin.

Table 9.3-1 lists the port 1 pin.

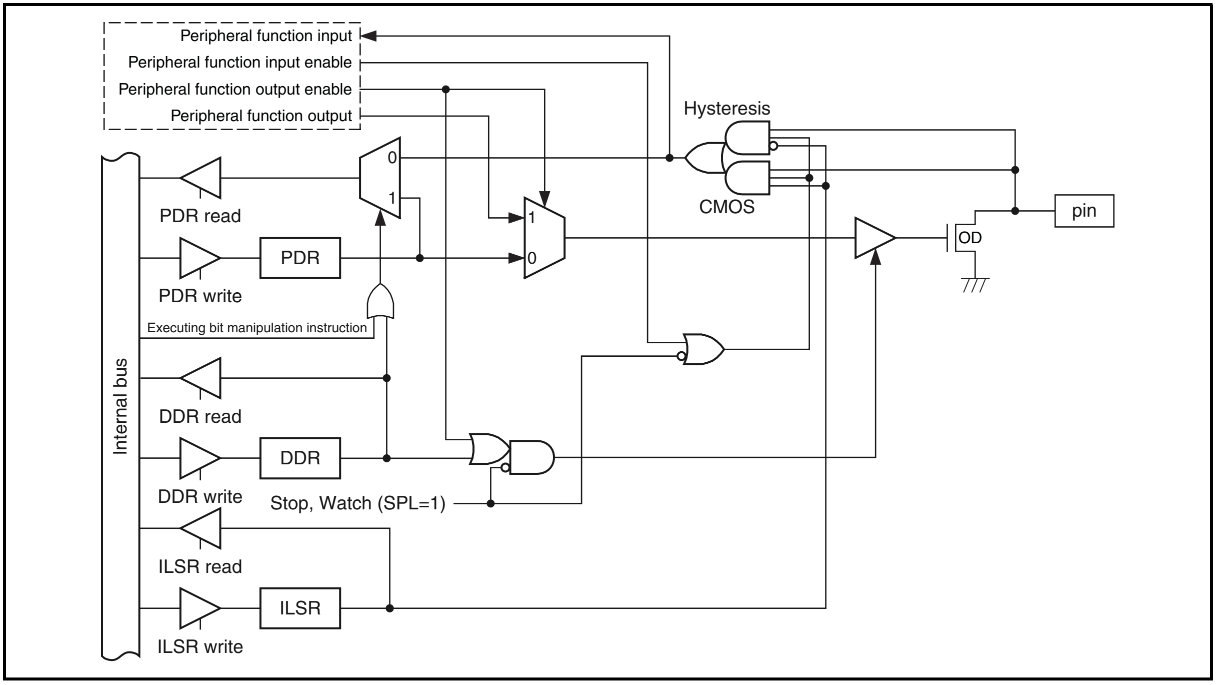
Table 9.3-1 Port 1 Pin

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P12/EC0/UI/SCL/DBG	P12: General-purpose I/O	EC0: 8/16-bit composite timer ch. 0 clock input	Hysteresis/CMOS	CMOS	❍	-
		UI: UART/SIO data input				
		SCL: I^{2}C clock I/O				
		DBG: On-chip debug communication pin				

OD: Open drain, PU: Pull-up

■ Block Diagram of Port 1

Figure 9.3-1 Block Diagram of Port 1


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
0
1
1
0
pin
OD
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

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 abbr.	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		
ILSR	0	Hysteresis input level selected		
	1	CMOS input level selected		

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

Table 9.3-3 shows 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								
ILSR	-	-	-	-	-	bit0	-	-

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 (RMW) type of instruction 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 (RMW) type of instruction 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 (RMW) type of instruction is used to read the PDR register, the PDR register value is returned.

● Operation of the input level select register

- Setting the bit in ILSR to "1" changes only P12 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P12 should become the hysteresis input level.
- When changing the input level of P12, ensure that the peripheral function (UART/I^{2}C/

8/16bit composite timer ch. 0 clock input) has been stopped.

● 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 in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

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.

■ 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)
- A./D input disable register upper (AIDRH)
- Input level select register (ILSR)

■ Port 6 Pins

Port 6 has eight I/O pins.

Table 9.4-1 lists the port 6 pins.

Table 9.4-1 Port 6 Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P60/ OPAMP_P	P60: General-purpose I/O	OPAMP_P: OPAMP positive input	Hysteresis/ Analog input	CMOS	-	-
P61/ OPAMP_N	P61: General-purpose I/O	OPAMP_N: OPAMP negative input	Hysteresis/ Analog input	CMOS	-	-
P62/ OPAMP_O	P62: General-purpose I/O	OPAMP_O: OPAMP output	Hysteresis	CMOS/ Analog	-	-
P63/ CMP2_P/ AN12	P63: General-purpose I/O	CMP2_P: Comparator positive input	Hysteresis/ Analog input	CMOS	-	-
		AN12: Analog input				
P64/ CMP2_N/ AN13	P64: General-purpose I/O	CMP2_N: Comparator negative input	Hysteresis/ Analog input	CMOS	-	-
		AN13: Analog input				
P65/ CMP3_O/ UO/SDA	P65: General-purpose I/O	CMP3_O: Comparator output	Hysteresis/ CMOS	CMOS	-	-
		UO: UART/SIO data output				
		SDA: I^{2}C data I/O			❍	
P66/ CMP3_P/ AN14	P66: General-purpose I/O	CMP3_P: Comparator positive input	Hysteresis/ Analog input	CMOS	-	-
		AN1: Analog input				
P67/ CMP3_N/ AN15	P67: General-purpose I/O	CMP3_N: Comparator negative input	Hysteresis/ Analog input	CMOS	-	-
		AN15: Analog input				

OD: Open drain, PU: Pull-up

■ Block Diagrams of Port 6

Figure 9.4-1 Block Diagram of P60 and P61

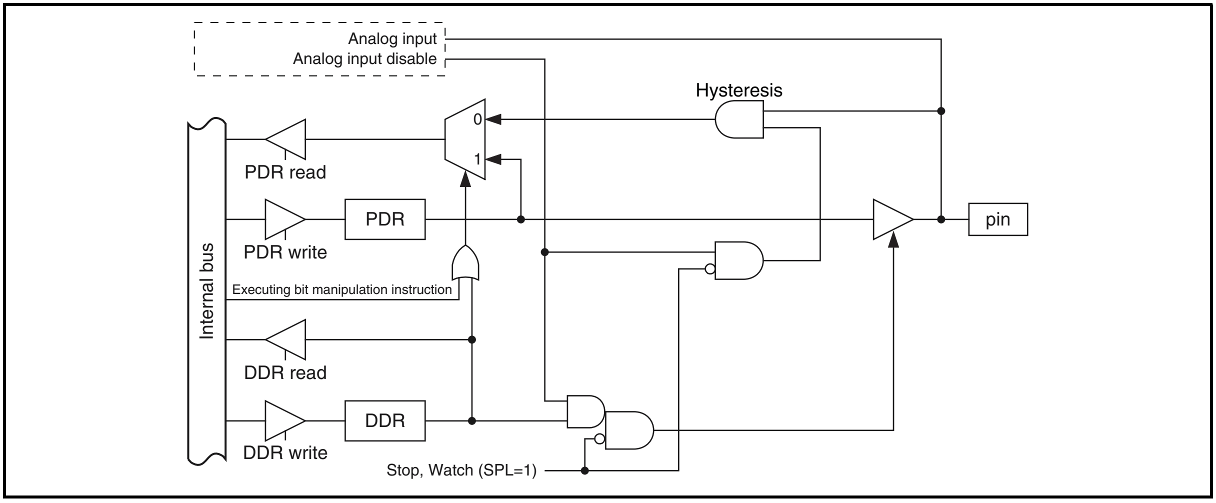


Figure 9.4-2 Block Diagram of P62

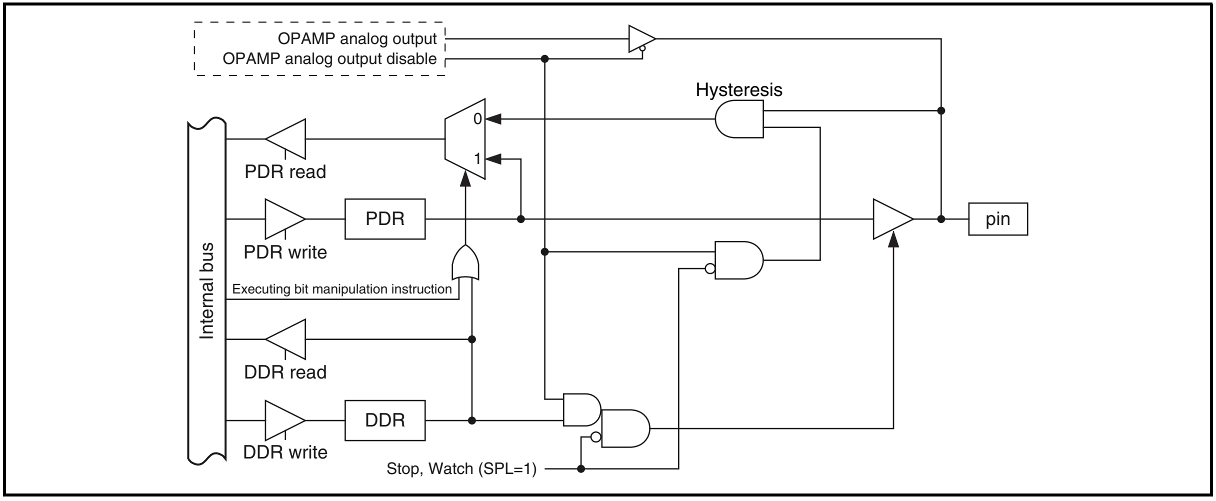


Figure 9.4-3 Block Diagram of P63, P64, P66 and P67

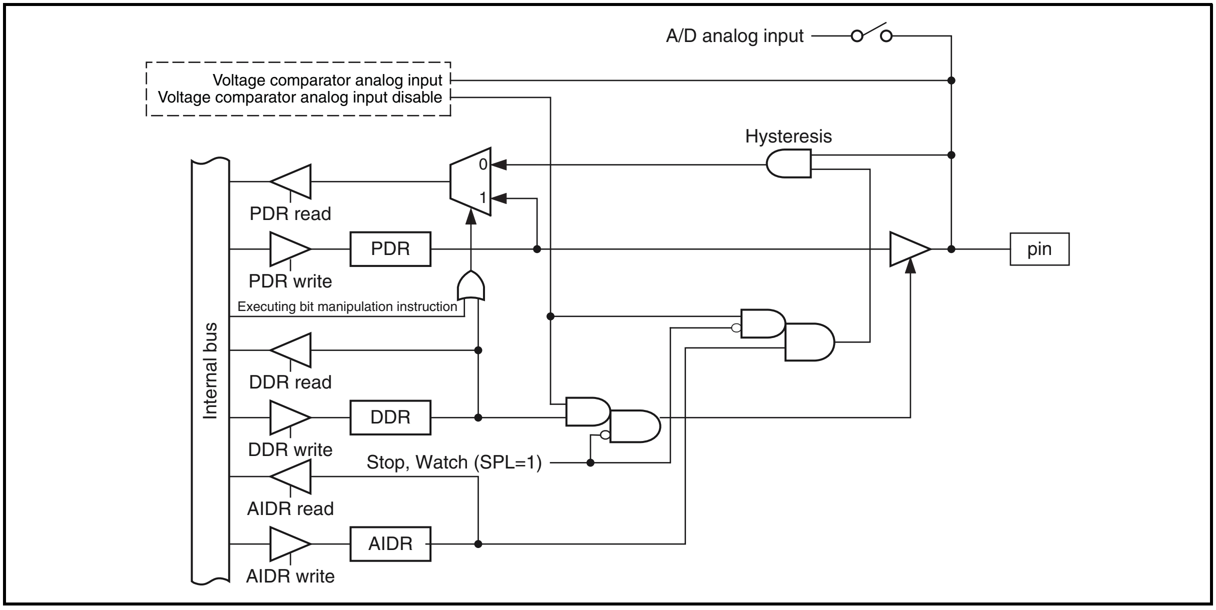
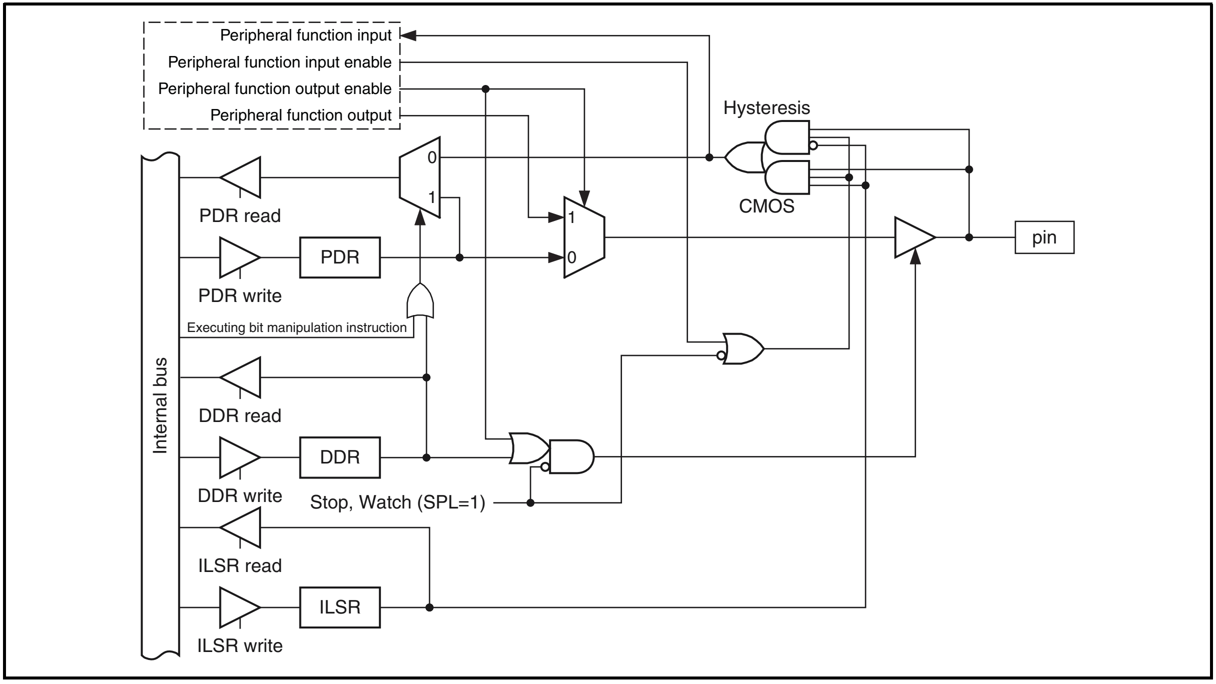


Figure 9.4-4 Block Diagram of P65


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
0
1
1
0
pin
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

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 abbr.	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		
AIDRH	0	Analog input enabled		
	1	Port input enabled		
ILSR	0	Hysteresis input level selected		
	1	CMOS input level selected		

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

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

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

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.
- When using an analog input shared pin as an input port, set the corresponding bit in the A/D input disable register upper (AIDRH) 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 (RMW) type of instruction 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 (RMW) type of instruction 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 (RMW) type of instruction 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 upper (AIDRH) 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.
- 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 AIDRH 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 of the input level select register

- Setting the bit in ILSR to "1" changes P65 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P65 should become the hysteresis input level.
- For pins other than P65, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.
- When changing the input level of P65, ensure that the peripheral function (I^2C) has been stopped.

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 in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

9.5 Port 7

Port 7 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 7 Configuration

Port 7 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 7 data register (PDR7)
- Port 7 direction register (DDR7)
- A/D input disable register upper (AIDRH)

■ Port 7 Pins

Port 7 has seven I/O pins.

Table 9.4-1 lists the port 7 pins.

Table 9.5-1 Port 7 Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P70/ CMP0_O/ OUT0/TRG	P70: General-purpose I/O	CMP0_O: Comparator output OUT0: Output compare 0 output TRG: 16-bit PPG trigger input	Hysteresis	CMOS	-	-
P71/ CMP0_P/ AN08	P71: General-purpose I/O	CMP0_P: Comparator positive input AN08:Analog input	Hysteresis/ Analog input	CMOS	-	-
P72/ CMP0_N/ AN09	P72: General-purpose I/O	CMP0_N: Comparator negative input AN09:Analog input	Hysteresis/ Analog input	CMOS	-	-
P73/ CMP1_O/ OUT1/PPG	P73: General-purpose I/O	CMP1_O: Comparator output OUT1: Output compare 1 output PPG: 16-bit PPG output	Hysteresis	CMOS	-	-
P74/ CMP1_P/ AN10	P74: General-purpose I/O	CMP1_P: Comparator positive input AN10:Analog input	Hysteresis/ Analog input	CMOS	-	-
P75/ CMP1_N/ AN11	P75: General-purpose I/O	CMP1_N: Comparator negative input AN11:Analog input	Hysteresis/ Analog input	CMOS	-	-
P76/ CMP2_O/ UCK	P76: General-purpose I/O	CMP2_O: Comparator output UCK: UART/SIO clock I/O	Hysteresis	CMOS	-	-

OD: Open drain, PU: Pull-up

■ Block Diagrams of Port 7

Figure 9.5-1 Block Diagram of P73

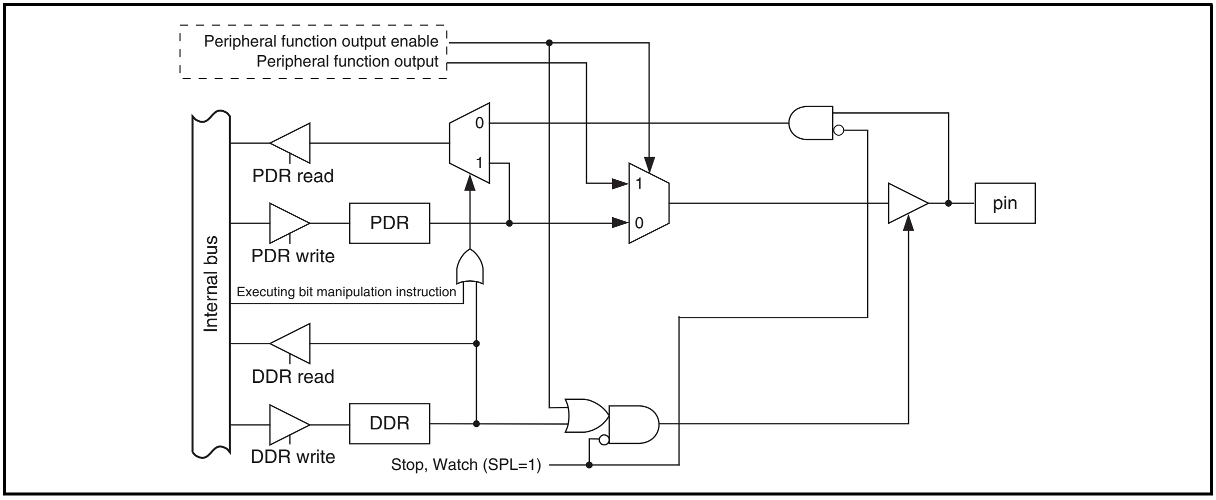

Peripheral function output enable
Peripheral function output
0
1
1
0
pin
PDR read
PDR
PDR write
Internal bus
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)

Figure 9.5-2 Block Diagram of P71, P72, P74 and P75

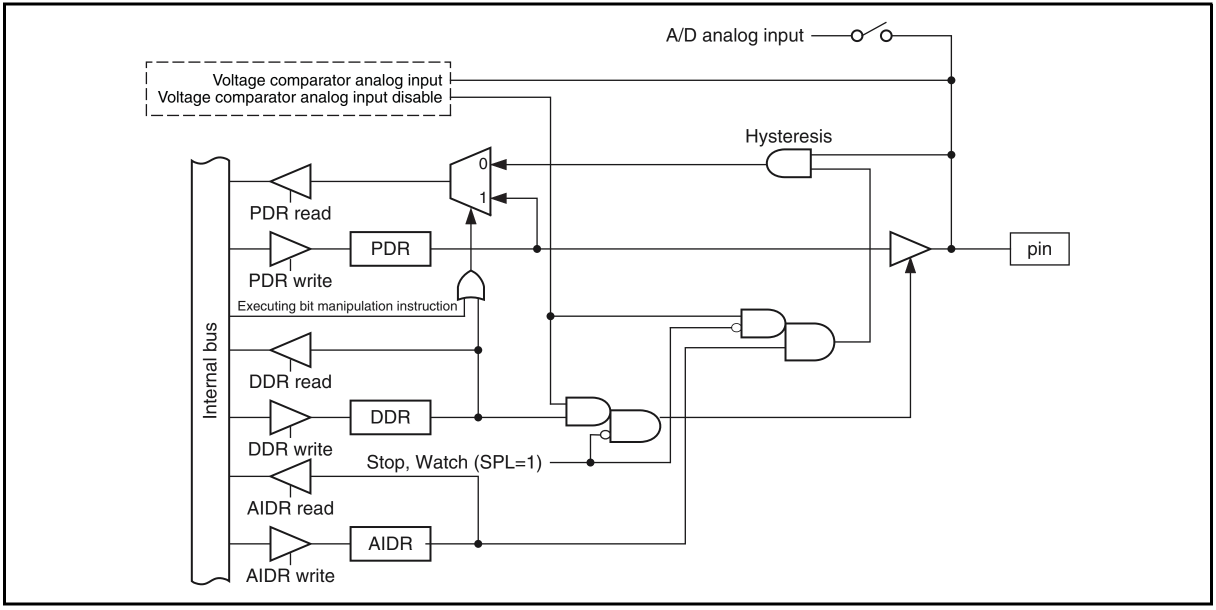
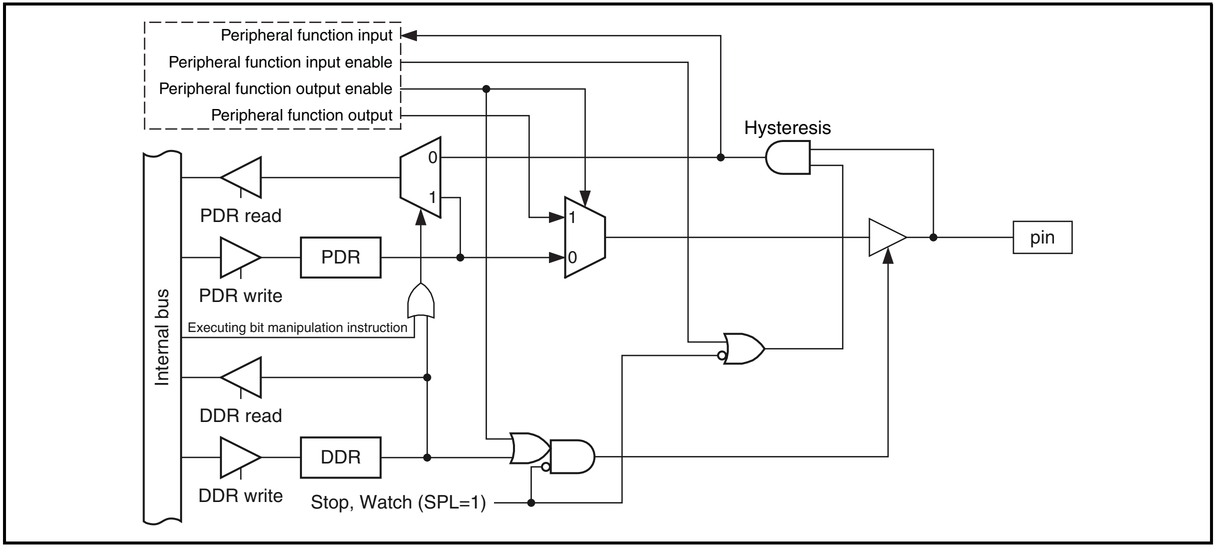


Figure 9.5-3 Block Diagram of P70 and P76


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
0
1
1
0
pin
PDR read
PDR
PDR write
Internal bus
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)

9.5.1 Port 7 Registers

This section describes the registers of port 7.

■ Port 7 Register Functions

Table 9.5-2 lists the port 7 register functions.

Table 9.5-2 Port 7 Register Functions

Register abbr.	Data	Read	Read by read-modify-write instruction	Write
PDR7	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.
DDR7	0	Port input enabled		
	1	Port output enabled		
AIDRH	0	Analog input enabled		
	1	Port output enabled		

Table 9.5-3 shows the correspondence between port 7 pins and each register bit.

Table 9.5-3 Correspondence Between Registers and Pins for Port 7

	Correspondence between related register bits and pins							
Pin name	-	P76	P75	P74	P73	P72	P71	P70
PDR7	-	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR7								
AIDRH	-	-	bit3	bit2	-	bit1	bit0	-

9.5.2 Operations of Port 7

This section describes the operations of port 7.

■ Operations of Port 7

● 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 (RMW) type of instruction 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 (RMW) type of instruction 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 (RMW) type of instruction 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 upper (AIDRH) 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.
- 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 AIDRH 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".

Table 9.5-4 shows the pin states of port 7.

Table 9.5-4 Pin State of Port 7

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 in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

9.6 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 9.6-1 lists the port F pins.

Table 9.6-1 Port F Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PF0/X0*1	PF0: General-purpose I/O	X0: Main clock oscillation pin	Hysteresis	CMOS	-	-
PF1/X1*1	PF1: General-purpose I/O	X1: Main clock oscillation pin	Hysteresis	CMOS	-	-
PF2/ $\overline{\text{RST}}$*2*3	PF2: General-purpose I/O	$\overline{\text{RST}}$: External reset pin	Hysteresis	CMOS	❍	-

OD: Open drain, PU: Pull-up

*1: If the main oscillation clock is selected (SYSC1:PFSEL = 0), the port function cannot be used.
*2: If the external reset is selected (SYSC1:RSTEN = 1), the port function cannot be used.
*3: This pin is a dedicated reset pin in MB95F432H/F433H/F434H.

■ Block Diagrams of Port F

Figure 9.6-1 Block Diagram of PF0 and PF1

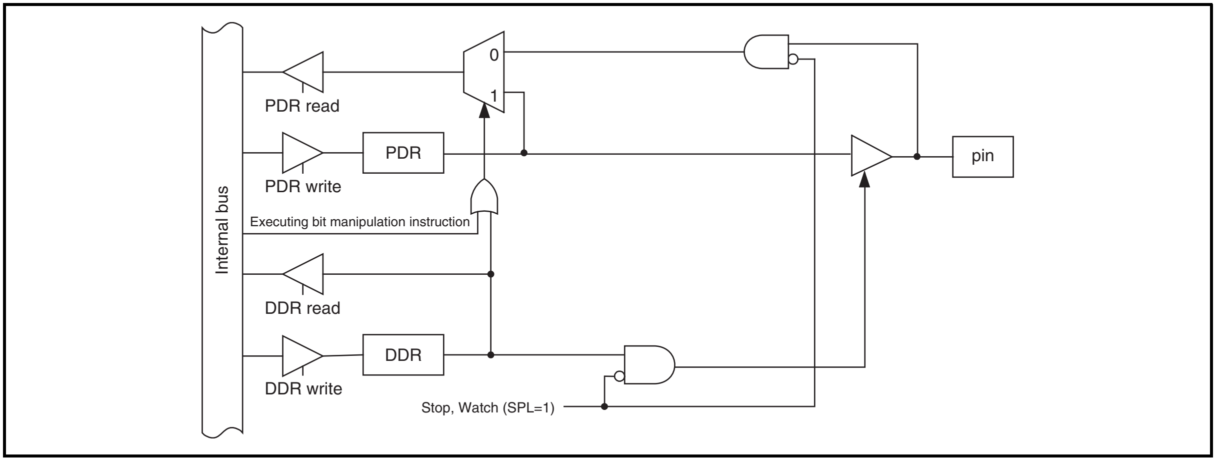
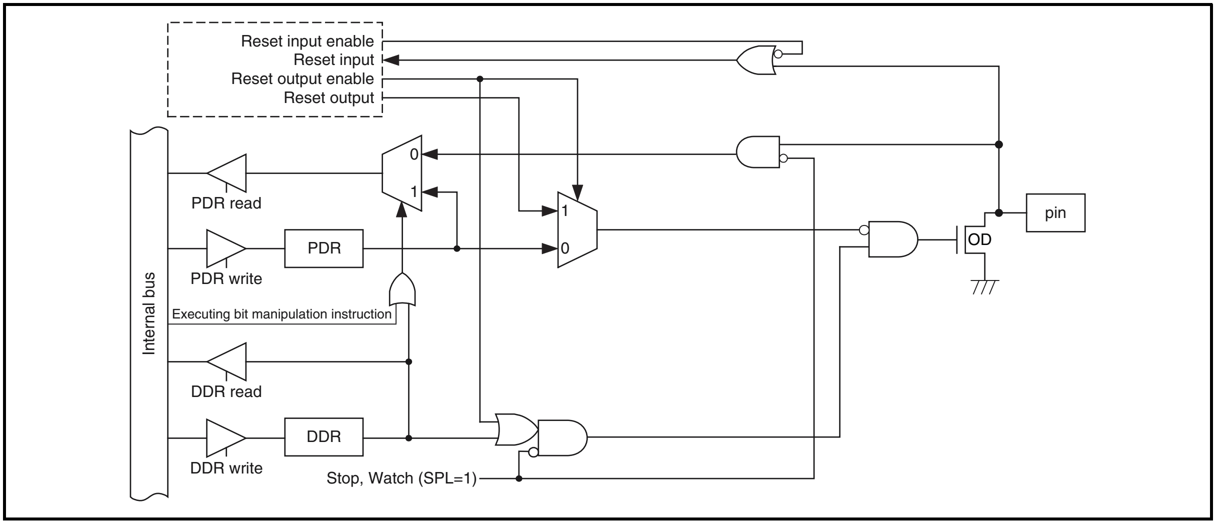


Figure 9.6-2 Block Diagram of PF2



9.6.1 Port F Registers

This section describes the registers of port F.

■ Port F Register Functions

Table 9.6-2 lists the port F register functions.

Table 9.6-2 Port F Register Functions

Register abbr.	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		

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

Table 9.6-3 shows the correspondence between port F pins and each register bit.

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

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

9.6.2 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 (RMW) type of instruction 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 (RMW) type of instruction 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 9.6-4 shows the pin states of port F.

Table 9.6-4 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 in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

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

9.7 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.**

■ 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 9.7-1 lists the port G pins.

Table 9.7-1 Port G Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PG1/TRG/ ADTG/ X0A[*2]/ BZ[*1]/OUT0	PG1: General-purpose I/O	TRG: 16-bit PPG trigger input	Hysteresis	CMOS	-	○
		ADTG: A/D converter trigger input				
		X0A: Subclock I/O oscillation pin				
		BZ: Buzzer output pin				
		OUT0: Output compare ch. 0 output pin				
PG2/PPG[*3]/ X1A[*2]/ OUT1	PG2: General-purpose I/O	PPG: 16-bit PPG output pin	Hysteresis	CMOS	-	○
		X1A: Subclock I/O oscillation pin				
		OUT1: Output compare ch. 1 output pin				

OD: Open drain, PU: Pull-up

*1: The BZ pin has priority over the OUT0 pin.
*2: If the sub-oscillation clock is selected (SYSC1:PGSEL = 0), the port function cannot be used.
*3; The PPG pin has priority over the OUT1 pin.

■ Block Diagrams of Port G

Figure 9.7-1 Block Diagram of PG1

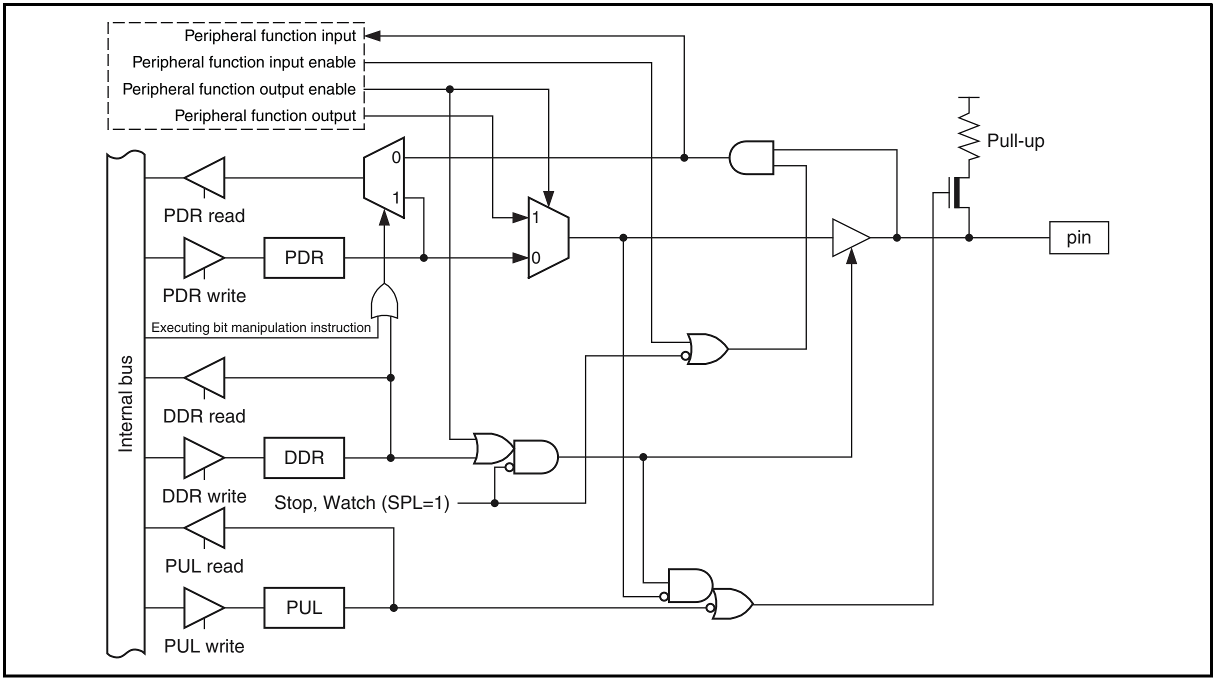
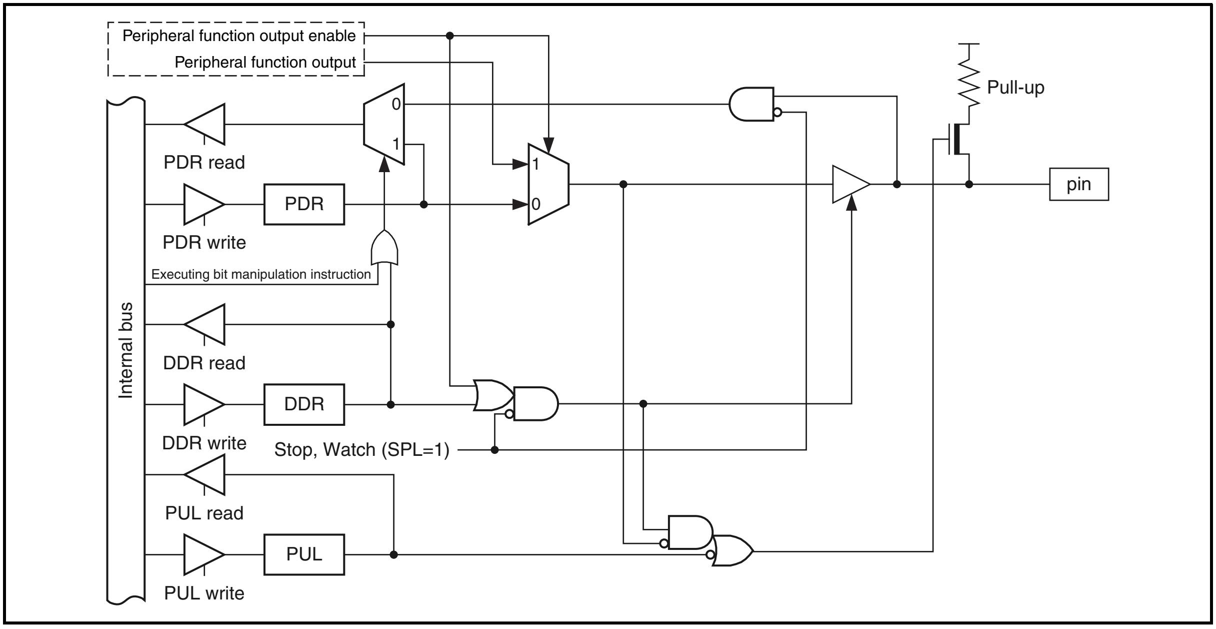

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
0
1
1
0
Pull-up
pin
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
PUL read
PUL
PUL write

Figure 9.7-2 Block Diagram of PG2



9.7.1 Port G Registers

This section describes the registers of port G.

■ Port G Register Functions

Table 9.7-2 lists the port G register functions.

Table 9.7-2 Port G Register Functions

Register abbr.	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 9.7-3 shows the correspondence between port G pins and each register bit.

Table 9.7-3 Correspondence between Registers and Pins for Port G

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

9.7.2 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 (RMW) type of instruction 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 9.7-4 shows the pin states of port G.

Table 9.7-4 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 in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

CHAPTER 10

TIME-BASE TIMER

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

10.1 Overview of Time-base Timer

The time-base timer is a 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 RCS[1:0] bits in the SYCC2 register. The time-base 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 time-base 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 time-base timer.

Table 10.1-1 Interval Times of Time-base 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

MB95430H Series

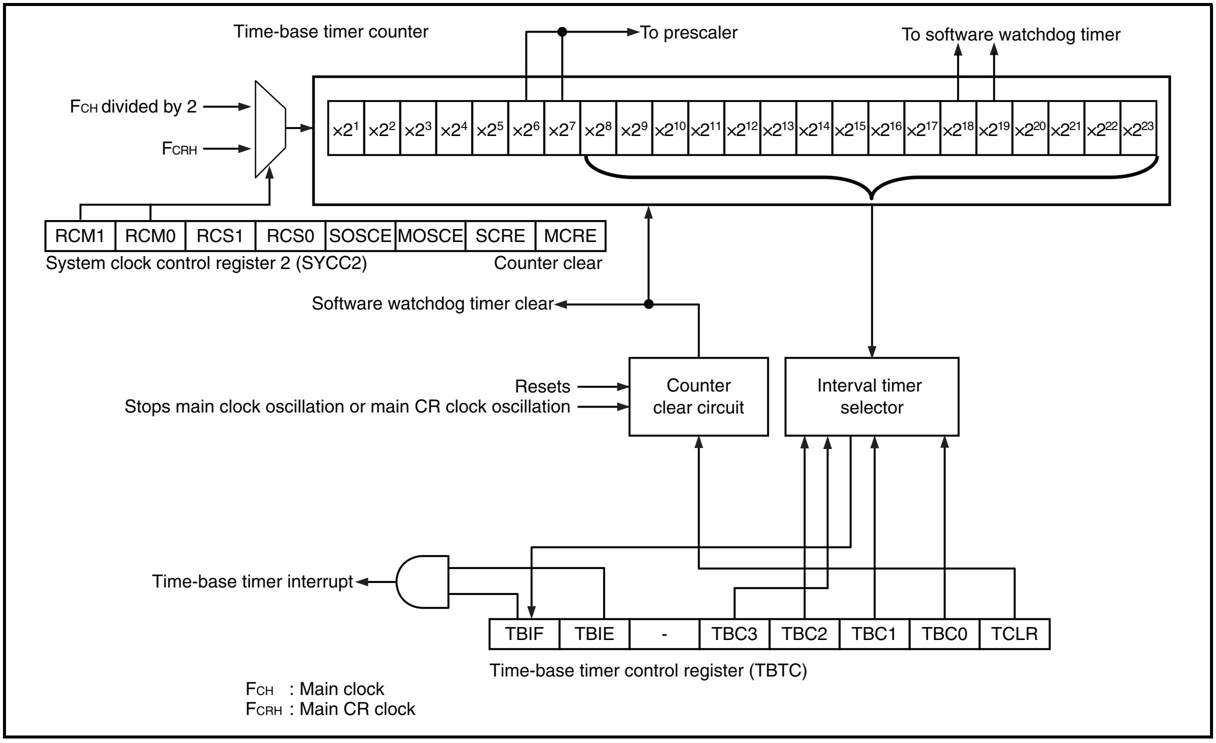
10.2 Configuration of Time-base Timer

The time-base timer consists of the following blocks:

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

■ Block Diagram of Time-base Timer

Figure 10.2-1 Block Diagram of Time-base Timer


Time-base timer counter
To prescaler
To software watchdog timer
FCH divided by 2
FCRH
×2^1 ×2^2 ×2^3 ×2^4 ×2^5 ×2^6 ×2^7 ×2^8 ×2^9 ×2^10 ×2^11 ×2^12 ×2^13 ×2^14 ×2^15 ×2^16 ×2^17 ×2^18 ×2^19 ×2^20 ×2^21 ×2^22 ×2^23
RCM1 RCM0 RCS1 RCS0 SOSCE MOSCE SCRE MCRE
System clock control register 2 (SYCC2)
Counter clear
Software watchdog timer clear
Resets
Stops main clock oscillation or main CR clock oscillation
Counter clear circuit
Interval timer selector
Time-base timer interrupt
TBIF TBIE - TBC3 TBC2 TBC1 TBC0 TCLR
Time-base timer control register (TBTC)
FCH : Main clock
FCRH : Main CR clock

● Time-base 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 time-base timer counter.

● Interval timer selector

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

● Time-base timer control register (TBTC)

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

■ Input Clock

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

■ Output Clock

The time-base timer supplies clocks to the main clock, the software watchdog timer and the prescaler.

10.3 Register of Time-base Timer

Figure 10.3-1 shows the register of the time-base timer.

■ Register of Time-base Timer

Figure 10.3-1 Register of Time-base Timer

Time-base timer control register (TBTC)

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	R0,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.)

R0,W : Write only (Writable. The read value is “0”.)

R0/WX : The read value is “0”. Writing a value to it has no effect on operation.

- : Undefined bit

10.3.1 Time-base Timer Control Register (TBTC)

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

■ Time-base Timer Control Register (TBTC)

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

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
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	Time-base timer initialization bit	
	Read	Write
0	"0" is always read	Has no effect on operation
1	-	Clears all counter bits of the time-base timer to "1".

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

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

TBIF	Time-base timer interrupt request flag bit	
	Read	Write
0	Interval time has not elapsed	Clears the bit
1	Interval time has elapsed	Has no effect on operation

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.)
R0,W : Write only (Writable. The read value is "0".)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

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

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>TBIF:
Time-base timer interrupt request flag bit</td><td>This flag is set to "1" when the interval time selected by the time-base timer elapses.
An interrupt request is output if this bit and the time-base timer interrupt request enable bit (TBIE) are set to "1".
<b>Writing "0"</b>: Clears this bit.
<b>Writing "1"</b>: Has no effect on operation.
When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".</td></tr>
<tr><td>bit6</td><td>TBIE:
Time-base timer interrupt request enable bit</td><td>This bit enables/disables output of interrupt requests to interrupt controller.
<b>Writing "0"</b>: Disables the output of time-base timer interrupt requests.
<b>Writing "1"</b>: Enables the output of time-base timer interrupt requests.
An interrupt request is output if this bit and the time-base timer interrupt request flag bit (TBIF) are set to "1".</td></tr>
<tr><td>bit5</td><td>Undefined bit</td><td>The read value is always "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:
Time-base timer initialization bit</td><td>This bit clears all counter bits of the time-base timer to "1".
<b>Writing "0"</b>: Has no effect on the operation.
<b>Writing "1"</b>: Initializes all counter bits to "1".
When this bit is read, it always returns "0".
Note: When the output of the time-base timer is selected as the count clock for the watchdog timer, using this bit to clear the time-base timer also clears the software watchdog timer.</td></tr>
</table>

10.4 Interrupts of Time-base Timer

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

■ Interrupts When Interval Function Is in Operation

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

Item	Description
Interrupt condition	The interval time set by "TBTC:TBC3-TBC0" has elapsed.
Interrupt flag	TBTC:TBIF
Interrupt enable	TBTC:TBIE

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

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

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

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

10.5 Operations of Time-base Timer and Setting Procedure Example

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

■ Operations of Time-base Timer

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

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

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

■ Clearing Time-base Timer

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

When clearing the counter by using the time-base timer initialization bit (TBTC:TCLR), 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 time-base timer is selected as the count clock for the watchdog timer, clearing the time-base timer also clears the watchdog timer.

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

■ Operation Examples of Time-base Timer

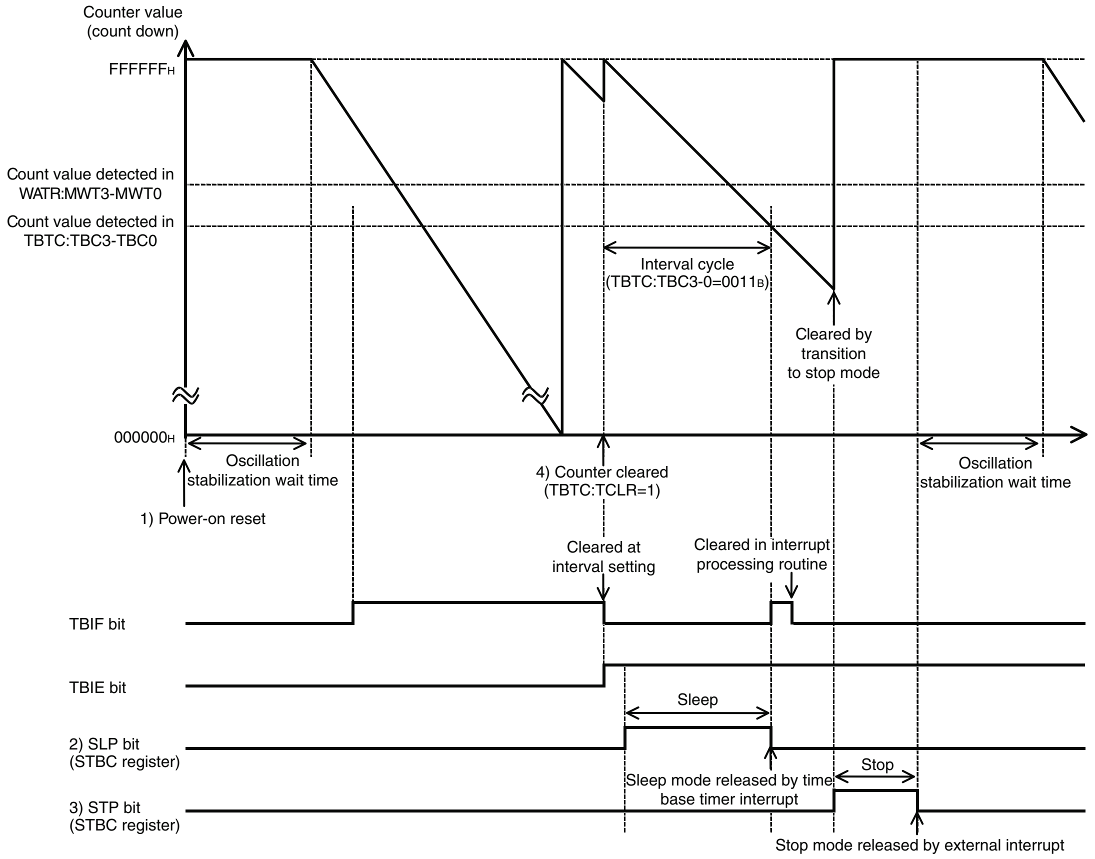
Figure 10.5-2 shows examples of operations under the following conditions:

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 time-base 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.

Figure 10.5-2 Operations of Time-base Timer



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

- TBTC:TBC3-TBC0 : Interval time select bits in time-base timer control register
- TBTC:TCLR : Time-base timer initialization bit in time-base timer control register
- TBTC:TBIF : Time-base timer interrupt request flag bit in time-base timer control register
- TBTC:TBIE : Time-base timer interrupt request enable bit in time-base timer control register
- STBC:SLP : Sleep bit in standby control register
- STBC:STP : Stop bit in standby control register
- WATR:MWT3-MWT0 : 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 time-base 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 Time-base Timer

This section provides notes on using the time-base timer.

■ Notes on Using Time-base Timer

● When setting the timer by program

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

● Clearing Time-base Timer

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

● Peripheral functions receiving clock from time-base timer

In the mode where the source oscillation of the main clock is stopped, the counter is cleared and the time-base timer stops operating. In addition, if the counter of the time-base timer is cleared with the output of the time-base 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 time-base timer is cleared, the clock that is output from the time-base 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.

● Count clock for the software/hardware watchdog timer

- For the software watchdog timer, the output of the time-base timer or of the watch prescaler or of the sub-CR timer can be selected as the count clock.
- For the hardware watchdog timer, only the output of the sub-CR timer can be used as the count clock.

● Activation of the software/hardware watchdog timer

- The software/hardware 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 30 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 bits CS[1:0], CSP	Interval time	
		Minimum time	Maximum time
Time-base 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]=00_B, CSP=1_B(read only)

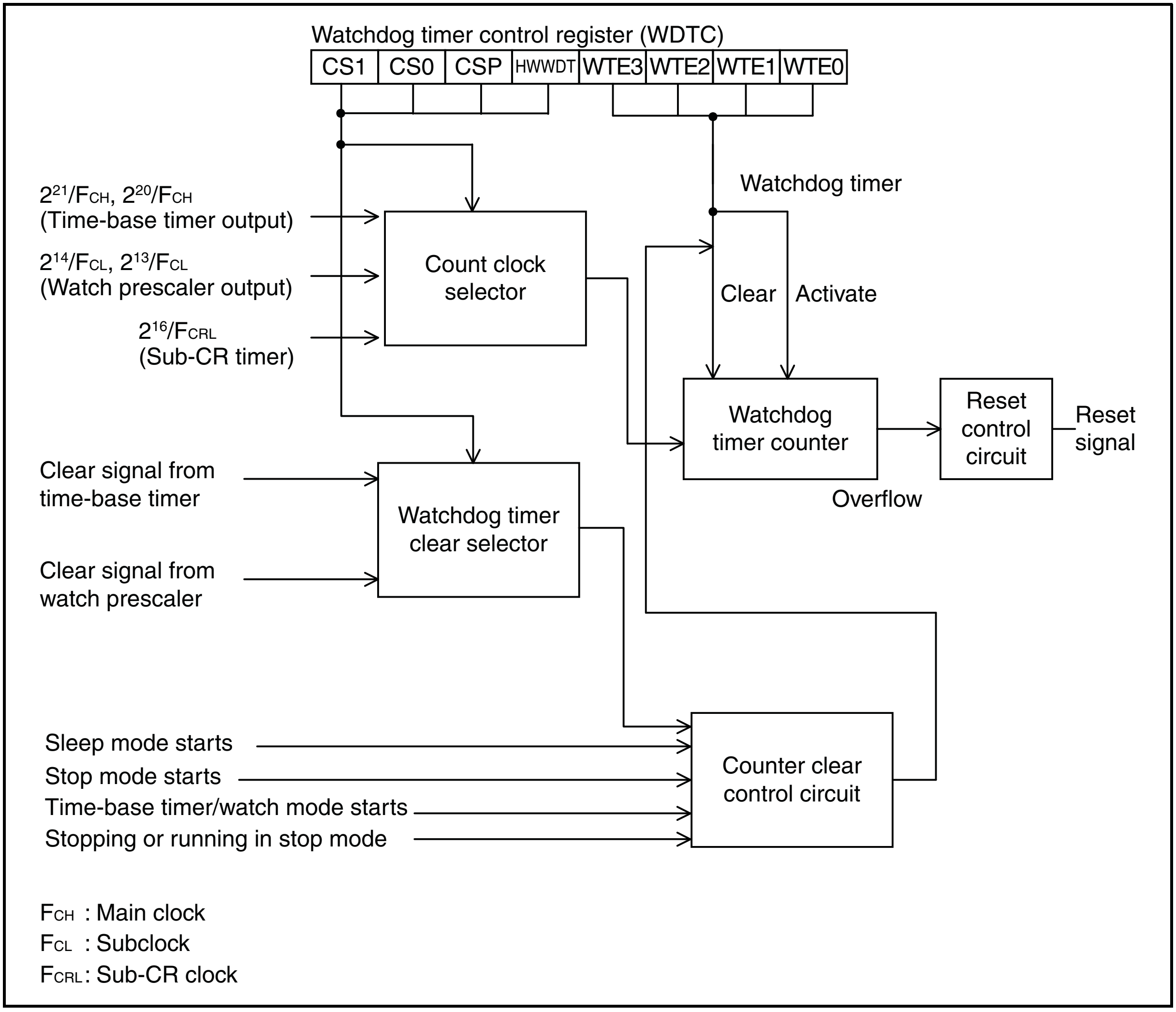
11.2 Configuration of Watchdog Timer

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 time-base 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 time-base 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

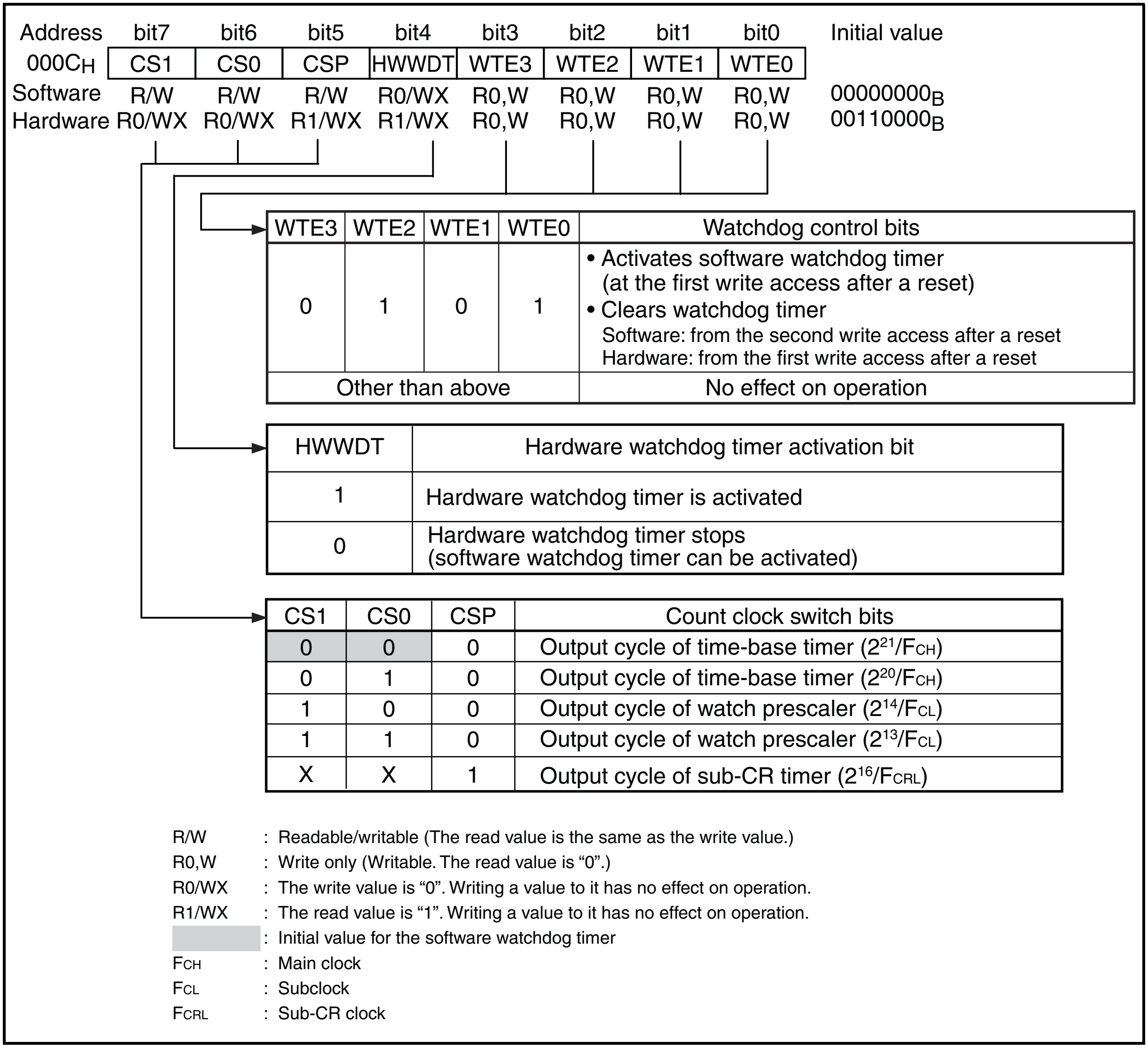
R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
R1/WX : The read value is "1". Writing a value to it has no effect on operation.
R0,W : Write only (Writable. "0" is read.)

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 time-base timer ($2^{21}/F_{CH}$)
0	1	0	Output cycle of time-base 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. The read value is "0".)
R0/WX	:	The write value is "0". Writing a value to it has no effect on operation.
R1/WX	:	The read value is "1". Writing a value to it has no effect on operation.
▨	:	Initial value for the 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 time-base timer ($2^{21}/F_{CH}$)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Output cycle of time-base 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 time-base timer in stopped in subclock mode, always select the output of the watch prescaler in subclock mode.</td></tr>
<tr><td>bit5</td><td>CSP:
Count clock select sub-CR selector bit</td></tr>
<tr><td>bit4</td><td>HWWDT:
Hardware watchdog activation bit</td><td>The bit is a read-only bit, used to confirm the start/stop of the hardware watchdog timer.
<b>Reading "1"</b>: The hardware watchdog timer has been activated.
<b>Reading "0"</b>: 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.
<b>Writing "0101$_B$"</b>:Activates the watchdog timer (at the first write access after a reset) or clears it (from the second write access after a reset).
<b>Writing other than "0101$_B$"</b>: Has no effect on operation.
• When these bits are read, they always return "0000$_B$".</td></tr>
</table>

Note:

Using the read-modify-write (RMW) type of instruction to access the WDTC register is prohibited.

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

- To activate the hardware watchdog timer, write any value except "$A596_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the Flash memory. After a reset, the data in $FFBE_H$ and $FFBF_H$ on the Flash memory are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$ /$0FEC_H$). Writing "$A597_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the Flash memory enables the hardware watchdog timer except in standby modes; writing any value other than "$A596_H$" and "$A597_H$" enables the hardware watchdog timer in all modes. See "CHAPTER 30 NON-VOLATILE REGISTER FUNCTION (NVR)" for details of the watchdog timer selection ID registers.
- 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 (time-base 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/time-base 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.

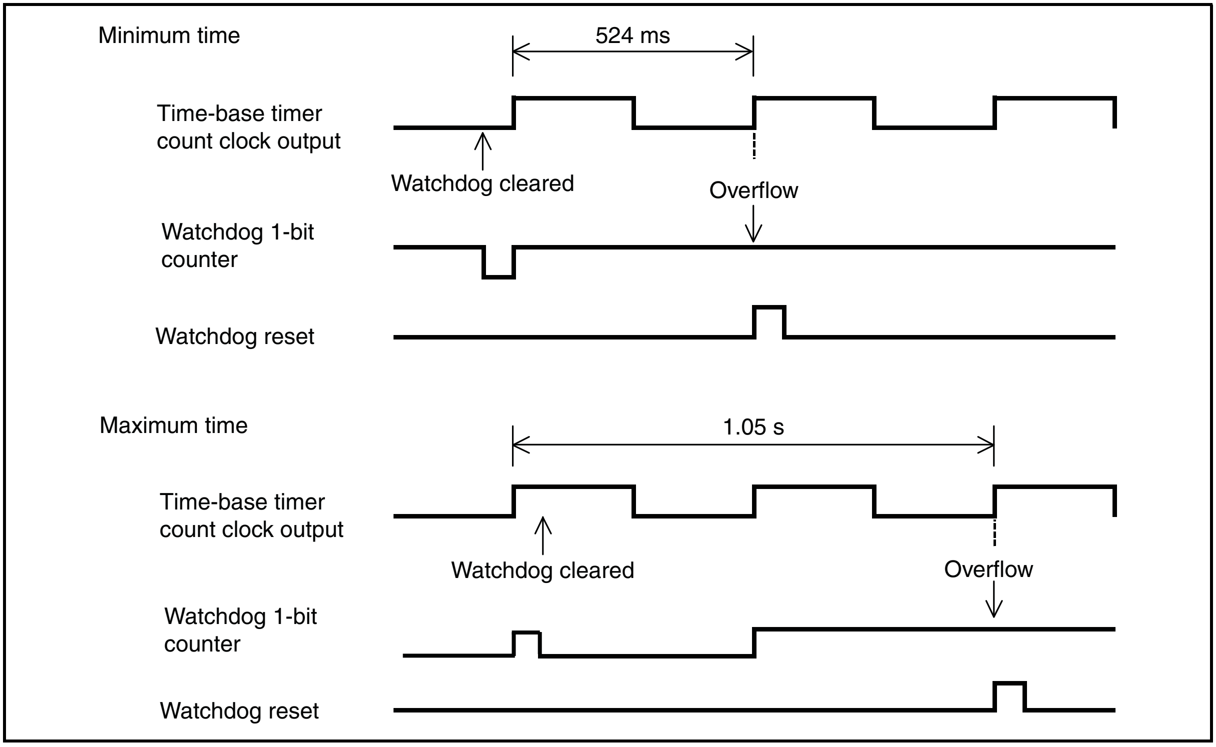
Note:

The watchdog timer is also cleared when the timer selected as the count clock (time-base 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 the watchdog timer and the interval time when the time-base 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 any value except "$A596_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the Flash memory. After a reset, the data in $FFBE_H$ and $FFBF_H$ on the Flash memory are copied to the watchdog timer selection ID registers WDTH/WDTL ($0FEB_H$ /$0FEC_H$). Writing "$A597_H$" to the addresses $FFBE_H$ and $FFBF_H$ on the Flash memory enables the hardware watchdog timer except in standby modes; writing any value other than "$A596_H$" and "$A597_H$" enables the hardware watchdog timer in all modes. See "CHAPTER 30 NON-VOLATILE REGISTER FUNCTION (NVR)" for details of the watchdog timer selection ID registers.
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 time-base 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 (time-base 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, time-base 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, time-base 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 MB95430H Series for the accuracy of the sub-CR clock frequency.

MB95430H Series

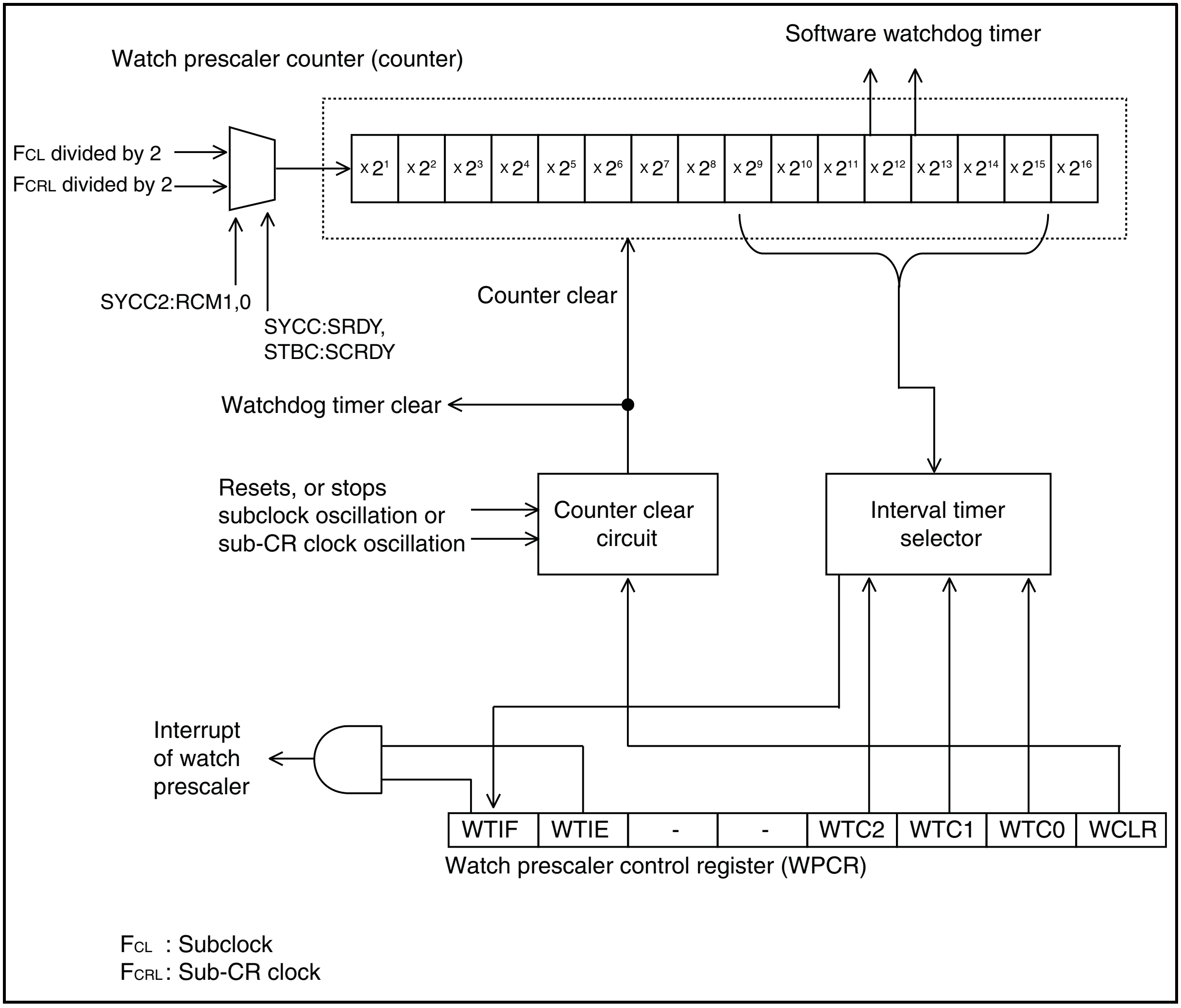
12.2 Configuration of Watch Prescaler

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.

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	

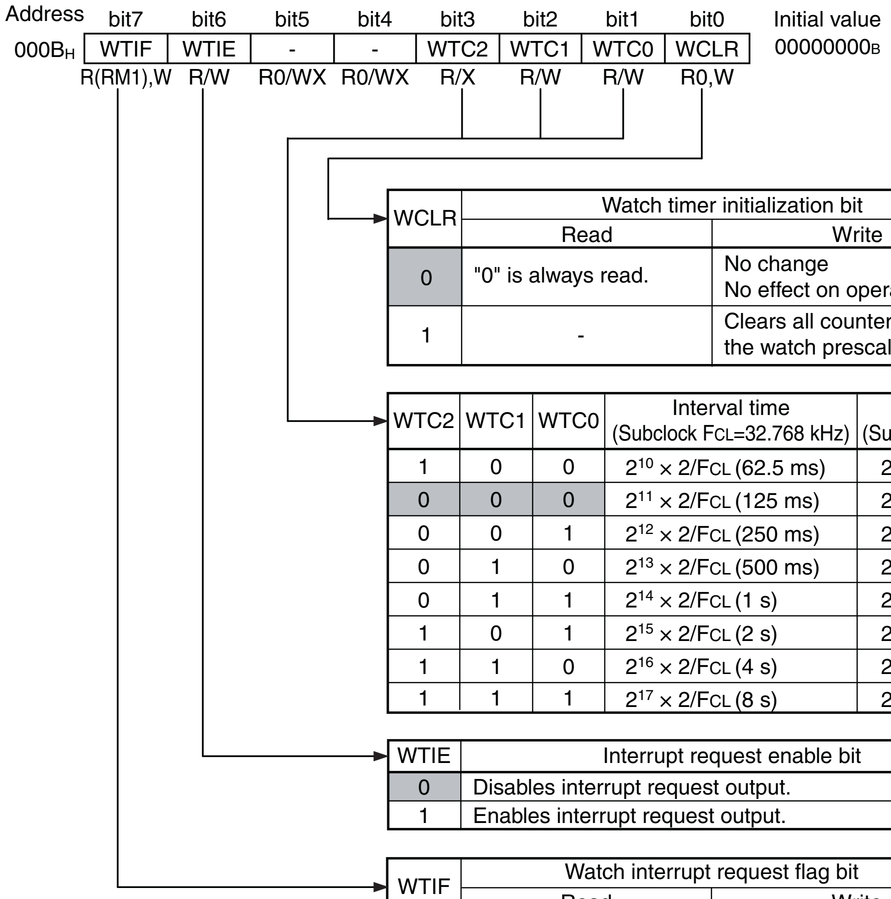
R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R0,W : Write only (Writable. The read value is "0".)
R0/WX : The read value is "0".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

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
000BH	WTIF	WTIE	-	-	WTC2	WTC1	WTC0	WCLR	00000000B
	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 all counter bits of the watch prescaler to "1".

WTC2	WTC1	WTC0	Interval time (Subclock FCL=32.768 kHz)	Interval time (Sub-CR clock FCRL=100 kHz)
1	0	0	2^{10} × 2/FCL (62.5 ms)	2^{10} × 2/FCRL (20.48 ms)
0	0	0	2^{11} × 2/FCL (125 ms)	2^{11} × 2/FCRL (40.96 ms)
0	0	1	2^{12} × 2/FCL (250 ms)	2^{12} × 2/FCRL (81.92 ms)
0	1	0	2^{13} × 2/FCL (500 ms)	2^{13} × 2/FCRL (163.84 ms)
0	1	1	2^{14} × 2/FCL (1 s)	2^{14} × 2/FCRL (327.68 ms)
1	0	1	2^{15} × 2/FCL (2 s)	2^{15} × 2/FCRL (655.36 ms)
1	1	0	2^{16} × 2/FCL (4 s)	2^{16} × 2/FCRL (1.311 s)
1	1	1	2^{17} × 2/FCL (8 s)	2^{17} × 2/FCRL (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/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.)
R0,W	:	Write only (Writable. The read value is "0".)
R0/WX	:	The read value is "0". Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	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".
<b>Writing "0"</b>: Clears this bit to "0".
<b>Writing "1"</b>: Has no effect on operation.
• When read by the read-modify-write (RMW) type of instruction, this bit 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.
<b>Writing "0"</b>: Disables the interrupt request output of the watch prescaler.
<b>Writing "1"</b>: 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>The read value is always "0". Writing a value to it 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 all counter bits of the watch prescaler to "1".
<b>Writing "0"</b>: Has no effect on operation.
<b>Writing "1"</b>: 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 subclock mode 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$_H$

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

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 subclock mode 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 is necessary. The watch prescaler is cleared in the following situations:

- 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.

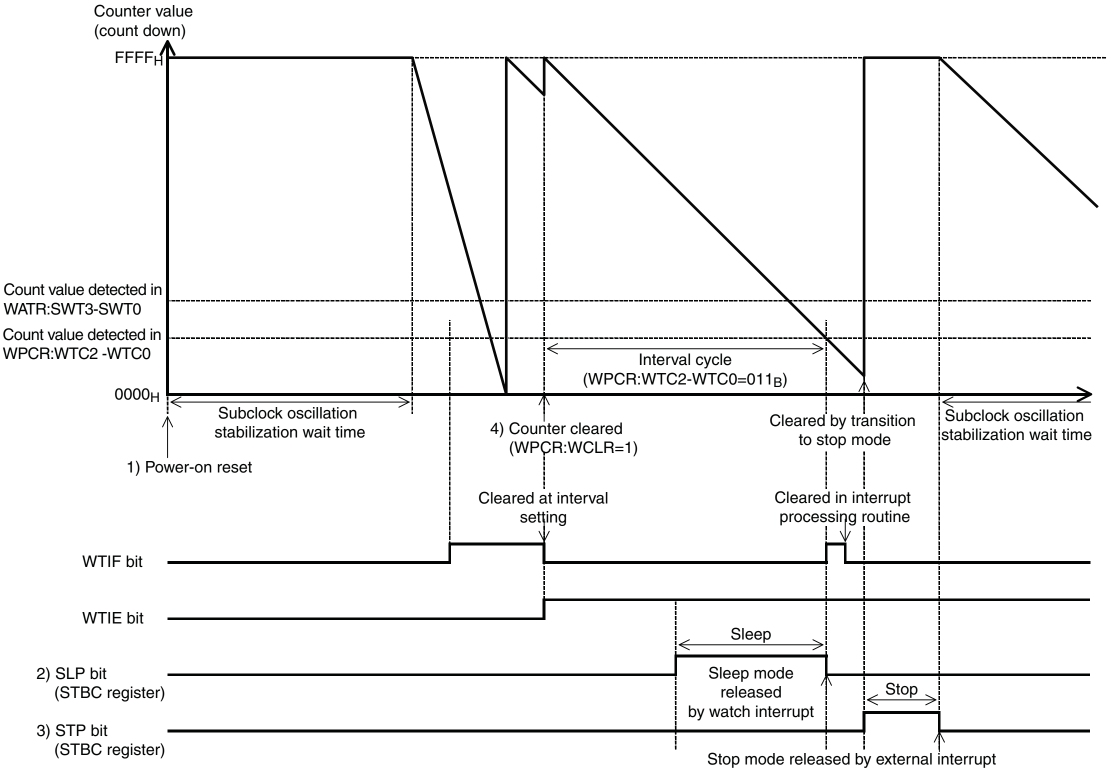
■ Operation Example of Watch Prescaler

Figure 12.5-1 shows an operation example 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 Watch Prescaler Operation Example


Counter value
(count down)
FFFFH
Count value detected in WATR:SWT3-SWT0
Count value detected in WPCR:WTC2 -WTC0
0000H
Interval cycle
(WPCR:WTC2-WTC0=011B)
Subclock oscillation stabilization wait time
1) Power-on reset
4) Counter cleared
(WPCR:WCLR=1)
Cleared by transition to stop mode
Subclock oscillation stabilization wait time
Cleared at interval setting
Cleared in interrupt processing routine
WTIF bit
WTIE bit
Sleep
2) SLP bit
(STBC register)
Sleep mode released by watch interrupt
Stop
3) STP bit
(STBC register)
Stop mode released by external interrupt

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

• WPCR:WTC2-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-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, and can also be made to wait for the end of subclock/sub-CR clock oscillation stabilization wait time before starting to perform counting. To make the prescaler wait for the end of subclock oscillation stabilization wait time or that of sub-CR clock oscillation stabilization wait time, set SYCC2:SOSCE or SYCC2:SCRE to "1" respectively. In addition, in the same mode, the watch prescaler can 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 Example of Setting Watch Prescaler

This section describes the example of setting watch prescaler.

■ Example of Setting Methods

● 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.

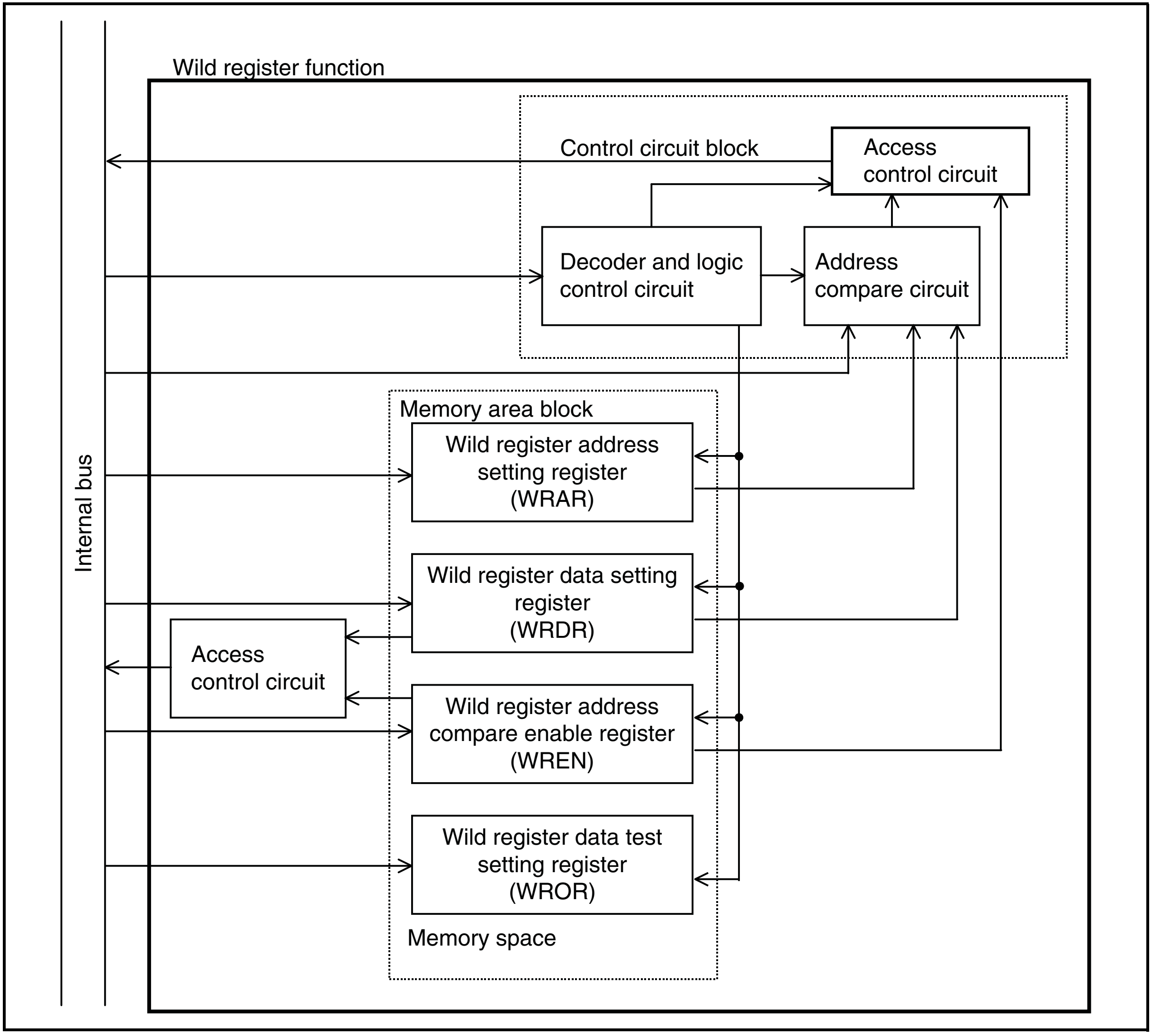
13.2 Configuration of Wild Register Function

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


Internal bus
Wild register function
Control circuit block
Access control circuit
Decoder and logic control circuit
Address compare circuit
Memory area block
Wild register address setting register (WRAR)
Wild register data setting register (WRDR)
Access control circuit
Wild register address compare enable register (WREN)
Wild register data test setting register (WROR)
Memory space

● 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
WRDR0	$0F82_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
WRDR1	$0F85_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRDR2	$0F88_H$									

Wild register address setting registers (WRAR0 to WRAR2)

	Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
WRAR0	$0F80_H$, $0F81_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000_B
WRAR1	$0F83_H$, $0F84_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRAR2	$0F86_H$, $0F87_H$	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
		RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Wild register address compare enable register (WREN)

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	

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	

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

■ 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	

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

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

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it 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	

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

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

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it 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^{2}PROM 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 is "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$_H$".

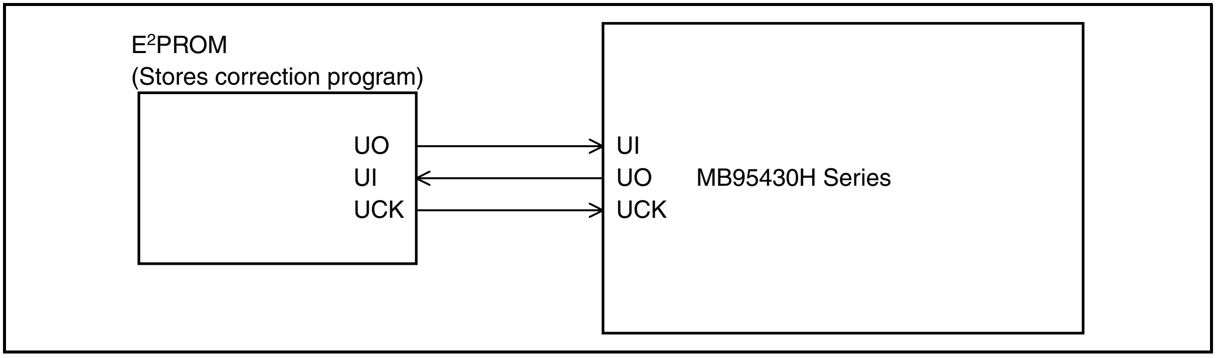
Since the address "0078$_H$" is used as a mirror address for the register bank pointer and the direct bank pointer, this address cannot be patched.

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 ch. 0, 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 ch. 0, 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 ch. 0, 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".

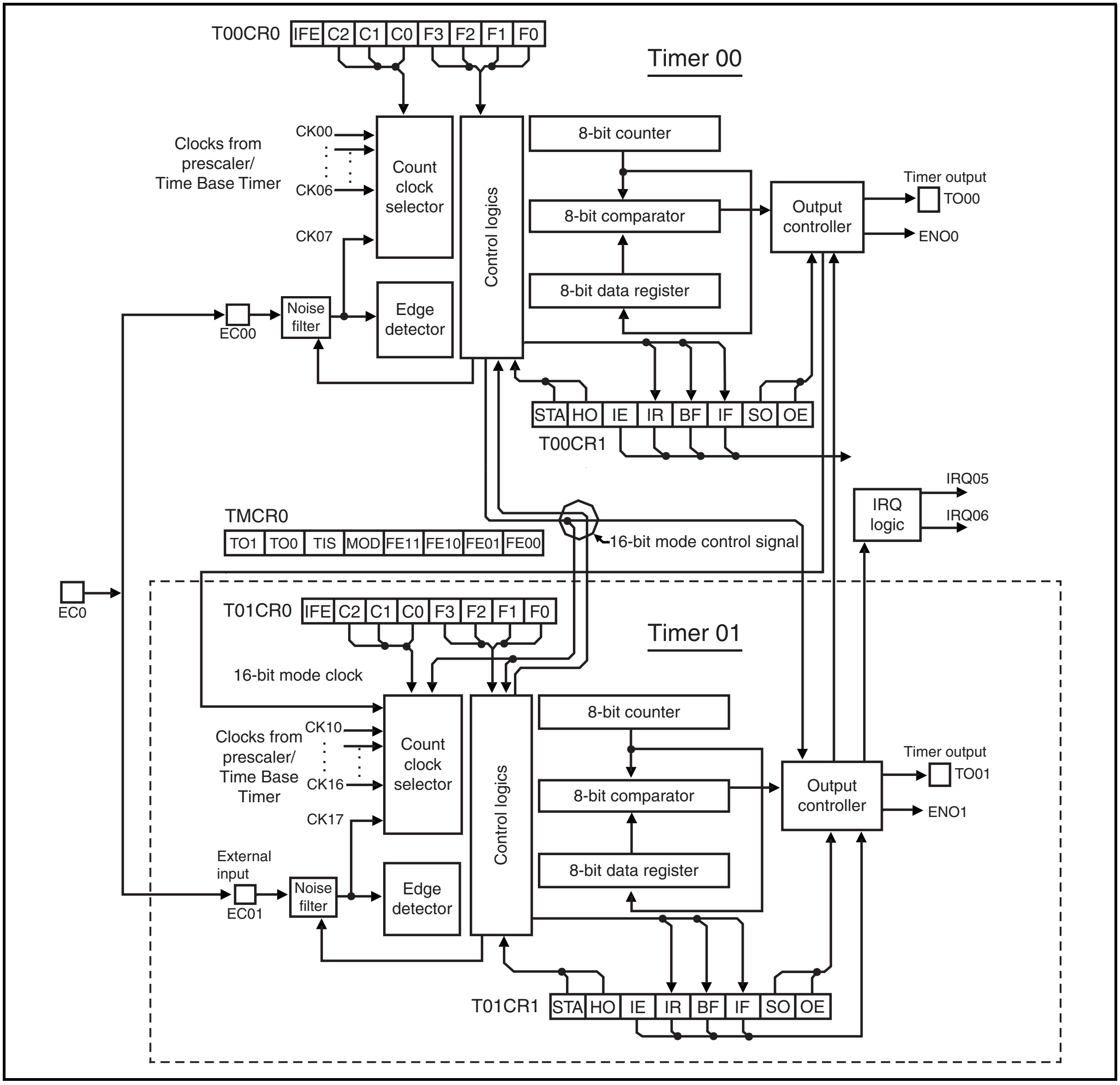
14.2 Configuration of 8/16-bit Composite Timer

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 × 1 channel (T00DR/T01DR)**
- **8/16-bit composite timer 00/01 status control register 0 × 1 channel (T00CR0/T01CR0)**
- **8/16-bit composite timer 00/01 status control register 1 × 1 channel (T00CR1/T01CR1)**
- **8/16-bit composite timer 00/01 timer mode control register (TMCR0)**
- **Output controller × 1 channel**
- **Control logic × 1 channel**
- **Count clock selector × 1 channel**
- **Edge detector × 1 channel**
- **Noise filter × 1 channel**

■ 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 data register and that in the counter. It incorporates a latch that temporarily stores the 8/16-bit composite timer 00 data register value.

● 8/16-bit composite timer 00/01 data register ch. 0 (T00DR/T01DR)

This 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 time-base 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.

■ Input Clock

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

14.3 Channel of 8/16-bit Composite Timer

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

■ Channel of 8/16-bit Composite Timer

There are two 8-bit counters in the channel. 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 Channel 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 14.3-2 8/16-bit Composite Timer Channel and Corresponding Registers

Channel	Register abbreviation	Corresponding register (Name in this manual)
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 of 8/16-bit Composite Timer

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

■ Pins of 8/16-bit Composite Timer

The external pins of the 8/16-bit composite timer are TO00, TO01 and EC0.

● 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 (DDR0: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 function, PWM timer function (fixed-cycle mode), or PWC timer function, the pin becomes an output pin automatically regardless of the port direction register (DDR0: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 of 8/16-bit Composite Timer

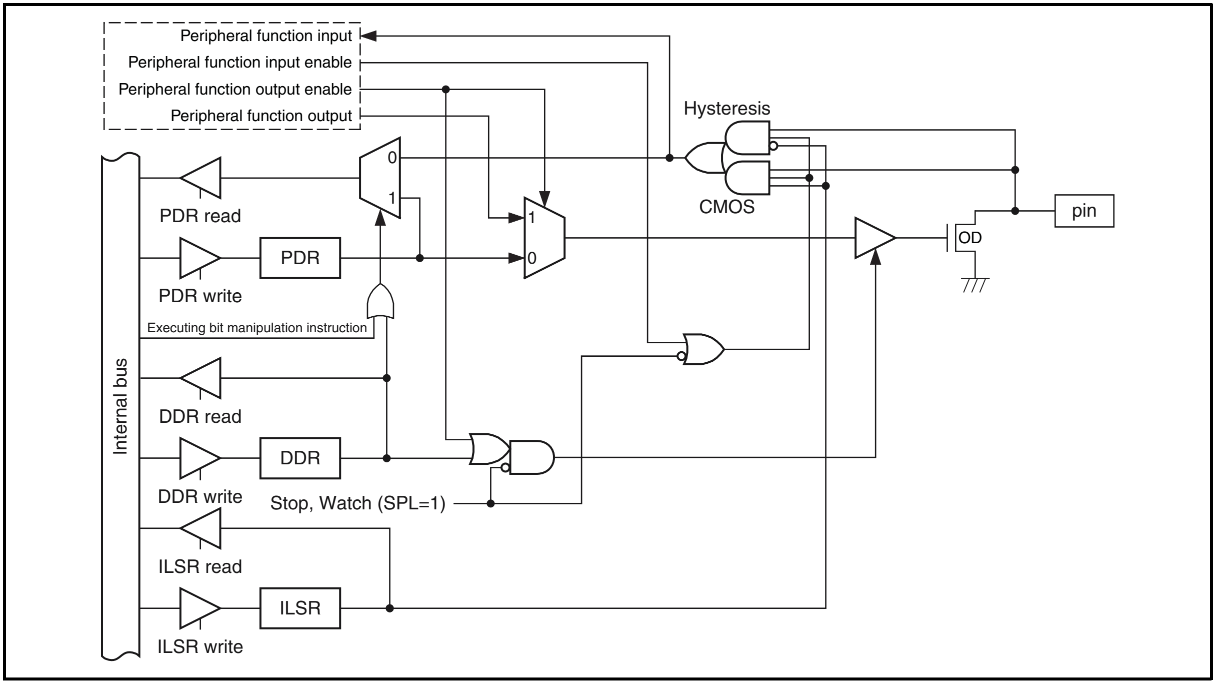

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
pin
OD
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

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

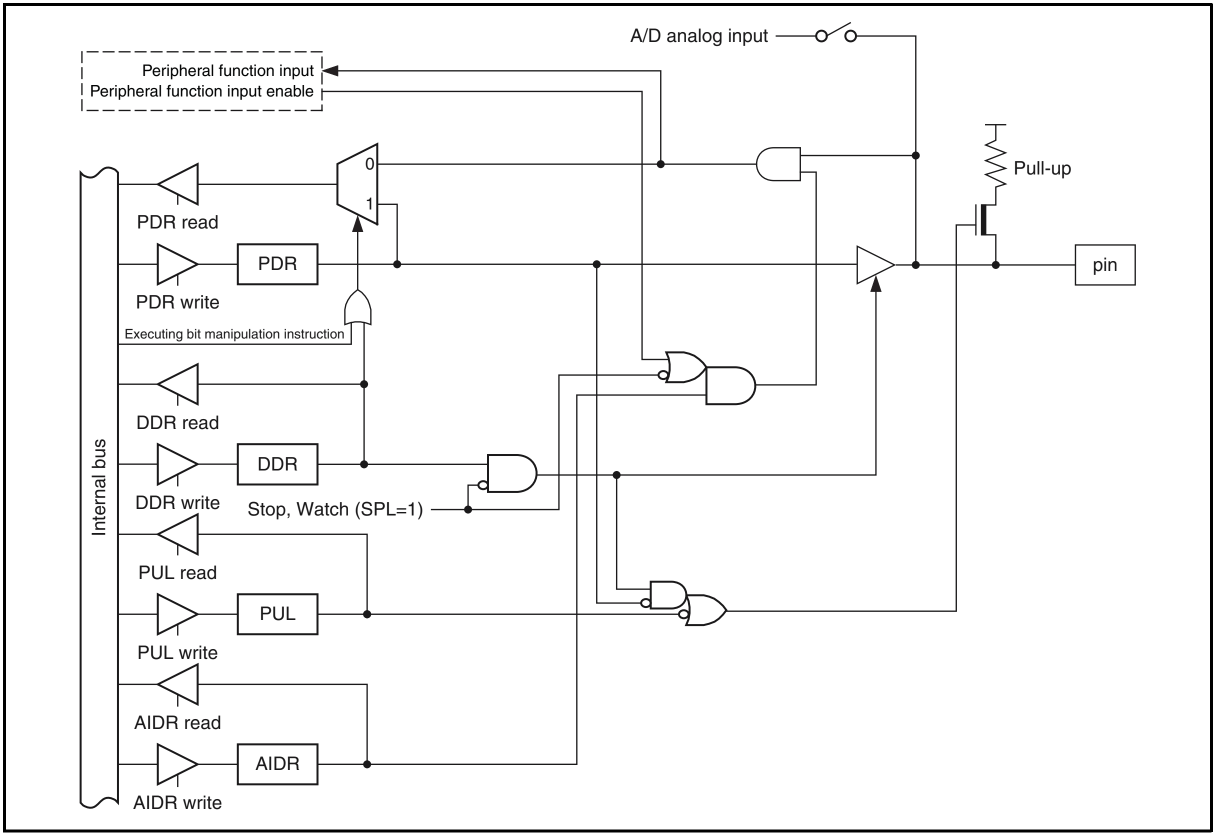


Figure 14.4-2 Block Diagram of Pin EC0 (P07/INT07/AN07/EC0) of 8/16-bit Composite Timer

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

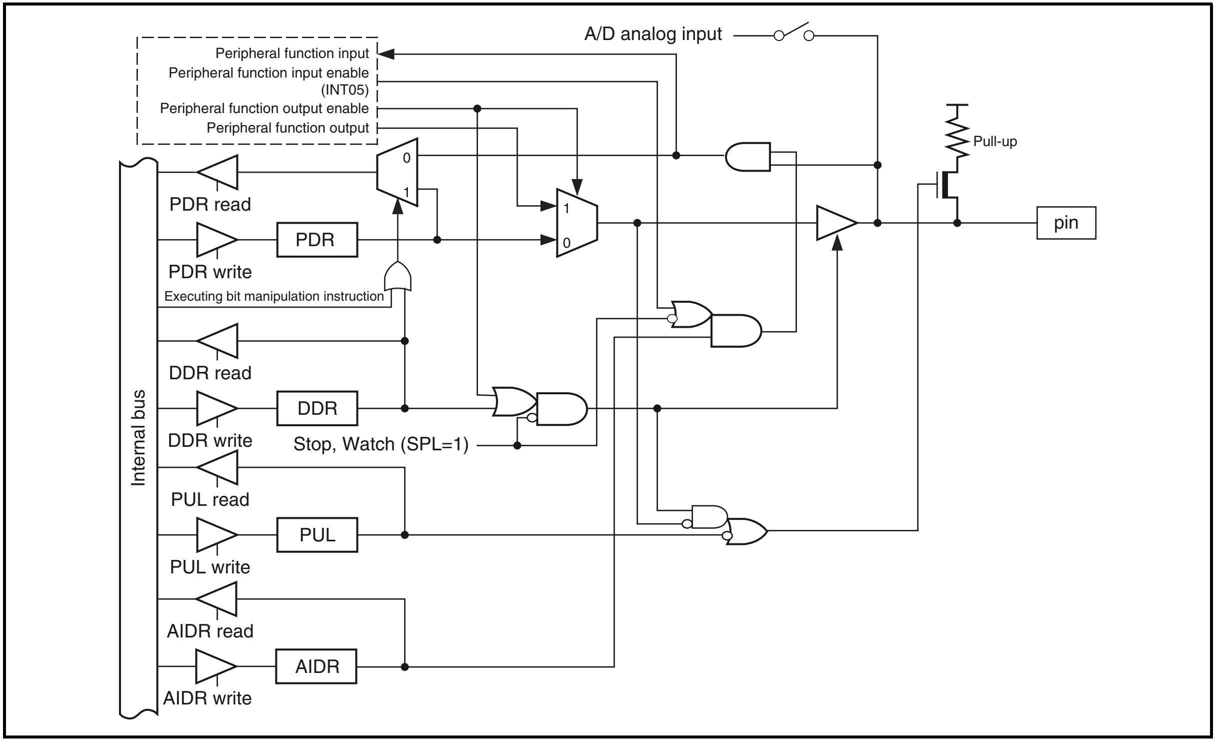
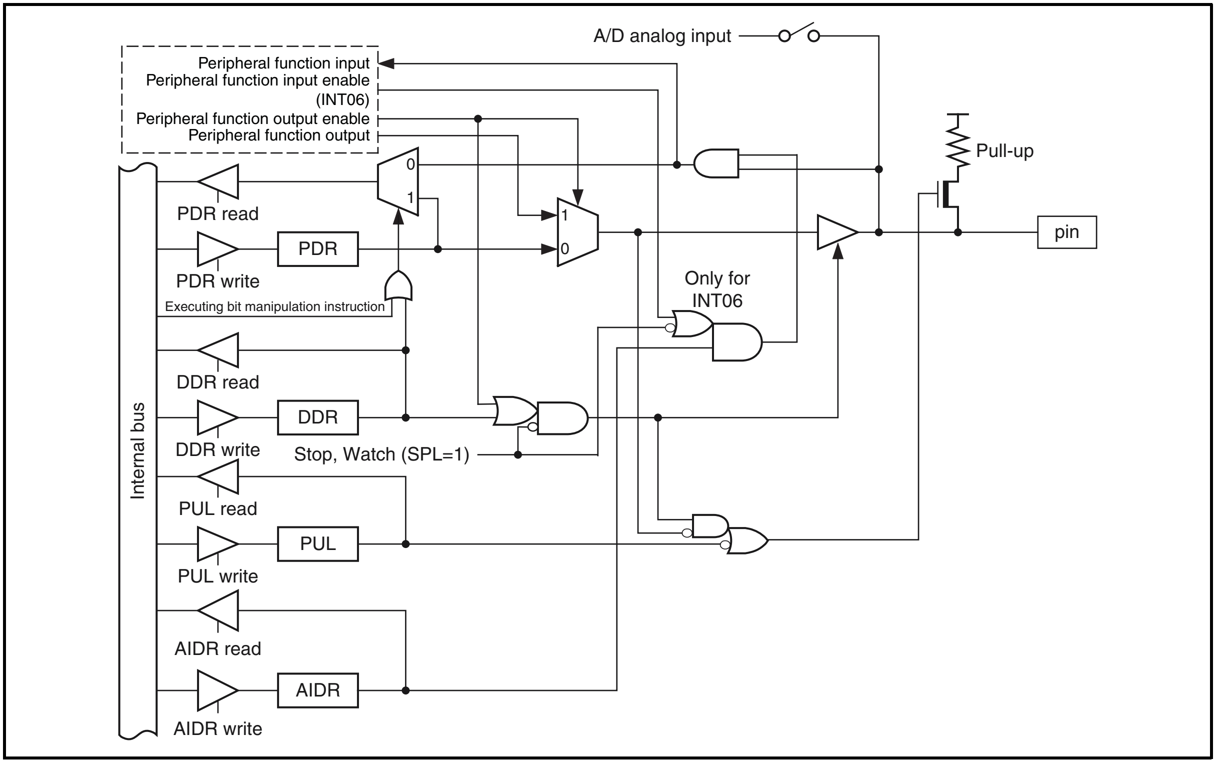

A/D analog input
Peripheral function input
Peripheral function input enable
(INT05)
Peripheral function output enable
Peripheral function output
Pull-up
0
1
1
0
pin
PDR read
PDR
PDR write
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
Internal bus
PUL read
PUL
PUL write
AIDR read
AIDR
AIDR write

Figure 14.4-4 Block Diagram of Pin TO01 (P06/INT06/AN06/TO01) of 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 0

Figure 14.5-1 Registers of 8/16-bit Composite Timer 0

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

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01CR0	$0F92_H$	IFE	C2	C1	C0	F3	F2	F1	F0	00000000_B
T00CR0	$0F93_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01CR1	0036_H	STA	HO	IE	IR	BF	IF	SO	OE	00000000_B
T00CR1	0037_H	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

8/16-bit composite timer 00/01 data register ch. 0 (T00DR/T01DR)

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

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

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	$0F96_H$	TO1	TO0	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
T01CR0 T00CR0	$0F92_H$ $0F93_H$	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 operating 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 × MCLK (machine clock)
0	0	1	1/2 × MCLK (machine clock)
0	1	0	1/4 × MCLK (machine clock)
0	1	1	1/8 × MCLK (machine clock)
1	0	0	1/16 × MCLK (machine clock)
1	0	1	1/32 × MCLK (machine clock)
1	1	0	1/128 × F_{CH} or 1/64 × F_{CRH}*
1	1	1	External clock

IFE	IF flag interrupt enable
0	Disables the IF flag interrupt
1	Enables the IF flag interrupt

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

▨ : Initial value

* : The value to be used as the count clock is decided according to the settings of the SYCC2 register.

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.
<b>Writing "0"</b>: disables IF flag interrupts.
<b>Writing "1"</b>: 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".
• When these bits are set to "110_B", the count clock from the time-base timer will be used as the count clock. Depending on the settings of the SYCC2 register, the count clock from the time-base timer can be generated from either main clock or main CR clock. In the case of using the count clock from the time-base timer as the count clock, resetting the time-base timer by writing "1" to the time-base timer initialization bit in the time-base timer control register (TBTC:TCLR) will affect the count time.
<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>$1/128 \times F_{CH}$ or $1/64 \times F_{CRH}$</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)

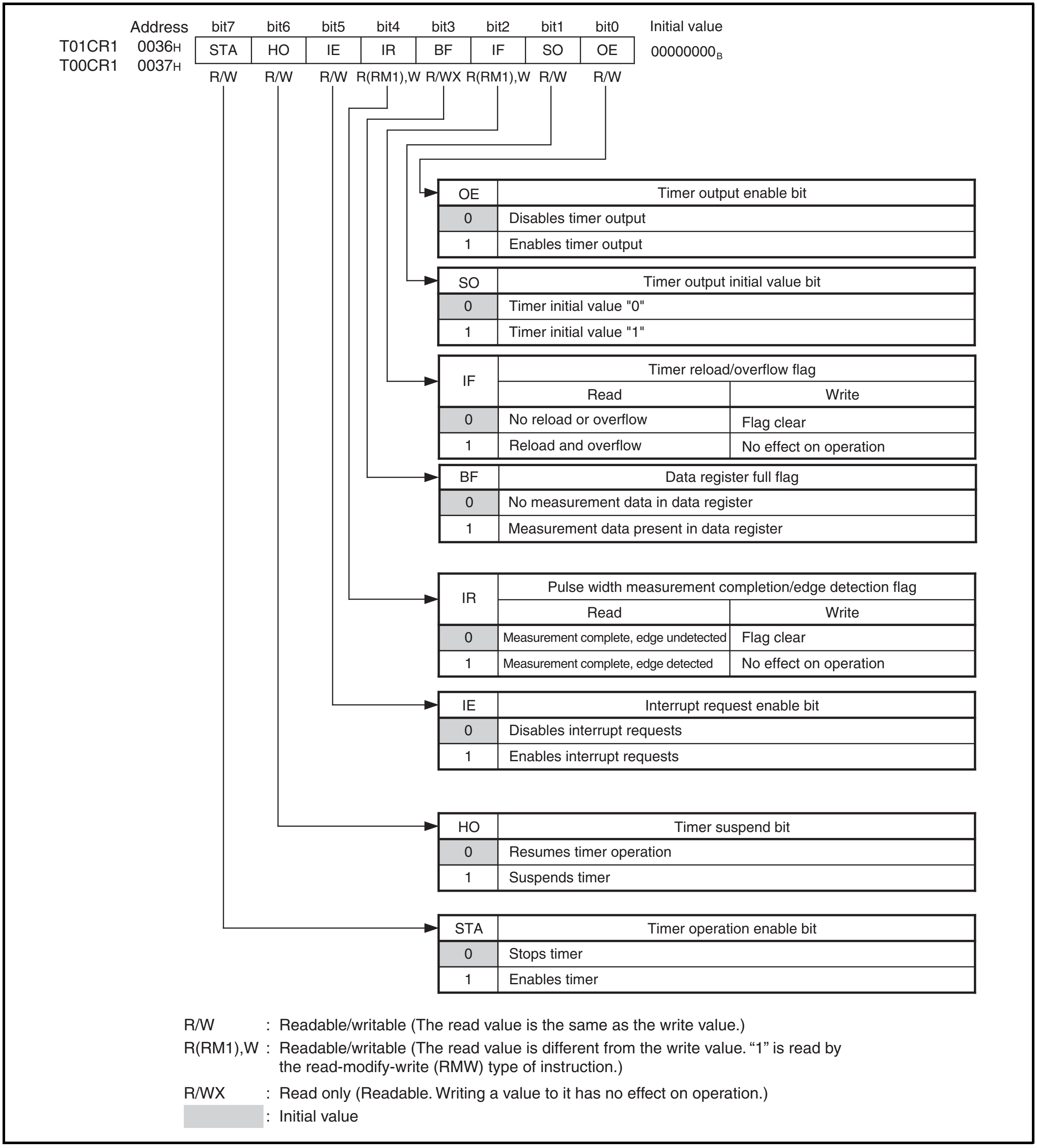
<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit3
to
bit0</td><td>F3, F2, F1, F0:
Timer operating mode select bits</td><td>These bits select the timer operating mode.
• The PWM timer function (variable-cycle mode; F3, F2, F1, F0 = 0100_B) is set by either the T00CR0 (timer 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
T01CR1 T00CR1	0036$_H$ 0037$_H$	STA	HO	IE	IR	BF	IF	SO	OE	00000000$_B$
		R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

OE	Timer output enable bit
0	Disables timer output
1	Enables timer output

SO	Timer output initial value bit
0	Timer initial value "0"
1	Timer initial value "1"

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

BF	Data register full flag
0	No measurement data in data register
1	Measurement data present in data register

IR	Pulse width measurement completion/edge detection flag	
	Read	Write
0	Measurement complete, edge undetected	Flag clear
1	Measurement complete, edge detected	No effect on operation

IE	Interrupt request enable bit
0	Disables interrupt requests
1	Enables interrupt requests

HO	Timer suspend bit
0	Resumes timer operation
1	Suspends timer

STA	Timer operation enable bit
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 (T00CR1/T01CR1) (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. • In 16-bit operation (TMCR0:MOD = 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 set 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:MOD), 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 in either T00CR1 (timer 00) or T01CR1 (timer 01) can be used 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. • In 16-bit operation (TMCR0:MOD = 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. • When read by the read-modify-write (RMW) type of instruction, this bit 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 (T00CR1/T01CR1) (2 / 2)

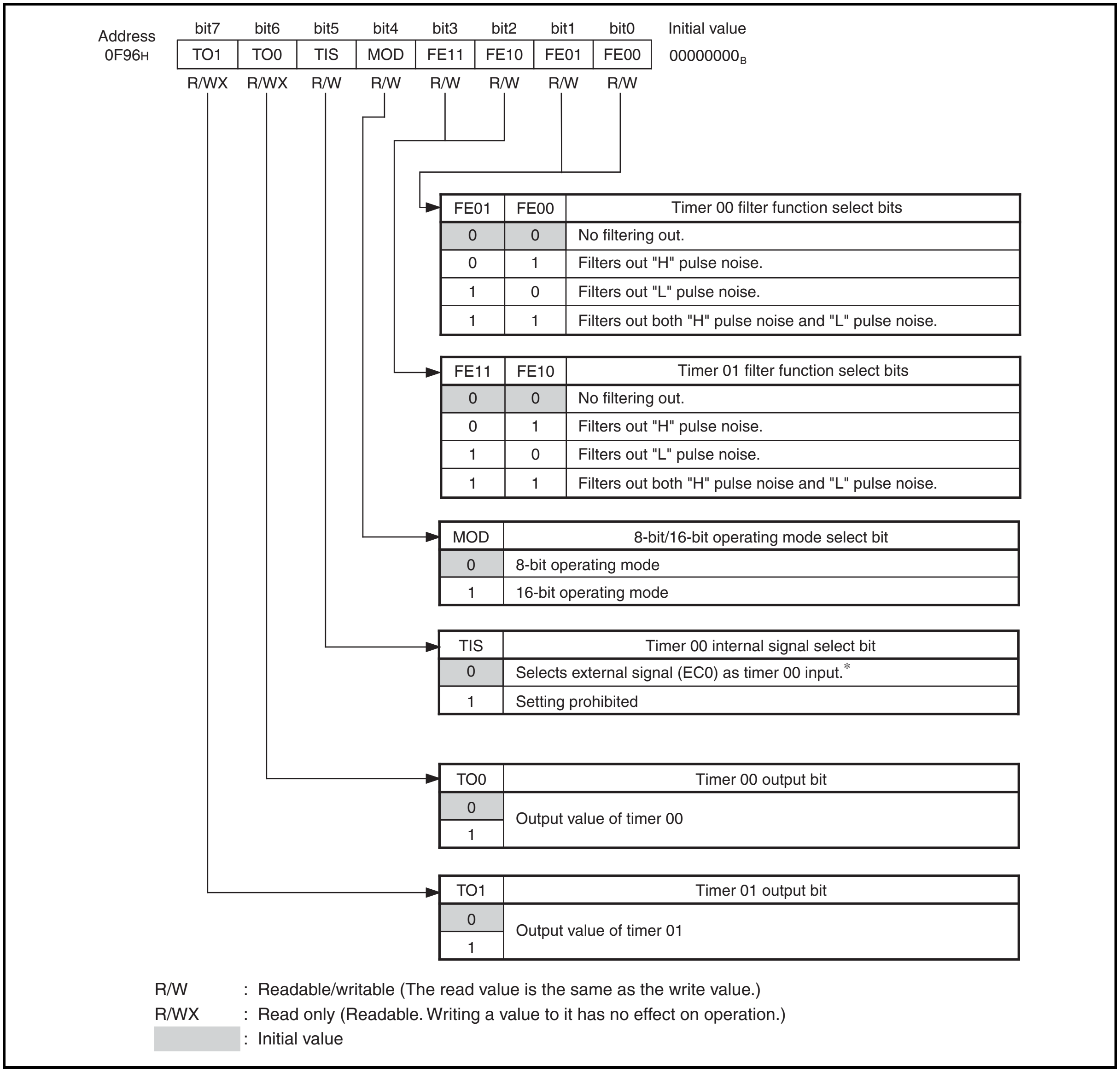
Bit name		Function
bit3	BF: Data register full flag	• With the PWC timer function in use, this bit is set to "1" when a count value is stored in the 8/16-bit composite timer 00/01 data register ch. 0 (T00DR/T01DR) immediately after pulse width measurement is complete. • In 8-bit operation, this bit is set to "0" when the 8/16-bit composite timer 00/01 data register ch. 0 (T00DR/T01DR) is read. • The 8/16-bit composite timer 00/01 data register ch. 0 (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 ch. 0 (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 ch. 0 (T00DR/T01DR), while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register ch. 0. 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 ch. 0 (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)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F96H	TO1	TO0	TIS	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 out.
0	1	Filters out "H" pulse noise.
1	0	Filters out "L" pulse noise.
1	1	Filters out both "H" pulse noise and "L" pulse noise.

FE11	FE10	Timer 01 filter function select bits
0	0	No filtering out.
0	1	Filters out "H" pulse noise.
1	0	Filters out "L" pulse noise.
1	1	Filters out both "H" pulse noise and "L" pulse noise.

MOD	8-bit/16-bit operating mode select bit
0	8-bit operating mode
1	16-bit operating mode

TIS	Timer 00 internal signal select bit
0	Selects external signal (EC0) as timer 00 input.*
1	Setting prohibited

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/WX : Read only (Readable. Writing a value to it has no effect on operation.)
▨ : Initial value

*: The EC0 input can be assigned to P12 or P07 by setting the SYSC1 register. For details, see "CHAPTER 31 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>TIS:
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.
<b>Writing "0"</b>: Selects the external signal (EC0) as the signal input for timer 00.
<b>Writing "1"</b>: Is prohibited because it selects the internal signal (TII0) as the signal input for timer 10; however, the TII0 pin for ch. 1 is internally fixed at "0".
The EC0 input can be assigned to P12 or P07 by setting the SYSC1 register. For details, see "31.2.1 System Configuration Register 1 (SYSC1)" in CHAPTER 31.</td></tr>
<tr><td>bit4</td><td>MOD:
8-bit/16-bit operating mode select bit</td><td>This bit selects 8-bit or 16-bit operating mode.
<b>Writing "0"</b>: Allows timers 00 and 01 to operate as separate 8-bit timers.
<b>Writing "1"</b>: 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 out.</td></tr>
<tr><td>0</td><td>1</td><td>Filters out "H" pulse noise.</td></tr>
<tr><td>1</td><td>0</td><td>Filters out "L" pulse noise.</td></tr>
<tr><td>1</td><td>1</td><td>Filters out both "H" pulse noise and "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 out.</td></tr>
<tr><td>0</td><td>1</td><td>Filters out "H" pulse noise.</td></tr>
<tr><td>1</td><td>0</td><td>Filters out "L" pulse noise.</td></tr>
<tr><td>1</td><td>1</td><td>Filters out both "H" pulse noise and "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 ch. 0 (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
T01DR	$0F94_H$	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000_B
T00DR	$0F95_H$	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	
R,W	: Readable/writable (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, write the upper timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

● PWM timer function (fixed-cycle)

The 8/16-bit composite timer 00/01 data register ch. 0 (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, write the upper timer data to T01DR and lower timer data to T00DR, and write or read T01DR first and then T00DR.

● PWM timer function (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 time 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 counter performs the next PWM cycle of counting.

The current count value can be read from this register. In 16-bit operation, write the upper timer data to T01DR and lower timer data to T00DR, and read T01DR first and then T00DR.

● PWC timer function

The 8/16-bit composite timer 00/01 data register ch. 0 (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 ch. 0 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 ch. 0.

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 ch. 0, while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register ch. 0. 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 ch. 0, 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 ch. 0, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, write the upper timer data to T01DR and lower timer data to T00DR, and read T01DR first and then T00DR.

● Input capture function

The 8/16-bit composite timer 00/01 data register ch. 0 (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 ch. 0.

If new data is written to the 8/16-bit composite timer 00/01 data register ch. 0, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, write the upper timer data to T01DR and lower timer data to T00DR, and read T01DR first and then T00DR.

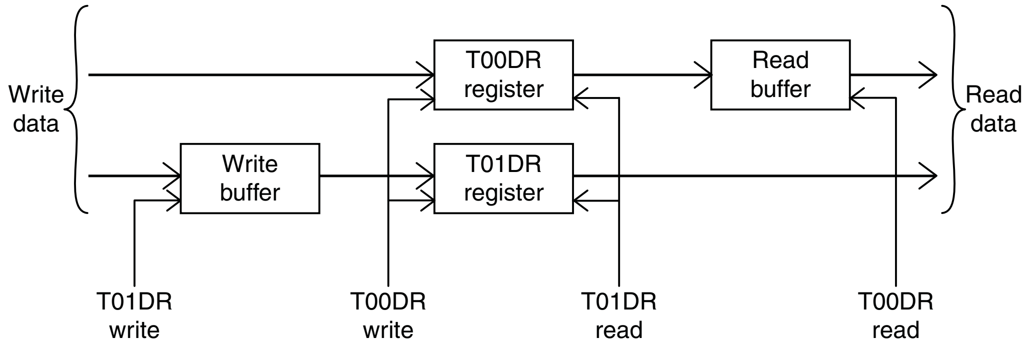
● Read and write operations

Read and write operations of T00DR and T01DR are performed in the following manner in 16-bit operation or when 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 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
8/16-bit composite timer ch. 0 (lower) / Timer 00	IRQ05	ILR1	L05	$FFF0_H$	$FFF1_H$
8/16-bit composite timer ch. 0 (upper) / Timer 01	IRQ06	ILR1	L06	$FFEE_H$	$FFEF_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

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) (Timer 0)

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 (One-shot Mode) (Timer 0)

	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	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval time (counter compare value)							

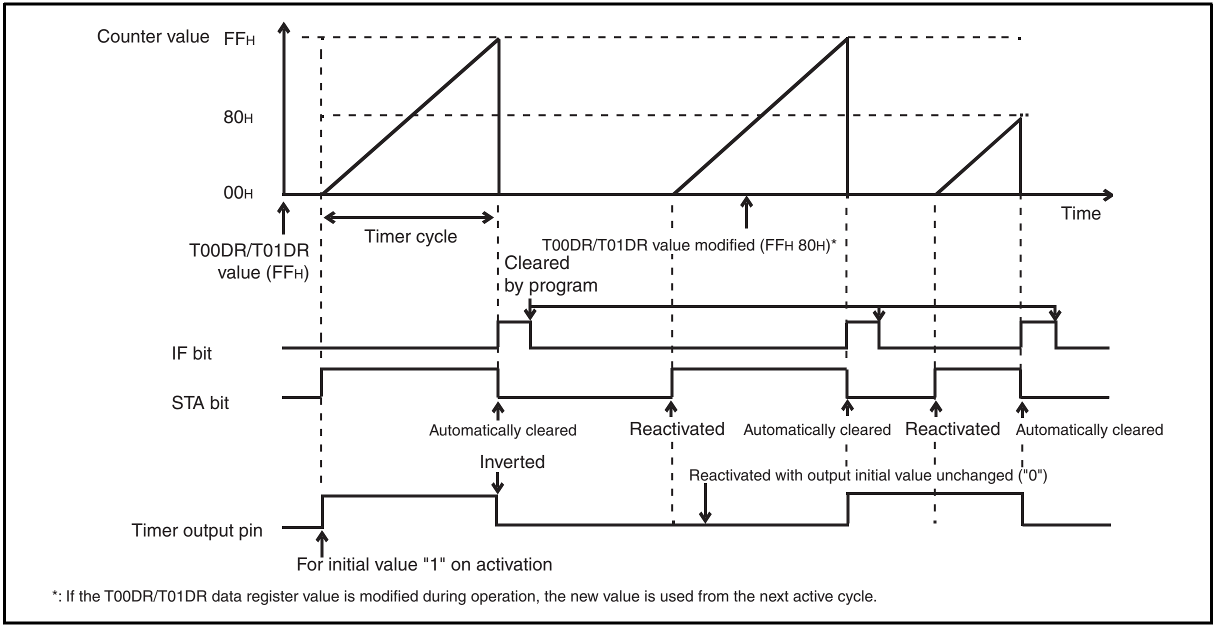
❍: Used bit
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the interval timer function (one-shot mode), enabling timer operation (T00CR1/T01CR1: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 ch. 0 (T00DR/T01DR), the timer output (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1", the start bit (T00CR1/T01CR1:STA) is set to "0", and the counter stops counting.

The value of the 8/16-bit composite timer 00/01 data register ch. 0 (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 ch. 0.

Figure 14.7-2 shows the operation of the interval timer function (timer 0) in 8-bit operation.

Figure 14.7-2 Operation of Interval Timer Function in 8-bit Operation (One-shot 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) (Timer 0)

The register settings shown in Figure 14.8-1 are required to use interval timer function (continuous mode).

Figure 14.8-1 Settings for Interval Timer Function (Continuous Mode) (Timer 0)

	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	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval time (counter compare value)							

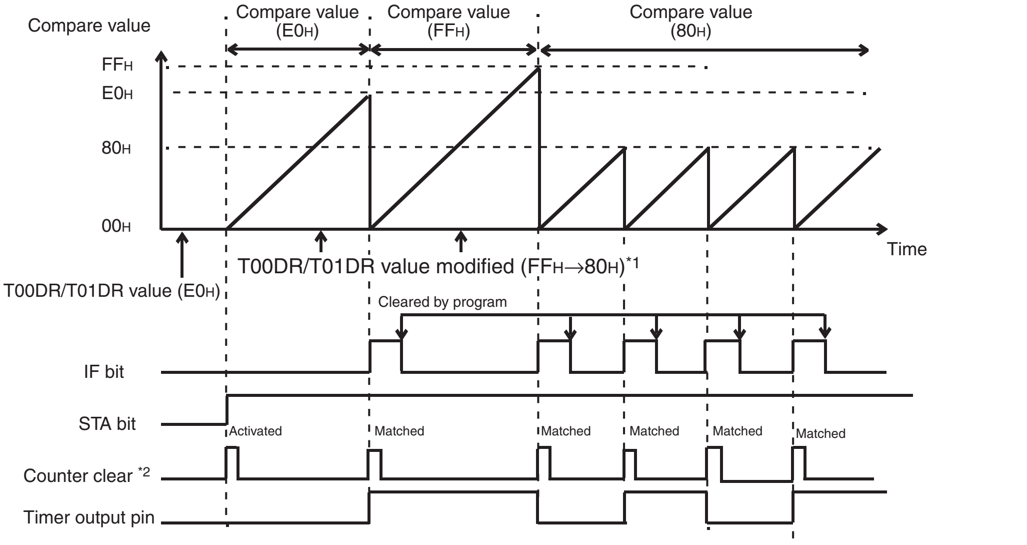
❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the interval timer function (continuous mode), enabling timer operation (T00CR1/T01CR1: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 ch. 0 (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 ch. 0 (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 ch. 0 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 Operation Diagram of Interval Timer Function (Continuous Mode) (Timer 0)



*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) (Timer 0)

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) (Timer 0)

	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	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T00DR/T01DR	Sets interval time (counter compare value)							

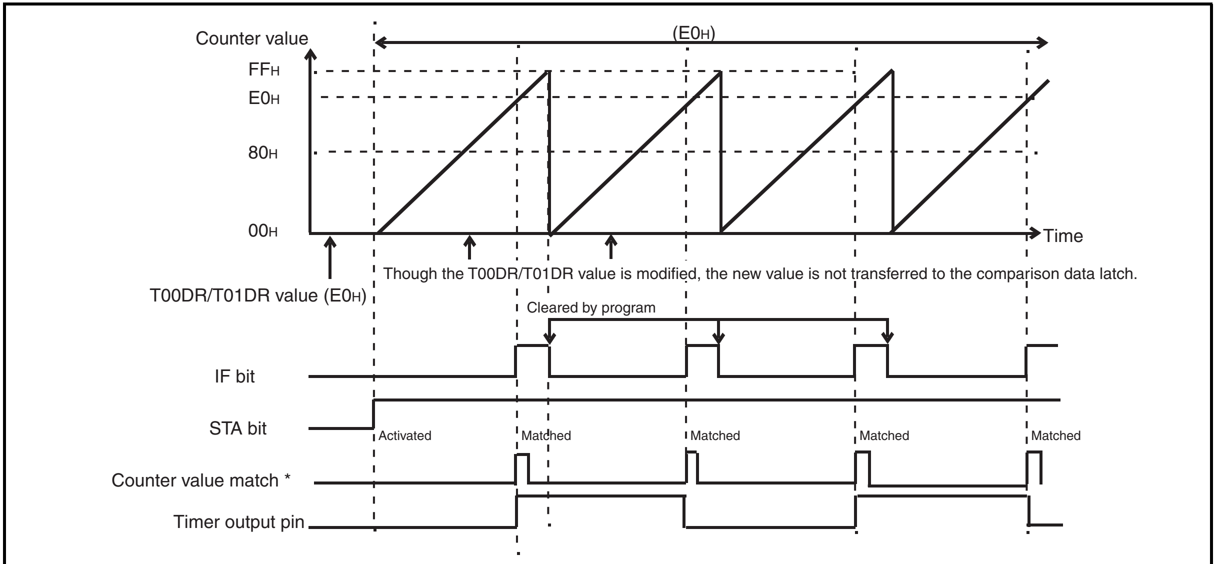
❍: Bit to be used
×: Unused bit
1: Set to "1"
0: Set to "0"

As for the interval timer function (free-run mode), enabling timer operation (T00CR1/T01CR1: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 ch. 0 (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 ch. 0 (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 ch. 0.

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) (Timer 0)



*: 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) (Timer 0)

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) (Timer 0)

	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
	❍	❍	×	×	×	×	×	❍
TMCR0	TO1	TO0	TIS	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 ch. 0 (T00DR/T01DR).

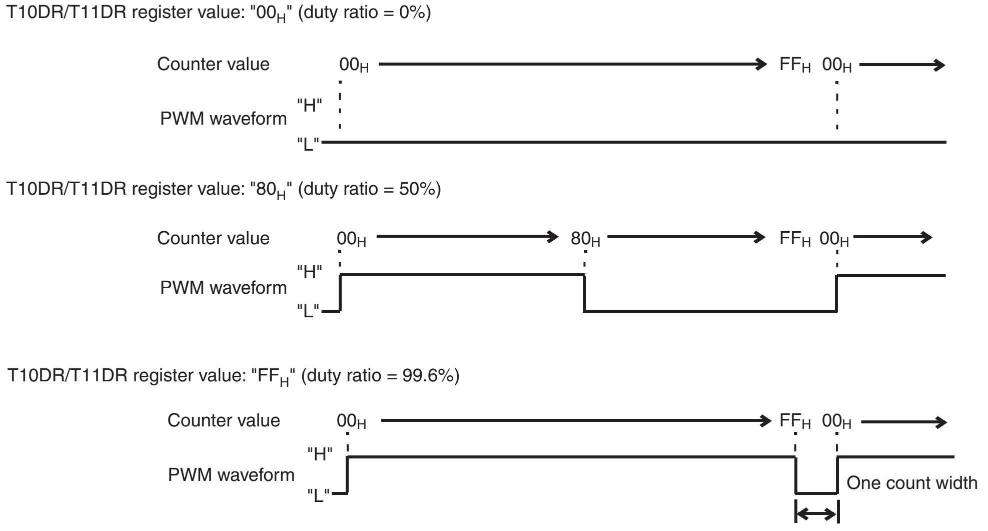
This function has no effect on the interrupt flag (T00CR1/T01CR1:IF). Since each cycle always starts with "H" pulse output, the timer output initial value setting bit (T00CR1/T01CR1:SO) has no effect on operation.

The value of the 8/16-bit composite timer 00/01 data register ch. 0 (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) (Timer 0)


T10DR/T11DR register value: "00H" (duty ratio = 0%)
Counter value
00H
FFH
00H
PWM waveform
"H"
"L"
T10DR/T11DR register value: "80H" (duty ratio = 50%)
Counter value
00H
80H
FFH
00H
PWM waveform
"H"
"L"
T10DR/T11DR register value: "FFH" (duty ratio = 99.6%)
Counter value
00H
FFH
00H
PWM waveform
"H"
"L"
One count width

Note: When the PWM function has been selected, the timer output pin holds the level at the point when the counter stops (T10CR0/T11CR0: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) (Timer 0)

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) (Timer 0)

	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	TIS	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 ch. 0 (T00DR/T01DR).

The 8/16-bit composite timer 00/01 data register ch. 0 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.

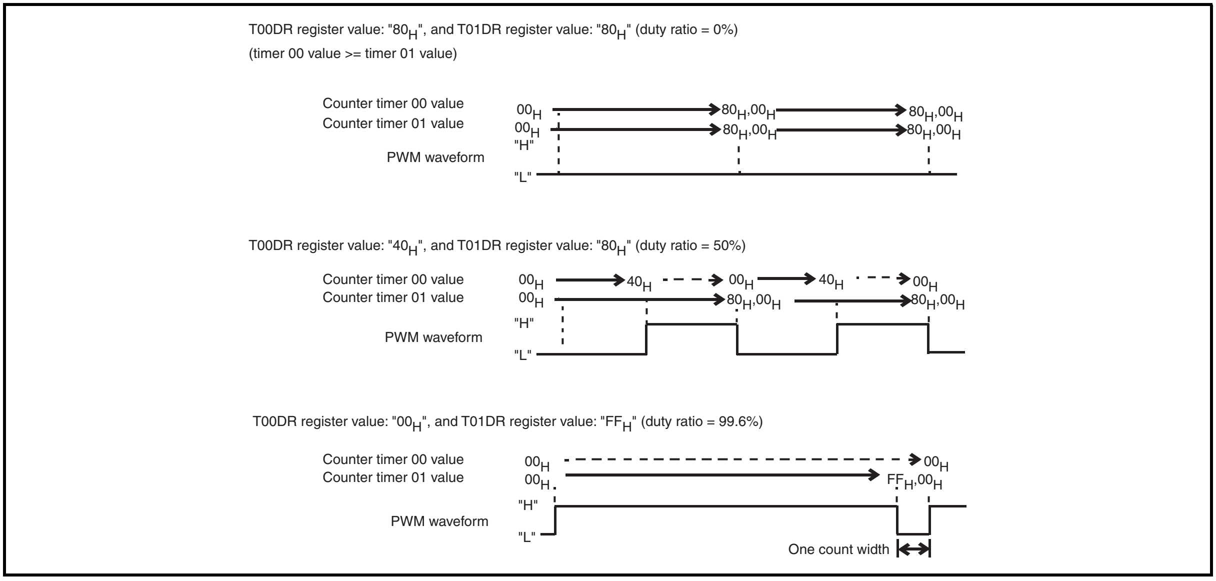
"H" is not output when the "L" pulse width setting value is greater than the cycle setting value.

The count clock must be selected for both 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 ch. 0 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) (Timer 0)



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 (Timer 0)

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 (Timer 0)

	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	TIS	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 ch. 0 (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 ch. 0 (T00DR/T01DR) is read.

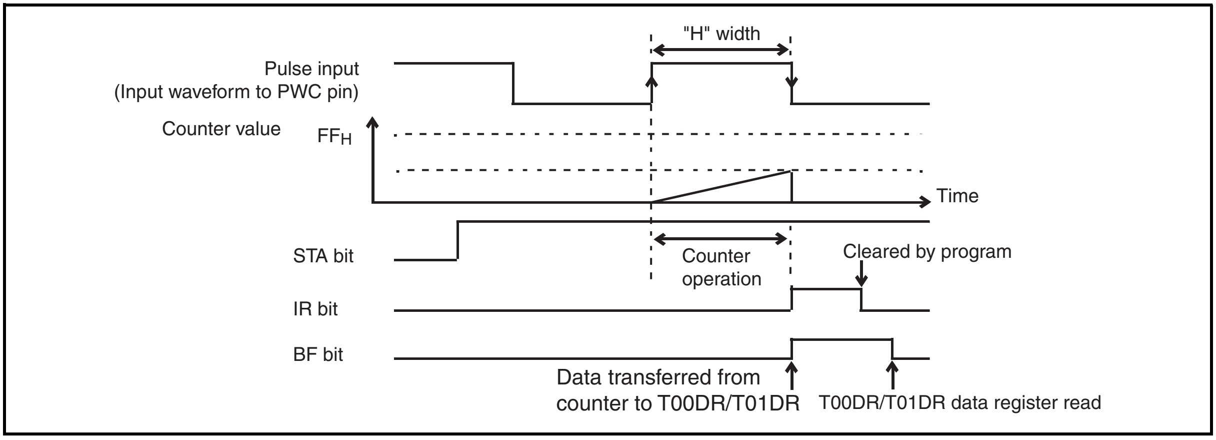
If the buffer full flag is set to "1", the 8/16-bit composite timer 00/01 data register ch. 0 holds data. Even if the next edge is detected during that time, the next measurement result is lost since the count value has not been transferred to the 8/16-bit composite timer 00/01 data register ch. 0.

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 ch. 0, while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register ch. 0. 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.

Figure 14.12-2 Operation Diagram of PWC Timer (Example of H-pulse Width Measurement) (Timer 0)



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 (Timer 0)

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 (Timer 0)

	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	TIS	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 ch. 0 (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 ch. 0 (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 ch. 0 (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) is set to "1". In this case, the counter continues to count without being cleared.

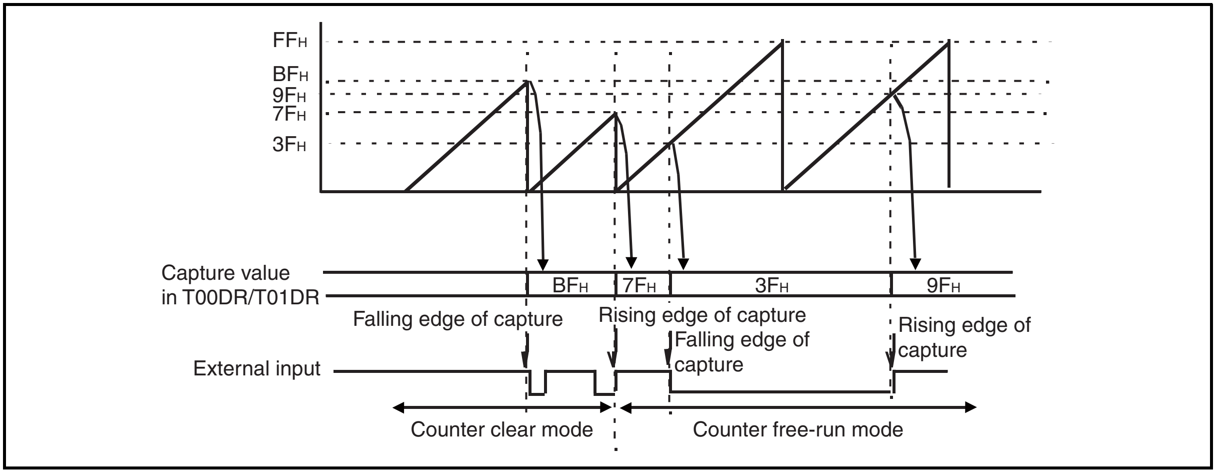
This function has no effect on the buffer full flag (T00CR1/T01CR1:BF).

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).

Note:

See "14.16 Notes on Using 8/16-bit Composite Timer" for notes on using the input capture function.

Figure 14.13-2 Operating Diagram of Input Capture Function (Timer 0)

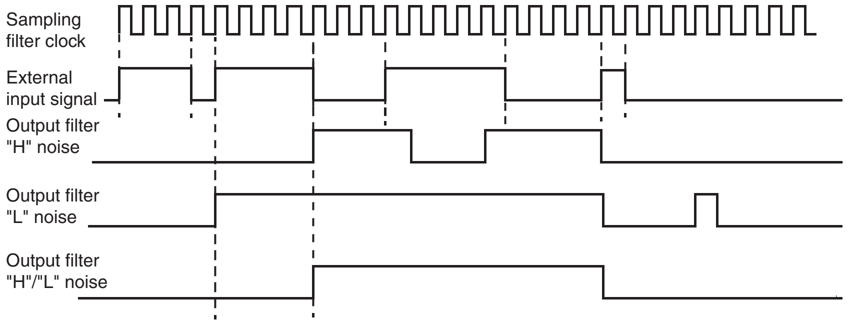


14.14 Operation of Noise Filter

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). 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)

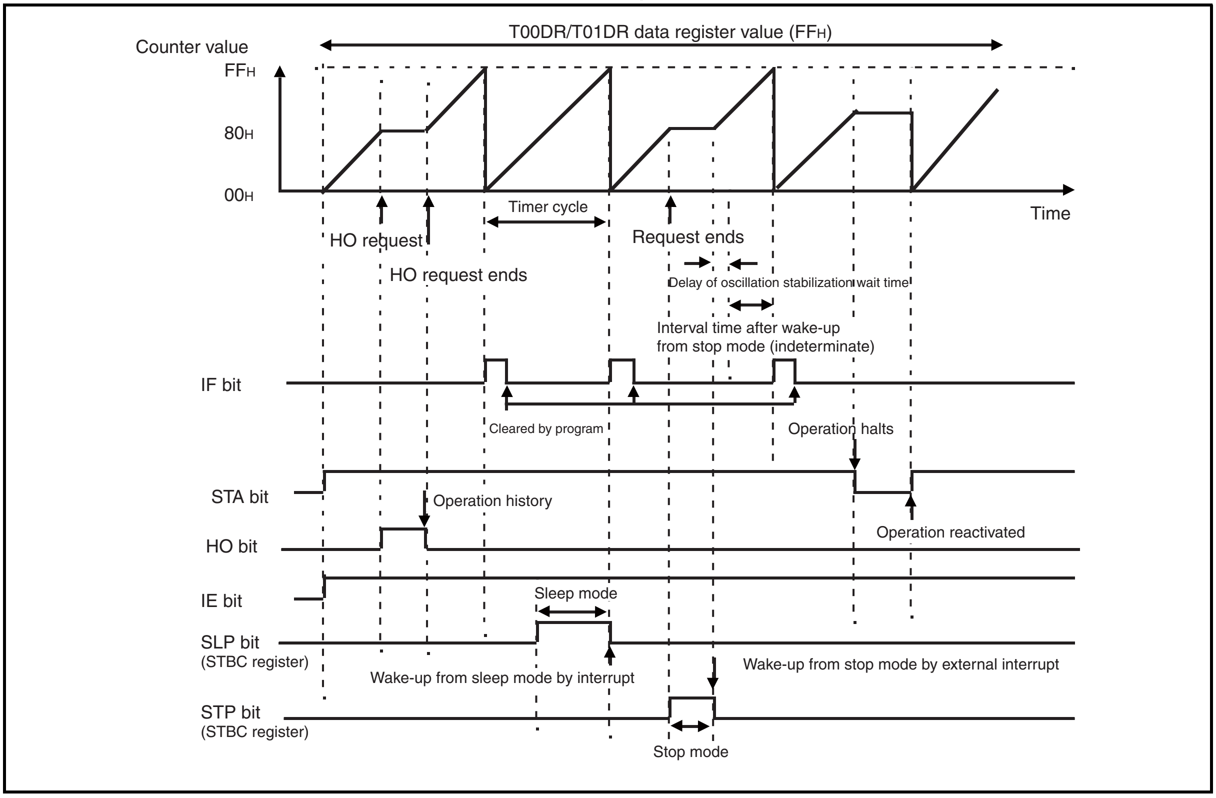
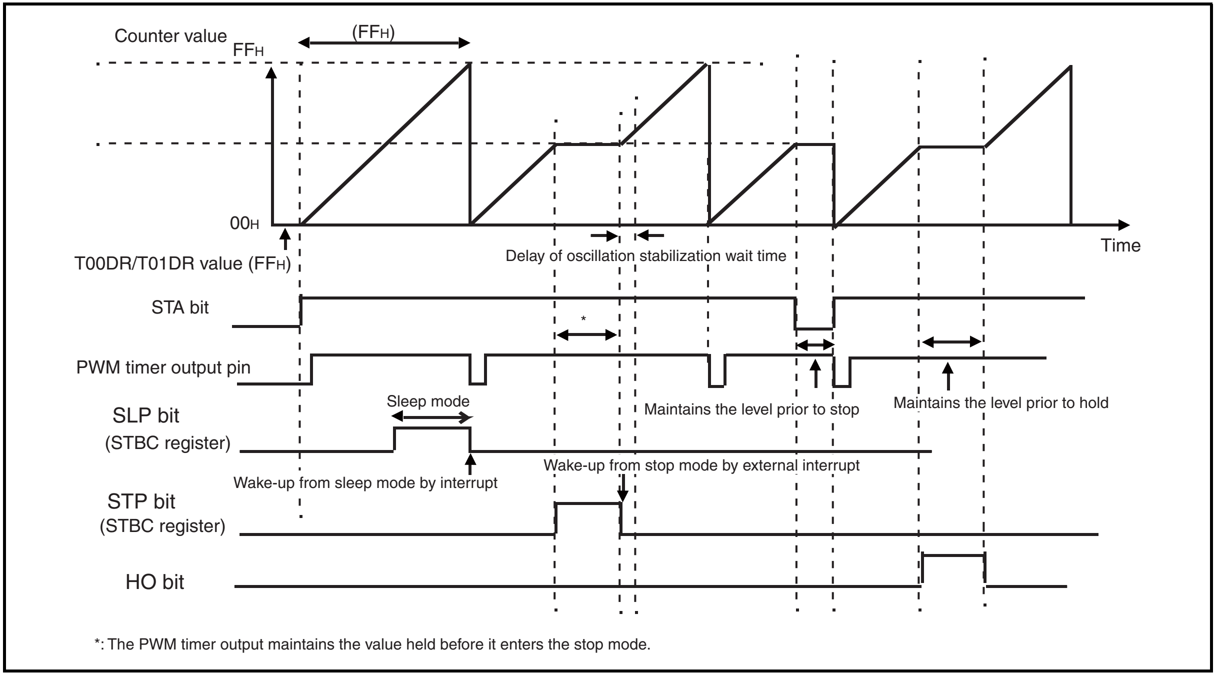


Figure 14.15-2 Operations of Counter in Standby Mode or in Pause (Serving as PWM Timer)


Counter value
FFH
(FFH)
00H
Time
T00DR/T01DR value (FFH)
Delay of oscillation stabilization wait time
STA bit
*
PWM timer output pin
Sleep mode
Maintains the level prior to stop
Maintains the level prior to hold
SLP bit
(STBC register)
Wake-up from stop mode by external interrupt
Wake-up from sleep mode by interrupt
STP bit
(STBC register)
HO bit

*: 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 case of using the input capture function, when both edges of the external input signal is selected as the timing at which the 8/16-bit composite timer captures a counter value (T00CR0/T01CR0:F3, F2, F1, F0 = 1100_B or 1111_B) while "H" level external input signal is being input, the first falling edge will be ignored, no counter value will be transferred to the data register (T00DR/T01DR), and pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) will not be set either.
  - In counter clear mode, the counter will not be cleared at the first falling edge and no data will be transferred to the data register either. The 8/16-bit composite timer will start the input capture operation from the next rising edge.
  - In counter free-run mode, no data will be transferred to the data register at the first falling edge. The 8/16-bit composite timer will start the input capture operation from the next rising edge.
- In 8-bit operating mode (TMCR0:MOD = 0) of the PWM timer function (variable cycle mode), when modifying the 8/16-bit composite timer 00/01 data register ch. 0 ch. 0 (T00DR/T01DR) during counter operation, modify T01DR first and then T00DR.

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

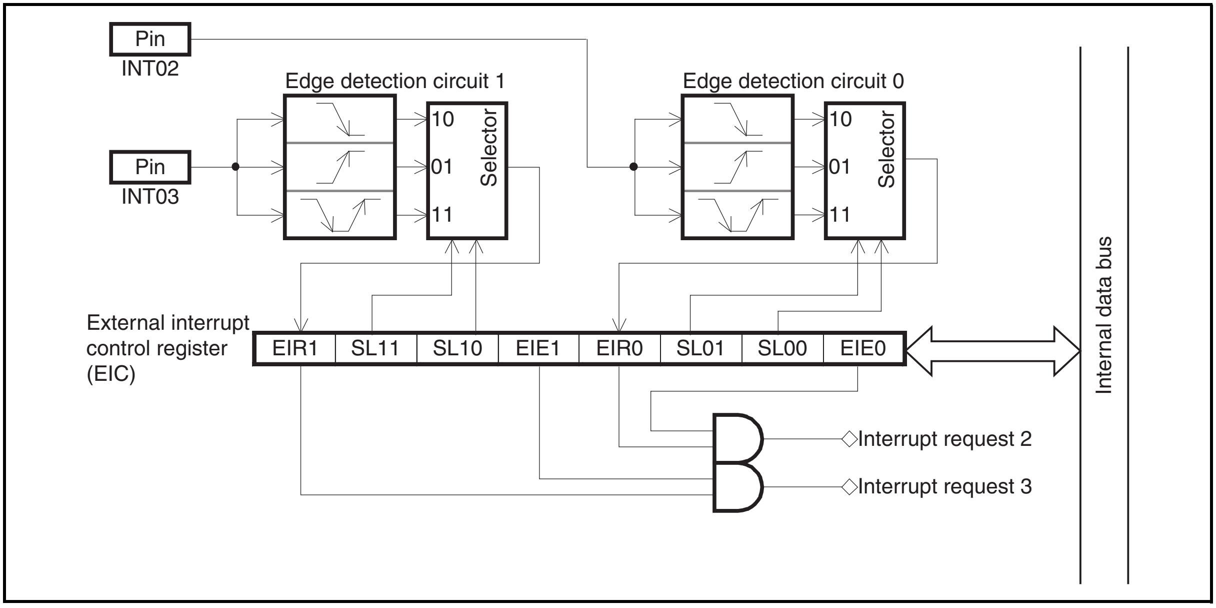
The external interrupt circuit consists of the following blocks:

- **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 MB95430H Series has four 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
0	INT00	External interrupt input ch. 0
	INT01	External interrupt input ch. 1
1	INT02	External interrupt input ch. 2
	INT03	External interrupt input ch. 3
2	INT04	External interrupt input ch. 4
	INT05	External interrupt input ch. 5
3	INT06	External interrupt input ch. 6
	INT07	External interrupt input ch. 7

Table 15.3-2 Registers of External Interrupt Circuit

Unit	Register abbreviation	Corresponding register (Name in this manual)
0	EIC00	EIC: External Interrupt Control register
1	EIC10	
2	EIC20	
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 of the external interrupt circuit and the block diagrams of such pins.

■ Pins of External Interrupt Circuit

In the MB95430H Series, the pins related to the external interrupt circuit are the INT00 to INT07 pins.

● INT00 to INT07 pins

These pins serve both as external interrupt input pins and as general-purpose I/O ports.

INT00 to INT07: If a pin of INT00 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 (INT00 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.

■ Block Diagrams of Pins of External Interrupt Circuit

Figure 15.4-1 Block Diagram of Pins INT00 and INT07 (P00/INT00/AN00, P07/INT07/AN07/EC0) of External Interrupt Circuit

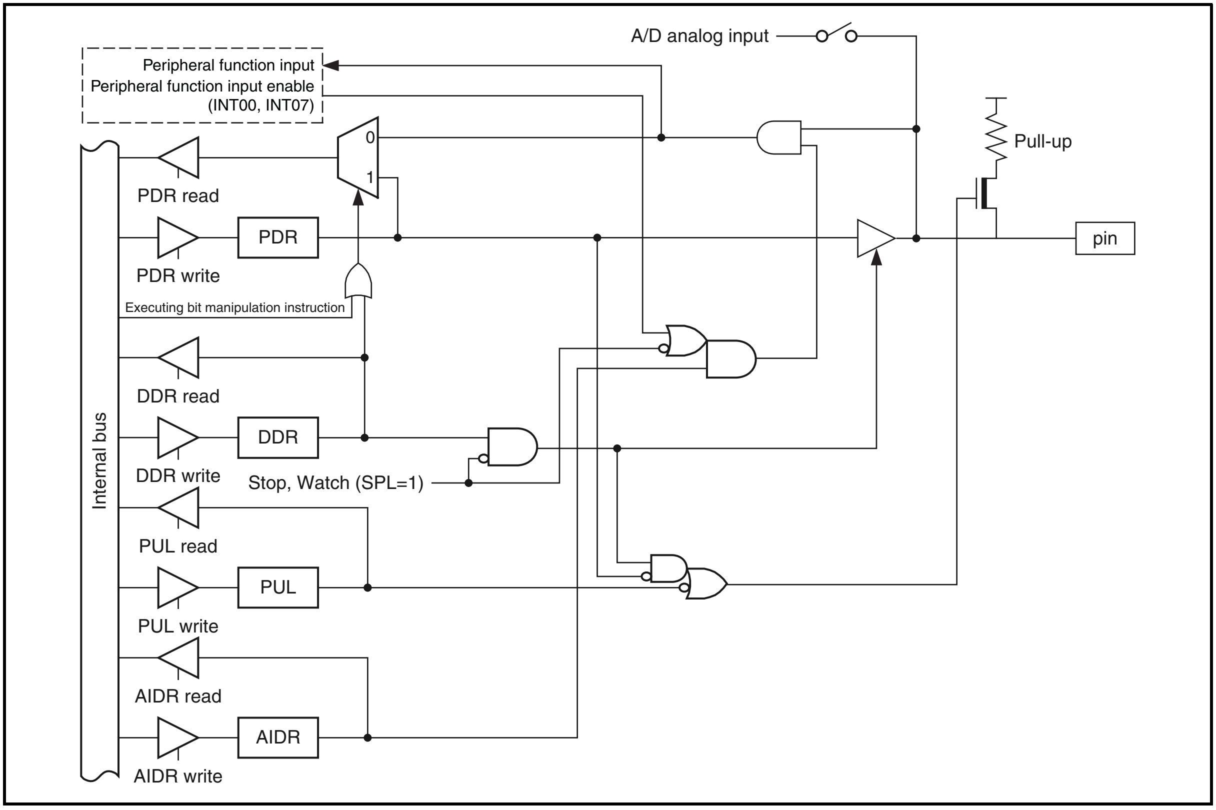


Figure 15.4-2 Block Diagram of Pins INT03 and INT04 (P03/INT03/AN03/UO/SDA, P04/INT04/AN04/UI/SCL) of External Interrupt Circuit

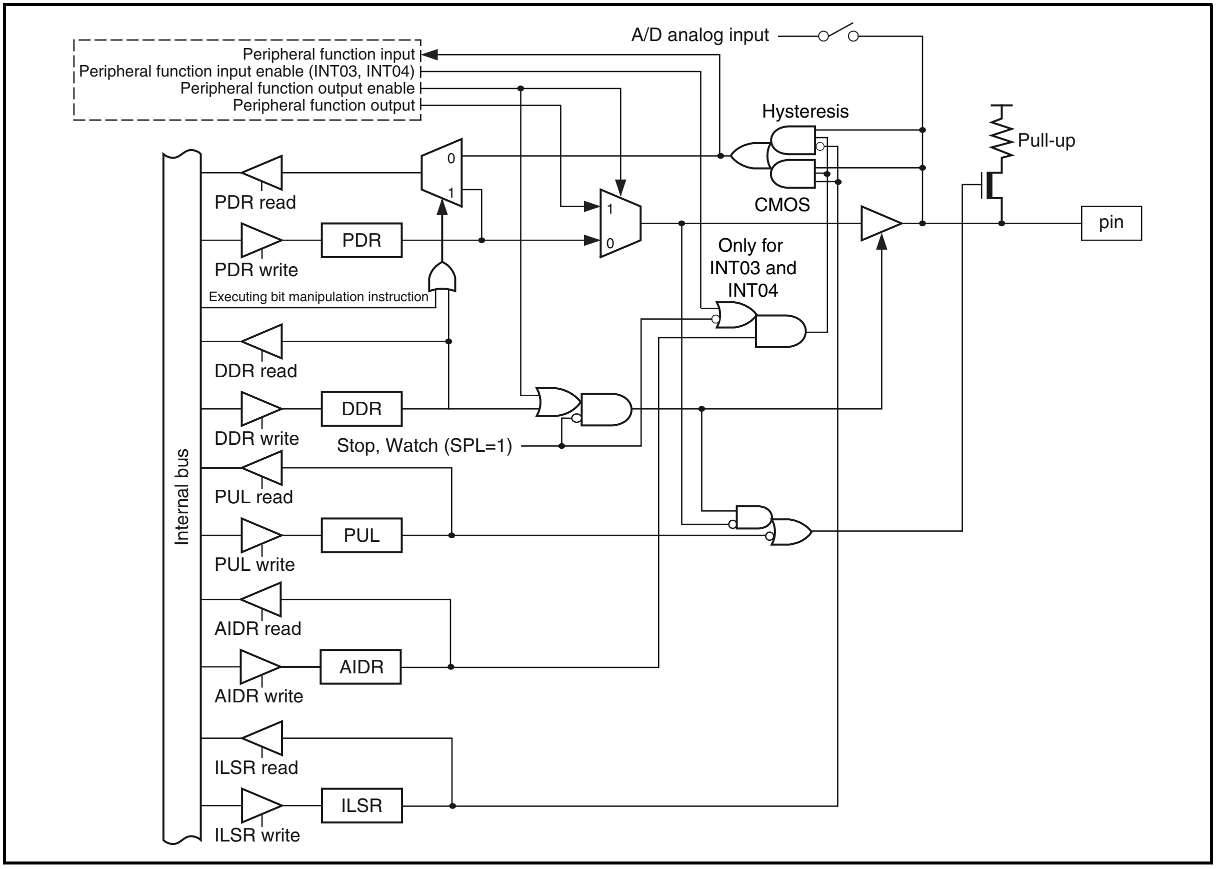
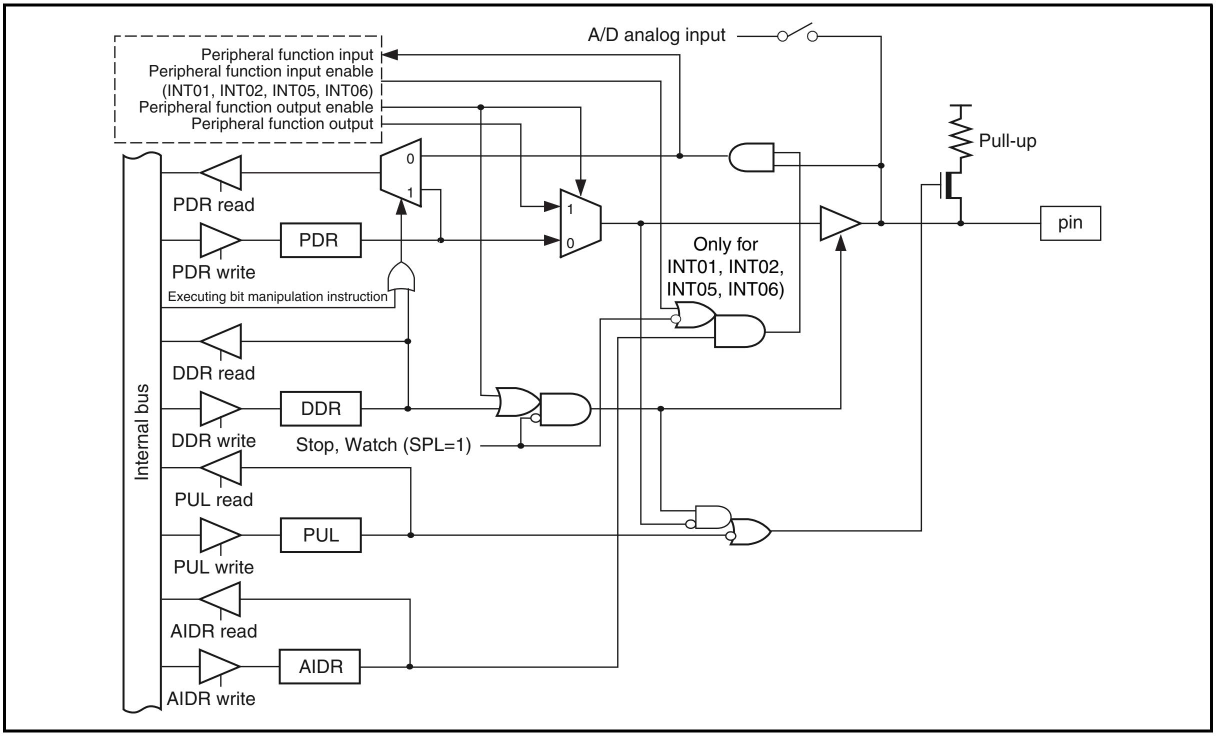


Figure 15.4-3 Block Diagram of Pins INT01, INT02, INT05 and INT06 (P01/INT01/AN01/BZ, P02/INT02/AN02/UCK, P05/INT05/AN05/TO0, P06/INT06/AN06/TO01) of External Interrupt Circuit


A/D analog input
Peripheral function input
Peripheral function input enable (INT01, INT02, INT05, INT06)
Peripheral function output enable
Peripheral function output
Pull-up
pin
0
1
1
0
PDR read
PDR
PDR write
Only for INT01, INT02, INT05, INT06)
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
Internal bus
PUL read
PUL
PUL write
AIDR read
AIDR
AIDR write

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
EIC00	0048_H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
		R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC10	0049_H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
		R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC20	$004A_H$	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
		R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC30	$004B_H$	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000_B
		R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	

R/W : Readable/writable (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.)

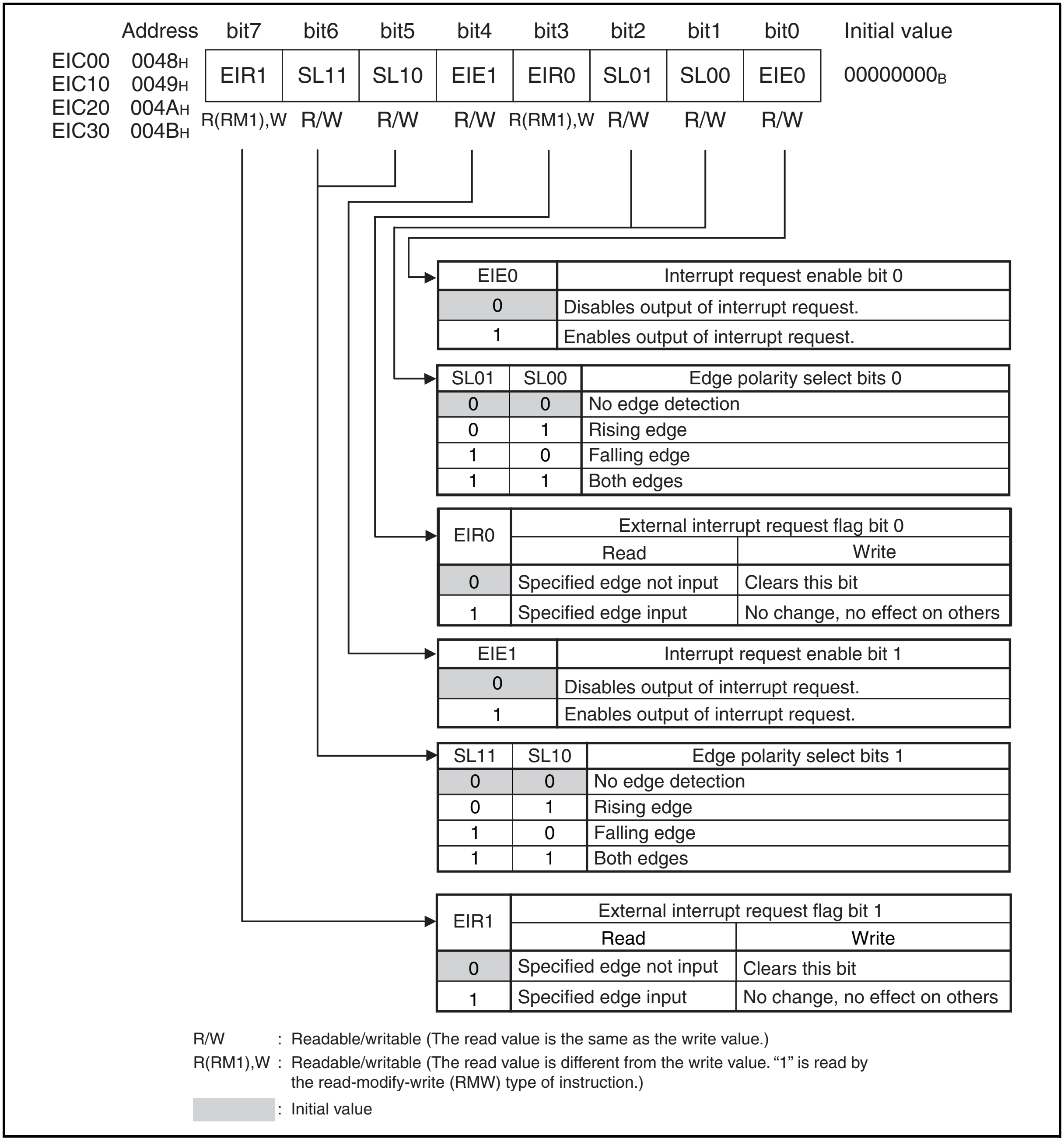
MB95430H Series

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
EIC00 EIC10 EIC20 EIC30	0048H 0049H 004AH 004BH	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000B
		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 this bit. Writing "1" has no effect on operation. • When read by the read-modify-write (RMW) type of instruction, this bit always 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 this bit. Writing "1" has no effect on operation. • When read by the read-modify-write (RMW) type of instruction, this bit always 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 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
External interrupt ch. 0	IRQ00	ILR0	L00	$FFFA_H$	$FFFB_H$
External interrupt ch. 4					
External interrupt ch. 1	IRQ01	ILR0	L01	$FFF8_H$	$FFF9_H$
External interrupt ch. 5					
External interrupt ch. 2	IRQ02	ILR0	L02	$FFF6_H$	$FFF7_H$
External interrupt ch. 6					
External interrupt ch. 3	IRQ03	ILR0	L03	$FFF4_H$	$FFF5_H$
External interrupt ch. 7					

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different 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

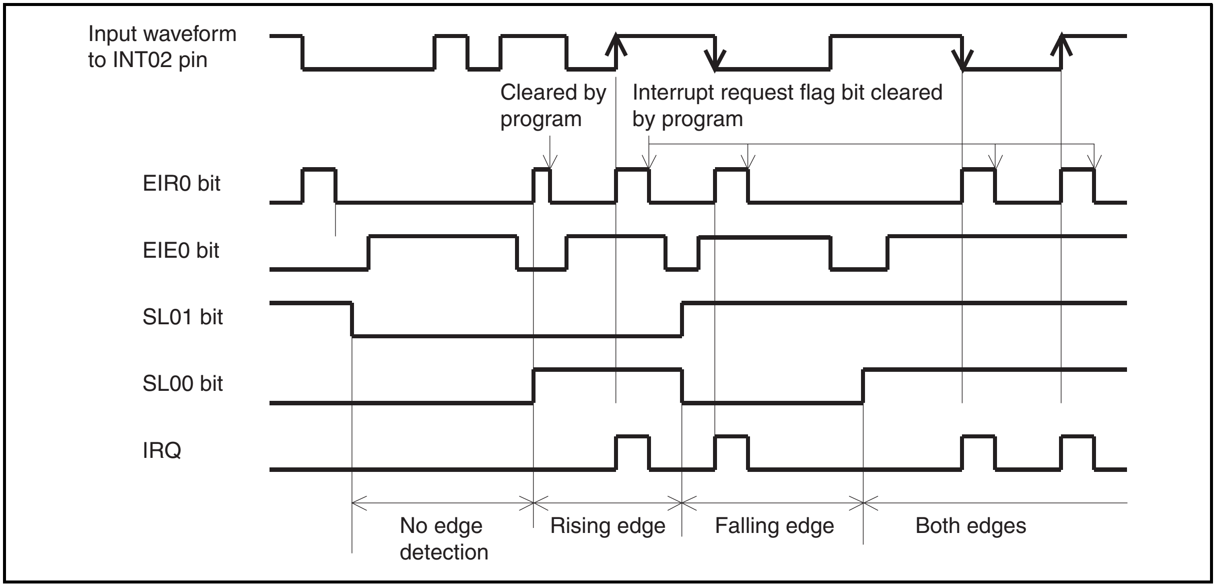
When the polarity of an edge of a signal input from one of the external interrupt pins (INT02, INT03) matches the polarity of the edge selected by the external interrupt control register (EIC: SL01, SL00 or EIC:SL11, SL10), the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set 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). 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 Example of Setting External Interrupt Circuit

This section describes the example of external interrupt circuit.

■ Example of Setting Methods

● 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 or SL11, SL10)
No edge detection	Set the bits to "00_B"
Detecting rising edges	Set the bits to "01_B"
Detecting falling edges	Set the bits to "10_B"
Detecting both edges	Set the bits 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 (P00 to P07)	Setting
Using INT00 pin for external interrupt	DDR0: P00	Set to "0"
Using INT01 pin for external interrupt	DDR0: P01	Set to "0"
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. 0	Interrupt level register (ILR0) Address: 00079$_H$	#0 Address: 0FFFA$_H$
ch. 1	Interrupt level register (ILR0) Address: 00079$_H$	#1 Address: 0FFF8$_H$
ch. 2	Interrupt level register (ILR0) Address: 00079$_H$	#2 Address: 0FFF6$_H$
ch. 3	Interrupt level register (ILR0) Address: 00079$_H$	#3 Address: 0FFF4$_H$
ch. 4	Interrupt level register (ILR0) Address: 00079$_H$	#0 Address: 0FFFA$_H$
ch. 5	Interrupt level register (ILR0) Address: 00079$_H$	#1 Address: 0FFF8$_H$
ch. 6	Interrupt level register (ILR0) Address: 00079$_H$	#2 Address: 0FFF6$_H$
ch. 7	Interrupt level register (ILR0) Address: 00079$_H$	#3 Address: 0FFF4$_H$

- How to enable/disable/clear 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 the bit to "0".
To enable an interrupt request	Set the bit 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 the bit to "0".

CHAPTER 16

INTERRUPT PIN SELECTION CIRCUIT

This chapter describes the functions and operations of the interrupt pin selection circuit.

16.1 Overview of Interrupt Pin Selection Circuit

The interrupt pin selection circuit selects pins to be used as interrupt input pins from among various peripheral input pins.

■ Interrupt Pin Selection Circuit

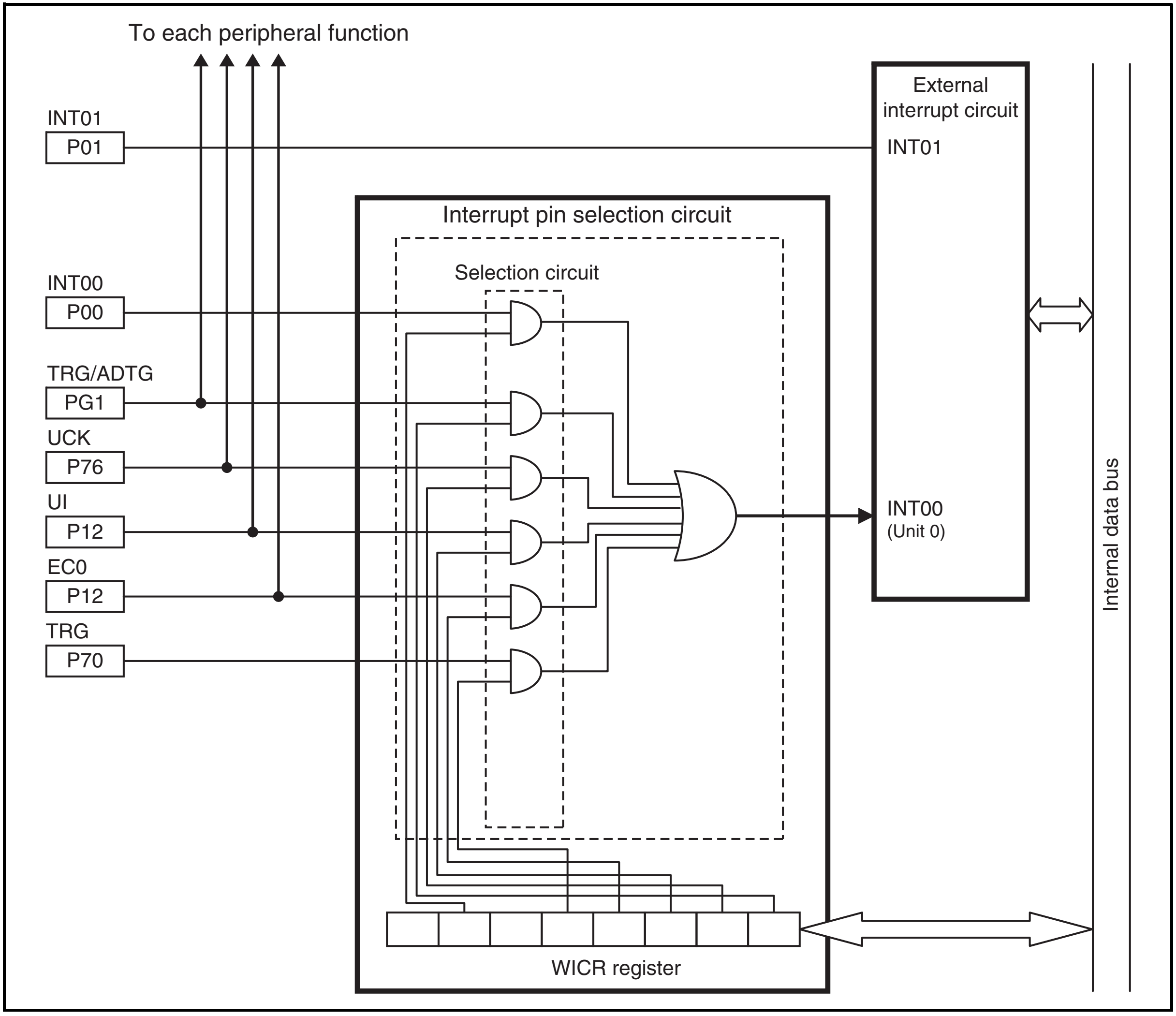
The interrupt pin selection circuit is used to select interrupt input pins from amongst various peripheral inputs (TRG/ADTG, EC0, UI, UCK, TRG and INT00). The input signal from each peripheral function pin is selected by this circuit and the signal is used as the INT00 (channel 0) input of external interrupt. This enables the input signals to the peripheral function pins to also serve as external interrupt pins.

16.2 Configuration of Interrupt Pin Selection Circuit

Figure 16.2-1 shows the block diagram of the interrupt pin selection circuit.

■ Block Diagram of Interrupt Pin Selection Circuit

Figure 16.2-1 Block Diagram of Interrupt Pin Selection Circuit



- WICR register (interrupt pin selection circuit control register)

  This register is used to determine which of the available peripheral input pins should be output to the interrupt circuit and which interrupt pins they should serve as.

- Selection circuit

  This circuit outputs the input from the pin selected by the WICR register to the INT00 input of the external interrupt circuit (ch. 0).

16.3 Pins of Interrupt Pin Selection Circuit

This section describes the pins of the interrupt pin selection circuit.

■ Pins of Interrupt Pin Selection Circuit

The peripheral function pins of the interrupt pin selection circuit are the TRG/ADTG(PG1), UCK(P76), UI(P12), EC0(P12), TRG(P70) and INT00 pins. These inputs (except INT00) are also connected to their respective peripheral units in parallel and can be used for both functions simultaneously. Table 16.3-1 shows the correspondence between the peripheral functions and peripheral input pins.

Table 16.3-1 Correspondence between Peripheral Functions and Peripheral Input Pins

Peripheral input pin name	Peripheral functions name
INT00	Interrupt pin selection circuit
TRG/ADTG	Interrupt pin seelction circuit A/D (trigger input)
	Interrupt pin seelction circuit 16-bit PPG timer (trigger input)
UCK	Interrupt pin seelction circuit UART/SIO (clock input/output)
UI	Interrupt pin seelction circuit UART/SIO (data input)
EC0	Interrupt pin seelction circuit 8/16-bit composite timer (event input)
TRG	Interrupt pin seelction circuit 16-bit PPG timer (trigger input)

16.4 Register of Interrupt Pin Selection Circuit

Figure 16.4-1 shows the register of the interrupt pin selection circuit.

■ Register of Interrupt Pin Selection Circuit

Figure 16.4-1 Register of Interrupt Pin Selection Circuit

Interrupt pin selection circuit control register (WICR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FEF_H$	-	INT00	-	TRG	EC0	UI	UCK	TRG/ADTG	01000000_B
	R0/WX	R/W	R0/WX	R/W	R/W	R/W	R/W	R/W	

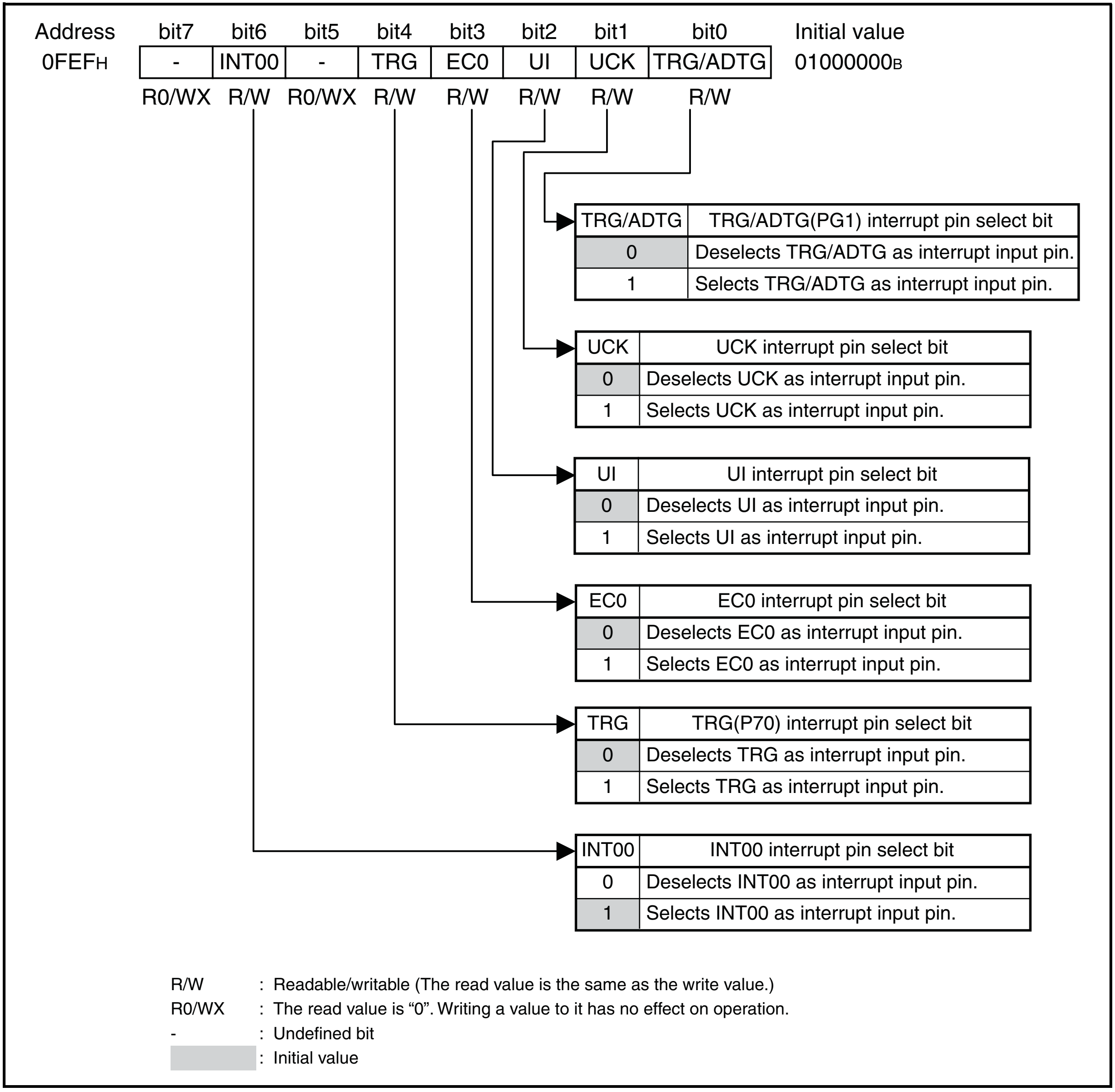
R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
- : Undefined bit

16.4.1 Interrupt Pin Selection Circuit Control Register (WICR)

This register is used to determine which of the available peripheral input pins should be output to the interrupt circuit and which interrupt pins they should serve as.

■ Interrupt Pin Selection Circuit Control Register (WICR)

Figure 16.4-2 Interrupt Pin Selection Circuit Control Register (WICR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FEFH	-	INT00	-	TRG	EC0	UI	UCK	TRG/ADTG	01000000B
	R0/WX	R/W	R0/WX	R/W	R/W	R/W	R/W	R/W	

TRG/ADTG	TRG/ADTG(PG1) interrupt pin select bit
0	Deselects TRG/ADTG as interrupt input pin.
1	Selects TRG/ADTG as interrupt input pin.

UCK	UCK interrupt pin select bit
0	Deselects UCK as interrupt input pin.
1	Selects UCK as interrupt input pin.

UI	UI interrupt pin select bit
0	Deselects UI as interrupt input pin.
1	Selects UI as interrupt input pin.

EC0	EC0 interrupt pin select bit
0	Deselects EC0 as interrupt input pin.
1	Selects EC0 as interrupt input pin.

TRG	TRG(P70) interrupt pin select bit
0	Deselects TRG as interrupt input pin.
1	Selects TRG as interrupt input pin.

INT00	INT00 interrupt pin select bit
0	Deselects INT00 as interrupt input pin.
1	Selects INT00 as interrupt input pin.

R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 16.4-1 Functions of Bits in Interrupt Pin Selection Circuit Control Register (WICR)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation
bit6	INT00: INT00 interrupt pin select bit	This bit is used to determine whether to select the INT00 pin as an interrupt input pin. **Writing "0"** to the bit deselects the INT00 pin as an interrupt input pin and the circuit treats the INT00 pin input as being fixed at "0". **Writing "1"** to the bit selects the INT00 pin as an interrupt input pin and the circuit passes the INT00 pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the INT00 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit.
bit5	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation
bit4	TRG: TRG(P70) interrupt pin select bit	This bit is used to determine whether to select the TRG pin as an interrupt input pin. **Writing "0"** to the bit deselects the TRG pin as an interrupt input pin and the circuit treats the TRG1 pin input as being fixed at "0". **Writing "1"** to the bit selects the TRG pin as an interrupt input pin and the circuit passes the TRG pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the TRG1 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit. Note: Select P70 as the TRG pin before setting this bit to select the EC0 pin as an interrupt input pin.
bit3	EC0: EC0 interrupt pin select bit	This bit is used to determine whether to select the EC0 pin as an interrupt input pin. **Writing "0"** to the bit deselects the EC0 pin as an interrupt input pin and the circuit treats the EC0 pin input as being fixed at "0". **Writing "1"** to the bit selects the EC0 pin as an interrupt input pin and the circuit passes the EC0 pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the EC0 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit. Note: Select P12 as the TRG pin before setting this bit to select the EC0 pin as an interrupt input pin.
bit2	UI: UI interrupt pin select bit	This bit is used to determine whether to select the UI pin as an interrupt input pin. **Writing "0"** to the bit deselects the UI pin as an interrupt input pin and the circuit treats the UI pin input as being fixed at "0". **Writing "1"** to the bit selects the UI pin as an interrupt input pin and the circuit passes the UI pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the UI pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit. Note: Select P12 as the UI pin before setting this bit to select the UI pin as an interrupt input pin.
bit1	UCK: UCK interrupt pin select bit	This bit is used to determine whether to select the UCK pin as an interrupt input pin. **Writing "0"** to the bit deselects the UCK pin as an interrupt input pin and the circuit treats the UCK pin input as being fixed at "0". **Writing "1"** to the bit selects the UCK pin as an interrupt input pin and the circuit passes the UCK pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the UCK pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit. Note: Select P76 as the UCK pin before setting this bit to select the UCK pin as an interrupt input pin.
bit0	TRG/ADTG: TRG/ADTG(PG1) interrupt pin select bit	This bit is used to determine whether to select the TRG/ADTG pin as an interrupt input pin. **Writing "0"** to the bit deselects the TRG/ADTG pin as an interrupt input pin and the circuit treats the TRG1 pin input as being fixed at "0". **Writing "1"** to the bit selects the TRG/ADTG pin as an interrupt input pin and the circuit passes the TRG/ADTG pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the TRG/ADTG pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit. Note: Select PG1 as the TRG pin before setting this bit to select the TRG pin as an interrupt input pin.

When these bits are set to "1" and the operation of INT00 (ch. 0) of the external interrupt circuit is enabled in MCU standby mode, the selected pins are enabled to perform input operation. The MCU wakes up from the standby mode when a valid edge pulse is input to the pins. For information about the standby modes, see "6.8 Operations in Low-power Consumption Mode (Standby Mode)".

Note:

The input signals to the peripheral pins do not generate an external interrupt even when "1" is written to these bits if the INT00 (ch. 0) of the external interrupt circuit is disabled.

Do not modify the values of these bits while the INT00 (ch. 0) of the external interrupt circuit is enabled. If modified, the external interrupt circuit may detect a valid edge, depending on the pin input level.
If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch. 0) of the external interrupt circuit is enabled (the values other than "00_B" are written to the SL01, SL00 bits in the EIC00 register of external interrupt circuit.), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.

16.5 Operation of Interrupt Pin Selection Circuit

The interrupt pins are selected by setting WICR (interrupt pin selection circuit control register).

■ Operation of Interrupt Pin Selection Circuit

The WICR (interrupt pin selection circuit control register) setting is used to select the input pins to be input to INT00 of the external interrupt circuit (ch. 0). Shown below is the setup procedure for the interrupt pin selection circuit and external interrupt circuit (ch. 0), which must be followed when selecting the TRG (P70) pin as an interrupt pin.

1) Write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input.

2) Select the TRG1 pin as an interrupt input pin in WICR (interrupt pin selection circuit control register) (Write "01_H" to the WICR register. At this point, after writing "0" in the EIE0 bit of the EIC00 register of the external interrupt circuit, the operation of the external interrupt circuit is disabled).

3) Enable the operation of INT00 of the external interrupt circuit (ch. 0).
(Set the SL01 and SL00 bits in the EIC00 register to any value other than "00_B" in the external interrupt circuit to select the valid edge. Also write "1" to the EIE0 bit to enable interrupts).

4) The subsequent interrupt operation is the same as that of the external interrupt circuit.

When a reset is released, WICR (interrupt pin selection circuit control register) is initialized to "40_H" and the INT00 bit is selected as the only available interrupt pin. Update the value of this register before enabling the operation of the external interrupt circuit, when using any pins other than the INT00 pin as external interrupt pins.

16.6 Notes on Using Interrupt Pin Selection Circuit

This section provides notes on using the interrupt pin selection circuit.

- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch. 0) of the external interrupt circuit is enabled (Set the SL01 and SL00 bits in the EIC00 register to any value other than "00_B" in the external interrupt circuit to select the valid edge. Also write "1" to the EIE0 bit to enable interrupts), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.
- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously, an input to INT00 (ch. 0) of the external interrupt circuit is treated as "H" if any of the selected input signals is "H" (It becomes "OR" of the signals input to the selected pins).

CHAPTER 17
UART/SIO

This chapter describes the functions and operations of UART/SIO.

17.1 Overview of UART/SIO

The UART/SIO is a general-purpose serial data communication interface. Serial data transfers of variable-length data can be made with a synchronous or asynchronous clock. The transfer format is NRZ. The transfer rate can be set with the dedicated baud rate generator or external clock (in clock synchronous mode).

■ Functions of UART/SIO

The UART/SIO is capable of serial data transmission/reception (serial input/output) to and from another CPU or peripheral device.

- Equipped with a full-duplex double buffer that allows 2-way full-duplex communication.
- The synchronous or asynchronous transfer mode can be selected.
- The optimum baud rate can be selected with the dedicated baud rate generator.
- The data length is variable; it can be set to 5 bit to 8 bit when no parity is used or to 6 bit to 9 bit when parity is used. (See Table 17.1-1.)
- The serial data direction (endian) can be selected.
- The data transfer format is NRZ (Non-Return-to-Zero).
- Two operation modes (operation modes 0 and 1) are available.
  Operation mode 0 operates as asynchronous clock mode (UART).
  Operation mode 1 operates as clock synchronous mode (SIO).

Table 17.1-1 UART/SIO Operation Modes

Operation mode	Data length		Synchronization mode	Length of stop bit
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	-	Synchronous	-
	6	-		
	7	-		
	8	-		

MB95430H Series

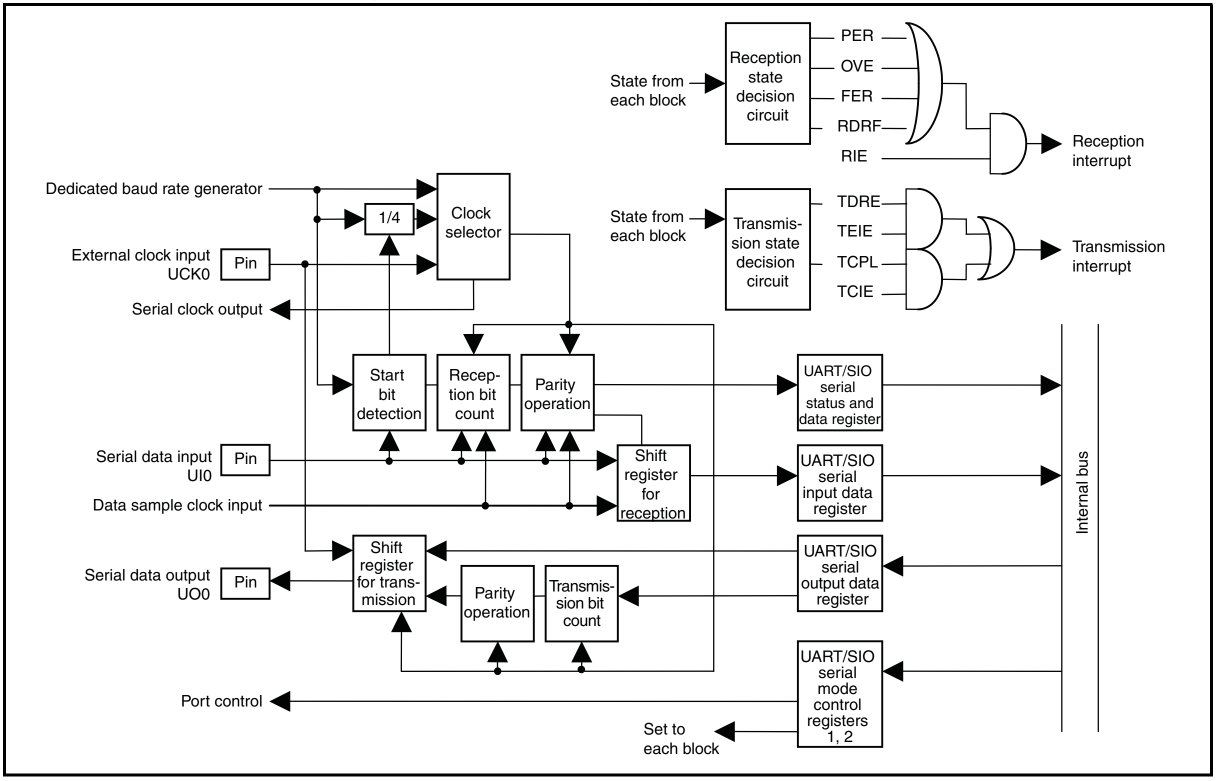
17.2 Configuration of UART/SIO

The UART/SIO consists of the following blocks:

- **UART/SIO serial mode control register 1 (SMC10)**
- **UART/SIO serial mode control register 2 (SMC20)**
- **UART/SIO serial status and data register (SSR0)**
- **UART/SIO serial input data register (RDR0)**
- **UART/SIO serial output data register (TDR0)**

■ Block Diagram of UART/SIO

Figure 17.2-1 Block Diagram of UART/SIO



● UART/SIO serial mode control register 1 (SMC10)

This register controls UART/SIO operation mode. It is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/asynchronous), data length, and serial clock.

● UART/SIO serial mode control register 2 (SMC20)

This register controls UART/SIO operation mode. It is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

● UART/SIO serial status and data register (SSR0)

This register indicates the transmission/reception status and error status of UART/SIO.

● UART/SIO serial input data register (RDR0)

This register holds the receive data. The serial input is converted and then stored in this register.

● UART/SIO serial output data register (TDR0)

This register sets the transmit data. Data written to this register is serial-converted and then output.

■ Input Clock

The UART/SIO uses the output clock (internal clock) from the dedicated baud rate generator or the input signal (external clock) from the UCK pin as its input clock (serial clock).

17.3 Channels of UART/SIO

This section describes the channels of UART/SIO.

■ Channels of UART/SIO

The MB95430H Series has one channel of UART/SIO.

Table 17.3-1 and Table 17.3-2 show the pins and registers of UART/SIO respectively.

Table 17.3-1 Pins of UART/SIO

Channel	Pin name	Pin function
0	UCK	Clock input/output
	UO	Data output
	UI	Data input

Table 17.3-2 Registers of UART/SIO

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	SMC10	UART/SIO serial mode control register 1
	SMC20	UART/SIO serial mode control register 2
	SSR0	UART/SIO serial status and data register
	TDR0	UART/SIO serial output data register
	RDR0	UART/SIO serial input data register

17.4 Pins of UART/SIO

This section describes the pins of the UART/SIO.

■ Pins of UART/SIO

The pins of UART/SIO are the clock input and output pin (UCK), serial data output pin (UO) and serial data input pin (UI).

● UCK

Clock input/output pin for UART/SIO.

When the clock output is enabled (SMC20:SCKE=1), it serves as a UART/SIO clock output pin (UCK) regardless of the value of the corresponding port direction register. At this time, do not select the external clock (set SMC10:CKS = 0).

When it is to be used as a UART/SIO clock input pin, disable the clock output (SMC20:SCKE = 0) and make sure that it is set as input port by the corresponding port direction register. At this time, be sure to select the external clock (set SMC10:CKS = 0).

The UCK pin can be assigned to P02 or P76 by using the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

● UO

Serial data output pin for UART/SIO. When the serial data output is enabled (SMC20:TXOE = 1), it serves as a UART/SIO serial data output pin (UO) regardless of the value of the corresponding port direction register.

The UO pin can be assigned to P03 or P65 by using the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

● UI

Serial data input pin for UART/SIO. When it is to be used as a UART/SIO serial data input pin, make sure that it is set as input port by the corresponding port direction register.

The UI pin can be assigned to P04 or P12 by using the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

■ Block Diagrams of Pins of UART/SIO

Figure 17.4-1 Block Diagram of Pin UO (P03/INT03/AN03/UO/SDA) of UART/SIO

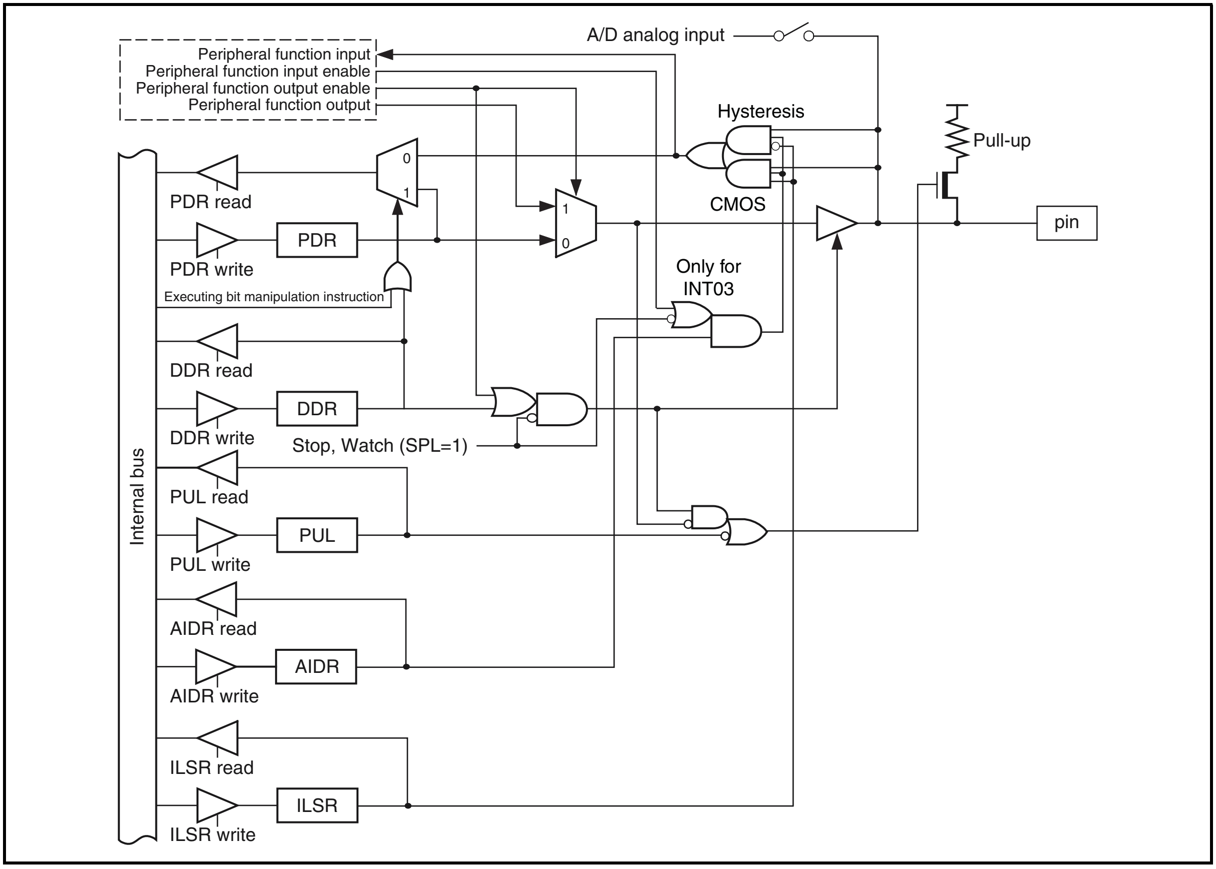


Figure 17.4-2 Block Diagram of Pin UO (P65/CMP3_O/UO/SDA) of UART/SIO

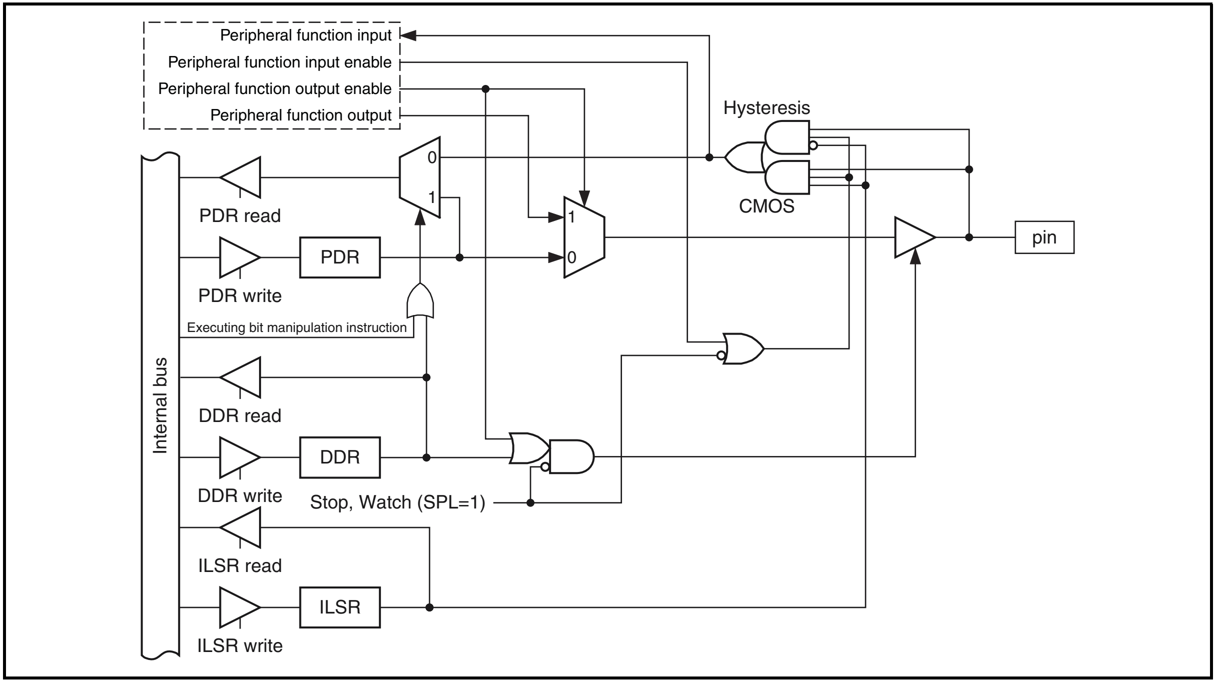

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
0
1
PDR read
CMOS
1
0
pin
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

Figure 17.4-3 Block Diagram of Pin UCK (P02/INT02/AN02/UCK) of UART/SIO

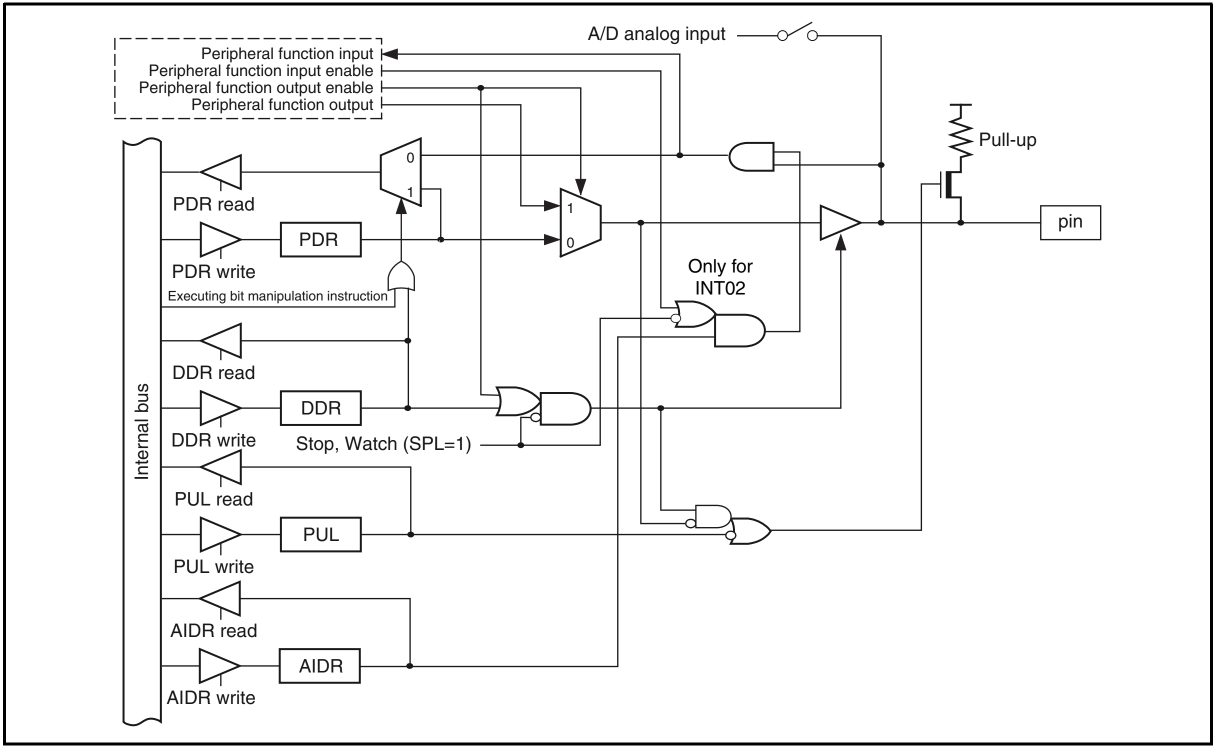


Figure 17.4-4 Block Diagram of Pin UCK (P76/CMP2_O/UCK)

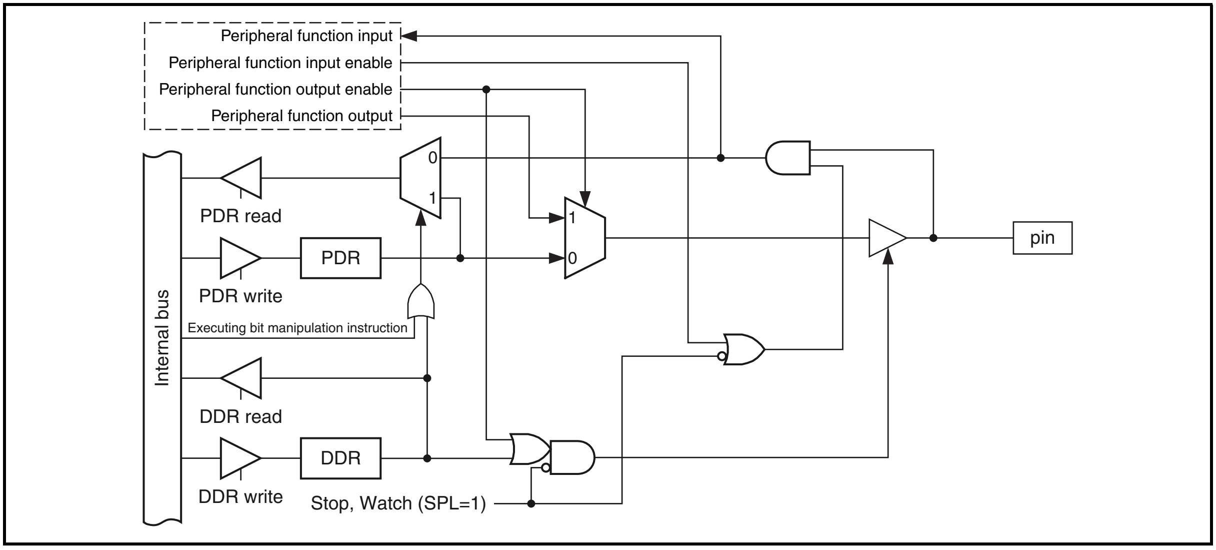

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
0
1
1
0
pin
PDR read
PDR
PDR write
Internal bus
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)

Figure 17.4-5 Block Diagram of UI (P04/INT04/AN04/UI/SCL)

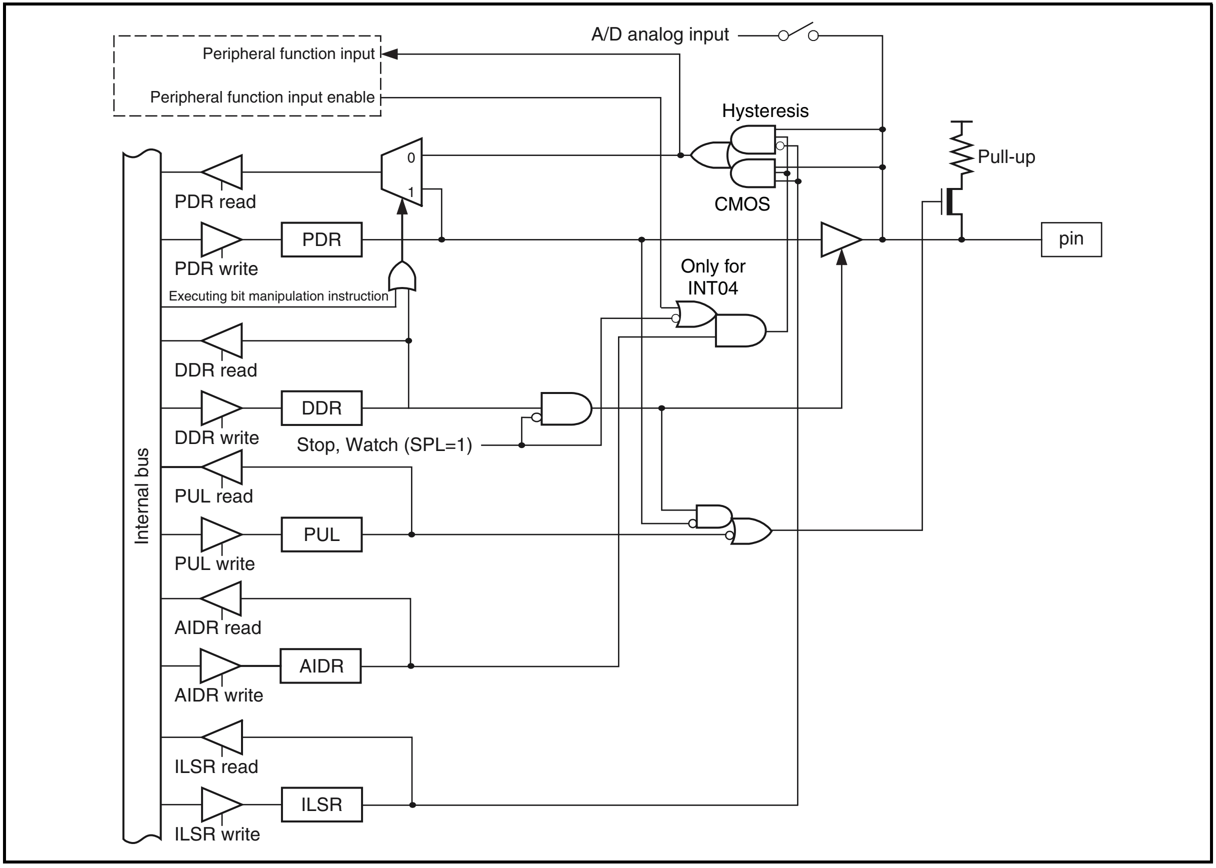
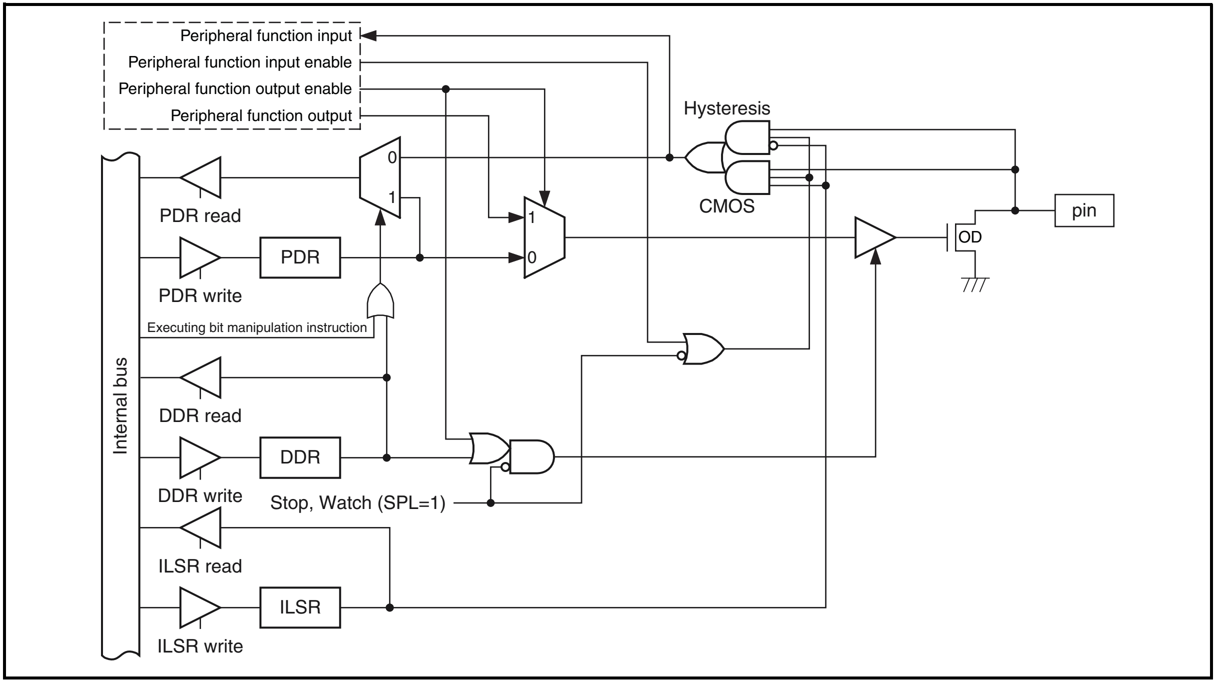


Figure 17.4-6 Block Diagram of UI (P12/EC0/UI/SCL/DBG)


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
pin
OD
0
1
PDR read
PDR
PDR write
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write
Internal bus

17.5 Registers of UART/SIO

The registers of UART/SIO are UART/SIO serial mode control register 1 (SMC10), UART/SIO serial mode control register 2 (SMC20), UART/SIO serial status and data register (SSR0), UART/SIO serial output data register (TDR0), and UART/SIO serial input data register (RDR0).

■ Registers of UART/SIO

Figure 17.5-1 Registers of UART/SIO

UART/SIO serial mode control register 1 (SMC10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0056_H	BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial mode control register 2 (SMC20)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0057_H	SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE	00100000_B
	R/W	R/W	R1/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial status and data register (SSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0058_H	-	-	PER	OVE	FER	RDRF	TCPL	TDRE	00000001_B
	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R(RM1), W	R/WX	

UART/SIO serial output data register (TDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0059_H	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UART/SIO serial input data register (RDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005A_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W	: Readable/writable (The read value is the same as the write value.)
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/WX	: Read only (Readable. Writing a value to it has no effect on operation.)
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.
R1/W	: Readable/writable (The read value is "1".)
-	: Undefined bit

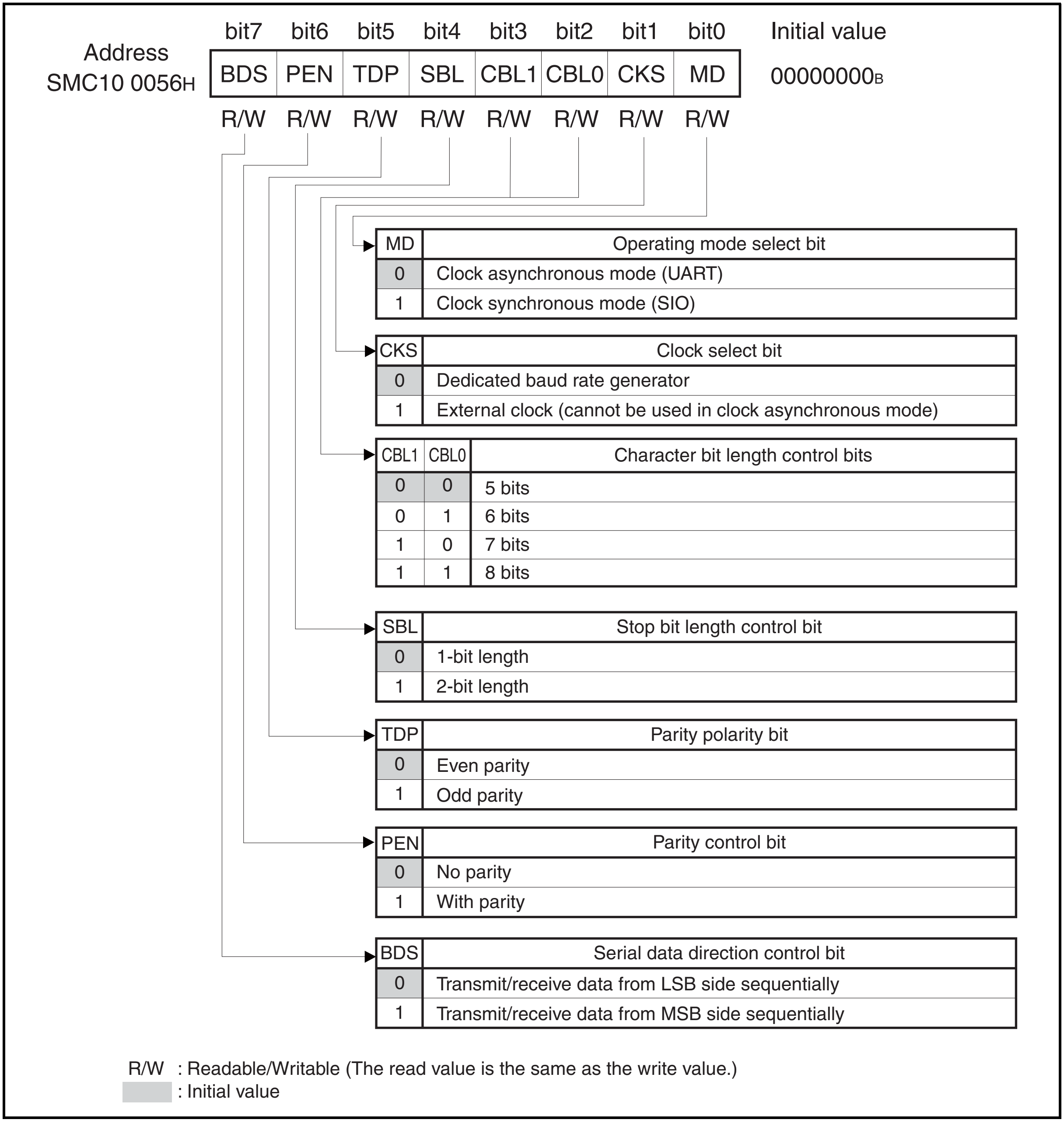
MB95430H Series

17.5.1 UART/SIO Serial Mode Control Register 1 (SMC10)

UART/SIO serial mode control register 1(SMC10) controls the UART/SIO operation mode. The register is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/ asynchronous), data length, and serial clock.

■ UART/SIO Serial Mode Control Register 1 (SMC10)

Figure 17.5-2 UART/SIO Serial Mode Control Register 1 (SMC10)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
SMC10 0056H	BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

MD	Operating mode select bit
0	Clock asynchronous mode (UART)
1	Clock synchronous mode (SIO)

CKS	Clock select bit
0	Dedicated baud rate generator
1	External clock (cannot be used in clock asynchronous mode)

CBL1	CBL0	Character bit length control bits
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

SBL	Stop bit length control bit
0	1-bit length
1	2-bit length

TDP	Parity polarity bit
0	Even parity
1	Odd parity

PEN	Parity control bit
0	No parity
1	With parity

BDS	Serial data direction control bit
0	Transmit/receive data from LSB side sequentially
1	Transmit/receive data from MSB side sequentially

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

▨ : Initial value

Table 17.5-1 Functions of Bits in UART/SIO Serial Mode Control Register 1 (SMC10)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>BDS:
Serial data direction control bit</td><td>This bit sets the serial data direction (endian).
Writing "0": The bit specifies transmission or reception to be performed sequentially starting from the LSB side in the serial data register.
Writing "1": The bit specifies transmission or reception to be performed sequentially starting from the MSB side in the serial data register.</td></tr>
<tr><td>bit6</td><td>PEN:
Parity control bit</td><td>This bit enables or disables parity in clock asynchronous mode.
Writing "0": No parity
Writing "1": With parity</td></tr>
<tr><td>bit5</td><td>TDP:
Parity polarity bit</td><td>This bit controls even/odd parity.
Writing "0": Even parity
Writing "1": Odd parity</td></tr>
<tr><td>bit4</td><td>SBL:
Stop bit length control bit</td><td>This bit controls the length of the stop bit in clock asynchronous mode.
Writing "0": Sets the stop bit length to "1".
Writing "1": Sets the stop bit length to "2".
Note: The setting of this bit is only valid for transmission operation in clock asynchronous mode.
For receiving operation, reception data register full flag is se to "1" after detecting stop bit(1-bit) and completing the reception regardless of this bit.</td></tr>
<tr><td>bit3,
bit2</td><td>CBL1, CBL0:
Character bit length control bit</td><td>These bits select the character bit length as shown in the following table:
<table>
<tr><th>CBL1</th><th>CBL0</th><th>Character bit length</th></tr>
<tr><td>0</td><td>0</td><td>5</td></tr>
<tr><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>8</td></tr>
</table>
The above setting is valid in both asynchronous and synchronous modes.</td></tr>
<tr><td>bit1</td><td>CKS:
Clock select bit</td><td>This bit selects the external clock or dedicated baud rate generator.
Writing "0": Selects the dedicated baud rate generator.
Writing "1": Selects the external clock.
Note: Setting this bit to "1" forcibly disables the output of the UCK pin.
The external clock cannot be used in clock asynchronous mode (UART).</td></tr>
<tr><td>bit0</td><td>MD:
Operation mode select bit</td><td>This bit selects clock asynchronous mode (UART) or clock synchronous mode (SIO).
Writing "0": Selects clock asynchronous mode (UART).
Writing "1": Selects clock synchronous mode (SIO).</td></tr>
</table>

Note:

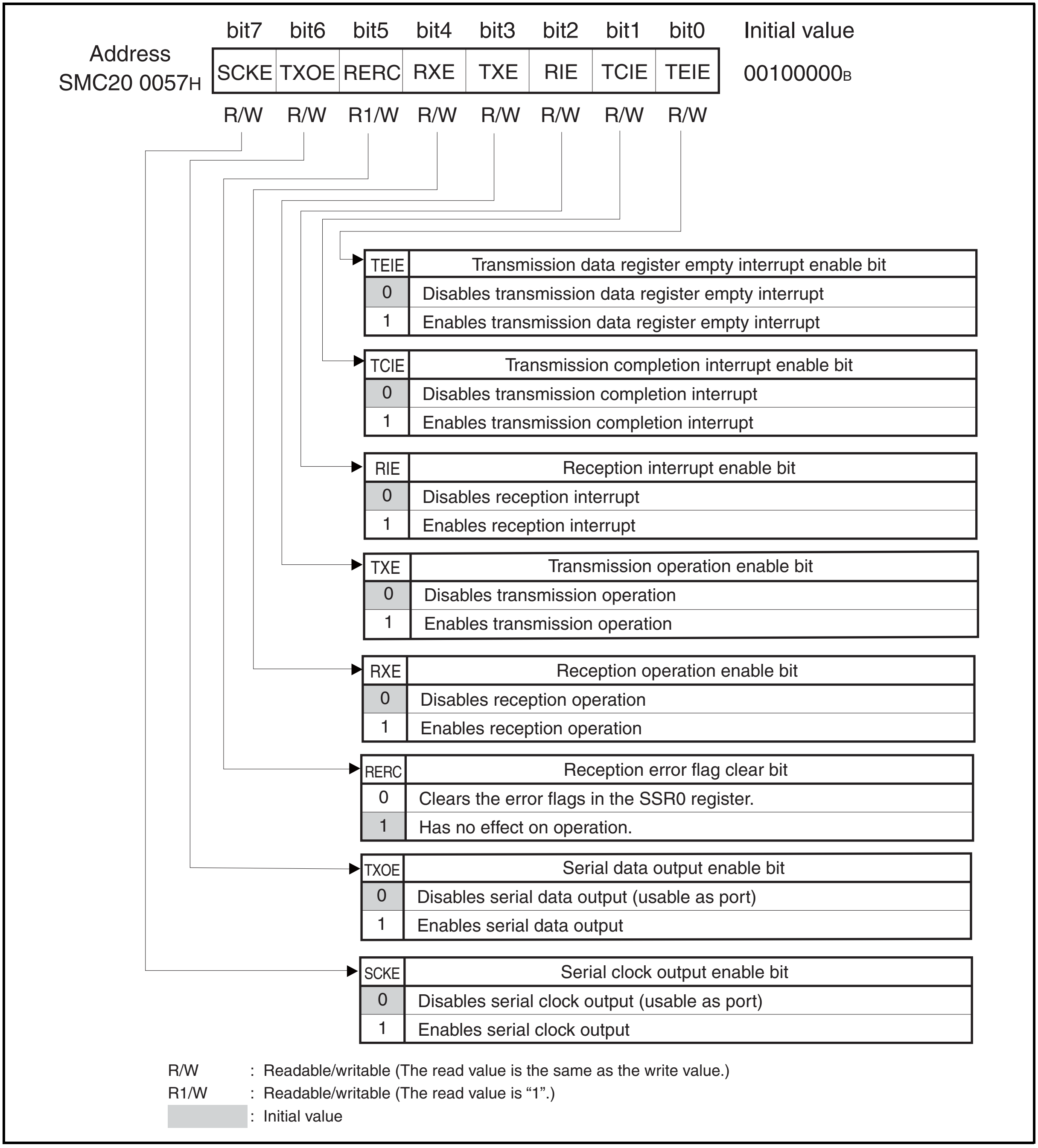
When modifying the UART/SIO serial mode control register 1 (SMC10), do not perform the modification during data transmission or reception.

17.5.2 UART/SIO Serial Mode Control Register 2 (SMC20)

UART/SIO serial mode control register 2 (SMC20) controls the UART/SIO operation mode. The register is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

■ UART/SIO Serial Mode Control Register 2 (SMC20)

Figure 17.5-3 UART/SIO Serial Mode Control Register 2 (SMC20)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
SMC20 0057H	SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE	00100000B
	R/W	R/W	R1/W	R/W	R/W	R/W	R/W	R/W	

TEIE	Transmission data register empty interrupt enable bit
0	Disables transmission data register empty interrupt
1	Enables transmission data register empty interrupt

TCIE	Transmission completion interrupt enable bit
0	Disables transmission completion interrupt
1	Enables transmission completion interrupt

RIE	Reception interrupt enable bit
0	Disables reception interrupt
1	Enables reception interrupt

TXE	Transmission operation enable bit
0	Disables transmission operation
1	Enables transmission operation

RXE	Reception operation enable bit
0	Disables reception operation
1	Enables reception operation

RERC	Reception error flag clear bit
0	Clears the error flags in the SSR0 register.
1	Has no effect on operation.

TXOE	Serial data output enable bit
0	Disables serial data output (usable as port)
1	Enables serial data output

SCKE	Serial clock output enable bit
0	Disables serial clock output (usable as port)
1	Enables serial clock output

R/W : Readable/writable (The read value is the same as the write value.)
R1/W : Readable/writable (The read value is "1".)
▨ : Initial value

Table 17.5-2 Functions of Bits in UART/SIO Serial Mode Control Register 2 (SMC20)

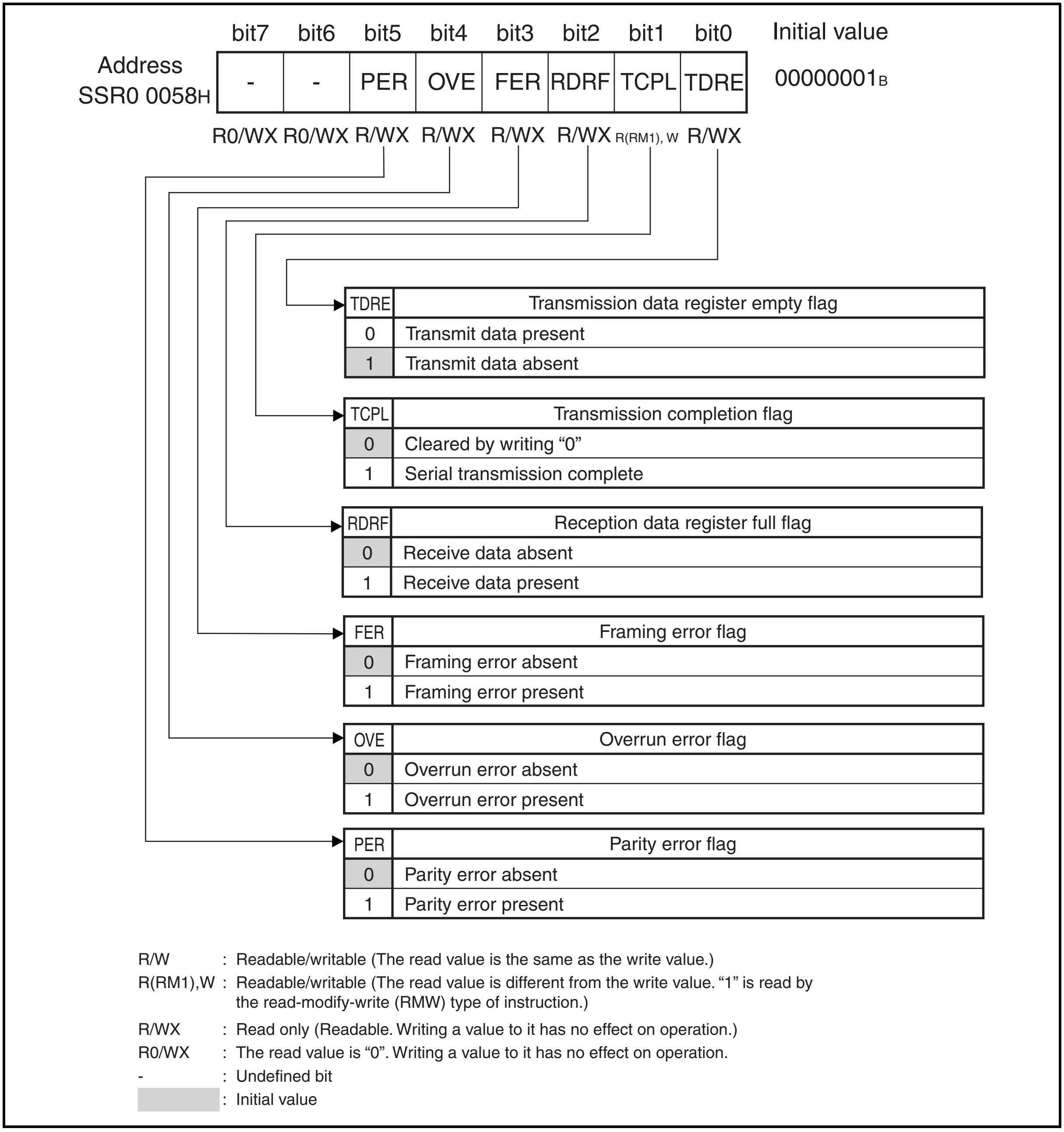
Bit name		Function
bit7	SCKE: Serial clock output enable bit	This bit controls the input/output of the serial clock (UCK) pin in clock synchronous mode. **Writing "0":** Allows the pin to be used as a general-purpose port. **Writing "1":** Enables clock output. Note: When CKS is "1", the internal clock signal is not output even with this bit set to "1". If this bit is set to "1" with SMC10:MD set to "0" (asynchronous mode), the output from the port will always be "H".
bit6	TXOE: Serial data output enable bit	This bit controls the output of the serial data (UO pin). **Writing "0":** Allows the pin to be used as a general-purpose port. **Writing "1":** Enables serial data output.
bit5	RERC: Reception error flag clear bit	**Writing "0":** Clears the error flags (PER, OVE, FER) in the SSR0 register. **Writing "1":** Has no effect on operation. This bit always returns "1" when read.
bit4	RXE: Reception operation enable bit	**Writing "0":** Disables the reception of serial data. **Writing "1":** Enables the reception of serial data. If this bit is set to "0" during reception, the reception operation will be immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the UART/SIO serial input data register. Note: Setting this bit to "0" initializes reception operation. It has no effect on the interrupt flags (PER, OVE, FRE, RDRF).
bit3	TXE: Transmission operation enable bit	**Writing "0":** Disables the transmission of serial data. **Writing "1":** Enables the transmission of serial data. If this bit is set to "0" during transmission, the transmission operation will be immediately disabled and initialization will be performed. The transmission completion flag (TCPL) will be set to "1" and the transmission data register empty (TDRE) bit will also be set to "1".
bit2	RIE: Receive interrupt enable bit	**Writing "0":** Disables receive interrupt. **Writing "1":** Enables receive interrupt. A receive interrupt occurs immediately after either the receive data register full (RDRF) bit or an error flag (PER, OVE, FER, or RDRF) is set to "1" with this bit set to "1" (enabled).
bit1	TCIE: Transmission completion interrupt enable bit	**Writing "0":** Disables interrupts by the transmission completion flag. **Writing "1":** Enables interrupts by the transmission completion flag. A transmit interrupt occurs immediately after the transmission completion flag (TCPL) bit is set to "1" with this bit set to "1" (enabled).
bit0	TEIE: Transmission data register empty interrupt enable bit	**Writing "0":** Disables interrupts by the transmission data register empty. **Writing "1":** Enables interrupts by the transmission data register empty. A transmit interrupt occurs immediately after the transmission data register empty (TDRE) bit is set to "1" with this bit set to "1" (enabled).

17.5.3 UART/SIO Serial Status and Data Register (SSR0)

The UART/SIO serial status and data register (SSR0) indicates the transmission/reception status and error status of the UART/SIO.

■ UART/SIO Serial Status and Data Register (SSR0)

Figure 17.5-4 UART/SIO Serial Status and Data Register (SSR0)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address SSR0 0058_H	-	-	PER	OVE	FER	RDRF	TCPL	TDRE	00000001_B
	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R(RM1), W	R/WX	

TDRE	Transmission data register empty flag
0	Transmit data present
▨ 1	Transmit data absent

TCPL	Transmission completion flag
▨ 0	Cleared by writing “0”
1	Serial transmission complete

RDRF	Reception data register full flag
▨ 0	Receive data absent
1	Receive data present

FER	Framing error flag
▨ 0	Framing error absent
1	Framing error present

OVE	Overrun error flag
▨ 0	Overrun error absent
1	Overrun error present

PER	Parity error flag
▨ 0	Parity error absent
1	Parity error present

R/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.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 17.5-3 Functions of Bits in UART/SIO Serial Status and Data Register (SSR0)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5	PER: Parity error flag	This flag detects a parity error in receive data. • The bit is set when a parity error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit4	OVE: Overrun error flag	This flag detects an overrun error in receive data. • The flag is set when an overrun error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit3	FER: Framing error flag	This flag detects a framing error in receive data. • The bit is set when a framing error occurs during reception. Writing "0" to the RERC bit clears this flag. • If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.
bit2	RDRF: Receive data register full flag	This flag indicates the status of the UART/SIO serial input data register. • The bit is set to "1" when receive data is loaded to the serial input data register. • The bit is cleared to "0" when data is read from the serial input data register.
bit1	TCPL: Transmission completion flag	This flag indicates the data transmission status. • The bit is set to "1" upon completion of serial transmission. Note, however, that the bit is not set to "1" even upon completion of transmission when the UART/SIO serial output data register contains data to be transmitted in succession. • Writing "0" to this bit clears its flag. • If events to set and clear the flag occur at the same time, it is set preferentially. • Writing "1" to this bit has no effect on operation.
bit0	TDRE: Transmission data register empty flag	This flag indicates the status of the UART/SIO serial output data register. • The bit is set to "0" when transmit data is written to the serial output register. • The bit is set to "1" when data is loaded to the transmission shift register and transmission starts.

17.5.4 UART/SIO Serial Input Data Register (RDR0)

The UART/SIO serial input data register (RDR0) is used to input (receive) serial data.

■ UART/SIO Serial Input Data Register (RDR0)

Figure 17.5-5 shows the bit configuration of the UART/SIO serial input data register.

Figure 17.5-5 UART/SIO Serial Input Data Register (RDR0)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RDR0	$005A_H$	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000_B
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)

This register stores received data. The serial data signals sent to the serial data input pin (UI pin) is converted by the shift register and stored in this register.

When received data is set correctly in this register, the receive data register full (RDRF) bit is set to "1". At this time, an interrupt occurs if receive interrupt requests have been enabled. If an RDRF bit check by the program or using an interruption shows that received data is stored in this register, the reading of the content for this register clears the RDRF flag to "0".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are set to "0".

17.5.5 UART/SIO Serial Output Data Register (TDR0)

The UART/SIO serial output data register (TDR0) is used to output (transmit) serial data.

■ UART/SIO Serial Output Data Register (TDR0)

Figure 17.5-6 shows the bit configuration of the UART/SIO serial output data register.

Figure 17.5-6 UART/SIO Serial Output Data Register (TDR0)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TDR0	0059_H	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

This register holds data to be transmitted. The register accepts a write when the transmission data register empty (TDRE) bit contains "1". An attempt to write to the bit is ignored when the bit contains "0".

When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of the UART/SIO serial mode control register is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible. Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR0, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". The transmission data is transferred to the shift register for the transmission, it is converted into the serial data, and it is transmitted from the serial data output pin.

When transmit data is written to the UART/SIO serial output data register (TDR0), the transmission data register empty bit (TDRE) is set to "0". Upon completion of transfer of transmit data to the transmission shift register, the transmission data register empty bit (TDRE) is set to "1", allowing the next piece of transmit data to be written. At this time, an interrupt occurs if transmission data register empty interrupts have been enabled. Write the next piece of transmit data when transmit data empty occurs or the transmit data empty (TDRE) bit is set to "1".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are ignored.

Note:

The data in this register cannot be updated when TDRE in UART/SIO serial status data register is "0".
When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of the UART/SIO serial mode control register 2 is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible.
Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". To change data, write it after making TDRE "1" once by writing "0" to TXE.

17.6 Interrupts of UART/SIO

The UART/SIO has six interrupt-related bits: error flag bits (PER, OVE, FER), receive data register full bit (RDRF), transmission data register empty bit (TDRE), and transmission completion flag (TCPL).

■ Interrupts of UART/SIO

Table 17.6-1 lists the UART/SIO interrupt control bits and interrupt sources.

Table 17.6-1 UART/SIO Interrupt Control Bits and Interrupt Sources

Item	Description					
Interrupt request flag bit	SSR0: TDRE	SSR0: TCPL	SSR0: RDRF	SSR0: PER	SSR0: OVE	SSR0: FER
Interrupt request enable bit	SMC20: TEIE	SMC20: TCIE	SMC20: RIE	SMC20: RIE	SMC20: RIE	SMC20: RIE
Interrupt source	Transmission data register empty	Transmission completion	Receive data full	Parity error	Overrun error	Framing error

■ Transmit Interrupt

When transmit data is written to the UART/SIO serial output data register (TDR0), the data is transferred to the transmission shift register. When the next piece of data can be written, the TDRE bit is set to "1". At this time, an interrupt request to the interrupt controller occurs when transmit data register empty interrupt enable bit has been enabled (SMC20:TEIE = 1).

The TCPL bit is set to "1" upon completion of transmission of all pieces of transmit data. At this time, an interrupt request to the interrupt controller occurs when transmission completion interrupt enable bit has been enabled (SMC20:TCIE = 1).

■ Receive Interrupt

If the data is input successfully up to the stop bit, the RDRF bit is set to 1. If an overrun, parity, or framing error occurs, the corresponding error flag bit (PER, OVE, or FER) is set to "1".

These bits are set when a stop bit is detected. If receive interrupt enable bit has been enabled (SMC20:RIE = 1), an interrupt request to the interrupt controller will be generated.

■ Register and Vector Table Addresses Related to UART/SIO Interrupts

Table 17.6-2 Register and Vector Table Addresses Related to UART/SIO Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
UART/SIO ch. 0	IRQ04	ILR1	L04	$FFF2_H$	$FFF3_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

17.7 Operations of UART/SIO Operations and Setting Procedure Example

The UART/SIO has a serial communication function (operation mode 0, 1).

■ Operations of UART/SIO

● Operation mode

Two operation modes are available in the UART/SIO. Clock synchronous mode (SIO) or clock asynchronous mode (UART) can be selected (See Table 17.7-1).

Table 17.7-1 Operation Modes of UART/SIO

Operation mode	Data length		Synchronization mode	Length of stop bit
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	-	Synchronous	-
	6	-		
	7	-		
	8	-		

■ Setting Procedure Example

Below is an example of procedure for setting the UART/SIO.

● Initial setup

1) Set the port input. (DDR1)
2) Set the interrupt level. (ILR1)
3) Set the prescaler. (PSSR0)
4) Set the baud rate. (BRSR0)
5) Select the clock. (SMC10:CKS)
6) Set the operation mode. (SMC10:MD)
7) Enable/disable the serial clock output. (SMC20:SCKE)
8) Enable reception. (SMC20:RXE = 1)
9) Enable interrupts. (SMC20:RIE = 1)

● Interrupt processing

Read receive data. (RDR0)

17.7.1 Operations in Operation Mode 0

Operation mode 0 operates as clock asynchronous mode (UART).

■ Operations in UART/SIO Operation Mode 0

Clock asynchronous mode (UART) is selected when the MD bit in the UART/SIO serial mode control register 1 (SMC10) is set to "0".

● Baud rate

The serial clock is selected by the CKS bit in the SMC10 register. Be sure to select the dedicated baud rate generator at this time.

The baud rate is equivalent to the output clock frequency of the dedicated baud rate generator, divided by four. The UART can perform communication within the range from -2% to +2% of the selected baud rate.

The baud rate generated by the dedicated baud rate generator is obtained from the equation illustrated below. (For information about the dedicated baud rate generator, see "CHAPTER 18 UART/SIO DEDICATED BAUD RATE GENERATOR".)

Figure 17.7-1 Baud Rate Calculation when Using Dedicated Baud Rate Generator

$$\text{Baud rate value} = \frac{\text{Machine clock (MCLK)}}{4\times \boxed{\begin{matrix}1\\2\\4\\8\end{matrix}} \times \boxed{\begin{matrix}2\\:\\255\end{matrix}}}\ \text{[bps]}$$

UART prescaler select register (PSSR0)
Prescaler select (PSS1, PSS0)

UART baud rate setting register (BRSR0)
Baud rate setting (BRS7 to BRS0)

Table 17.7-2 Sample Asynchronous Transfer Rates Based on Dedicated Baud Rate Generator (Clock Gear = 4/F_{CH}, Machine Clock = 10MHz, 16MHz, 16.25MHz)

Dedicated baud rate generator setting		Internal UART division	Total division ratio (PSS × BRS × 4)	Baud rate (10MHz / Total division ratio)	Baud rate (16MHz / Total division ratio)	Baud rate (16.25MHz / Total division ratio)
Prescaler select PSS[1:0]	Baud rate counter setting BRS[7:0]					
1 (Setting value: 0,0)	20	4	80	125000	200000	203125
1 (Setting value: 0,0)	22	4	88	113636	181818	184659
1 (Setting value: 0,0)	44	4	176	56818	90909	92330
1 (Setting value: 0,0)	87	4	348	28736	45977	46695
1 (Setting value: 0,0)	130	4	520	19231	30769	31250
2 (Setting value: 0,1)	130	4	1040	9615	15385	15625
4 (Setting value: 1,0)	130	4	2080	4808	7692	7813
8 (Setting value: 1,1)	130	4	4160	2404	3846	3906

The baud rate in clock asynchronous mode can be set in the following range.

Table 17.7-3 Baud Rate Setting Range in Clock Asynchronous Mode

PSS[1:0]	BRS[7:0]
"00_B" to "11_B"	02_H (2) to FF_H (255)

● Transfer data format

UART can treat data only in NRZ (Non-Return-to-Zero) format. Figure 17.7-2 shows the data format.

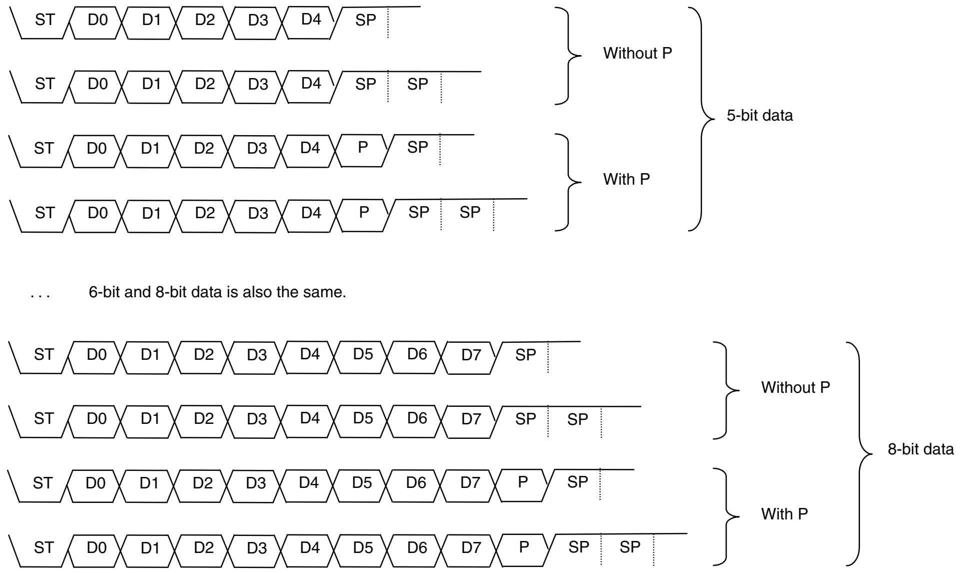
The character bit length can be selected from among 5 to 8 bits depending on the CBL1 and CBL0 settings.

The stop bit length can be set to 1 or 2 bits depending on the SBL setting.

PEN and TDP can be used to enable/disable parity and to select parity polarity.

As shown in Figure 17.7-2, the transfer data always starts from the start bit ("L" level) and ends with the stop bit ("H" level) by performing the specified data bit length transfer with MSB first or LSB first ("LSB first" or "MSB first" can be selected by the BDS bit). It becomes "H" level at the idle state.

Figure 17.7-2 Transfer Data Format


ST D0 D1 D2 D3 D4 SP
ST D0 D1 D2 D3 D4 SP SP
Without P
ST D0 D1 D2 D3 D4 P SP
ST D0 D1 D2 D3 D4 P SP SP
With P
5-bit data
. . . 6-bit and 8-bit data is also the same.
ST D0 D1 D2 D3 D4 D5 D6 D7 SP
ST D0 D1 D2 D3 D4 D5 D6 D7 SP SP
Without P
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP SP
With P
8-bit data

ST : Start bit
SP : Stop bit
P : Parity bit
D0 to D7:Data. The sequence can be selected from "LSB first" or "MSB first" by the direction control register (BDS bit)

● Receiving operation in asynchronous clock mode (UART)

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

Reception remains performed as long as the reception operation enable bit (RXE) contains "1".

Upon detection of a start bit in receive data with the reception operation enable bit (RXE) set to "1", one frame of data is received according to the data format set in UART/SIO serial control register 1 (SMC10).

When the reception of one frame of data has been completed, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next frame of serial data can be received.

When the UART/SIO serial input data register (RDR0) stores data, the receive data register full (RDRF) bit is set to "1".

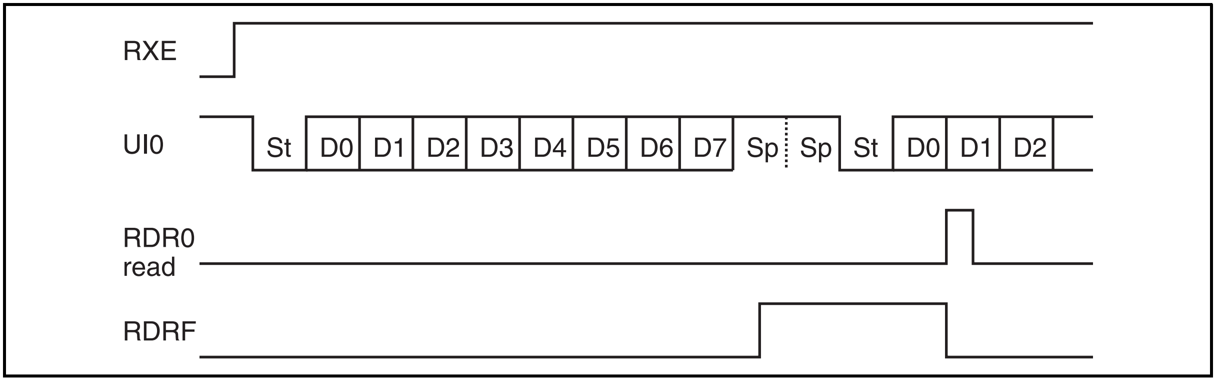
A receive interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the receive interrupt enable bit (RIE) contains "1".

Received data is read from the UART/SIO serial input data register (RDR0) after each error flag (PER, OVE, FER) in the UART/SIO serial status and data register is checked.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

Note that modifying UART/SIO serial mode control register 1 (SMC10) during reception may result in unpredictable operation. If the RXE bit is set to "0" during reception, the reception is immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the serial input data register.

Figure 17.7-3 Receiving Operation in Asynchronous Clock Mode



● Reception error in asynchronous clock mode (UART)

If any of the following three error flags (PER, FER, OVE) has been set, receive data is not transferred to the UART/SIO serial input data register (RDR0) and the receive data register full (RDRF) bit is not set to "1" either.

- Parity error (PER)

  The parity error (PER) bit is set to "1" if the parity bit in received serial data does not match the parity polarity bit (TDP) when the parity control bit (PEN) contains "1".

- Framing error (FER)

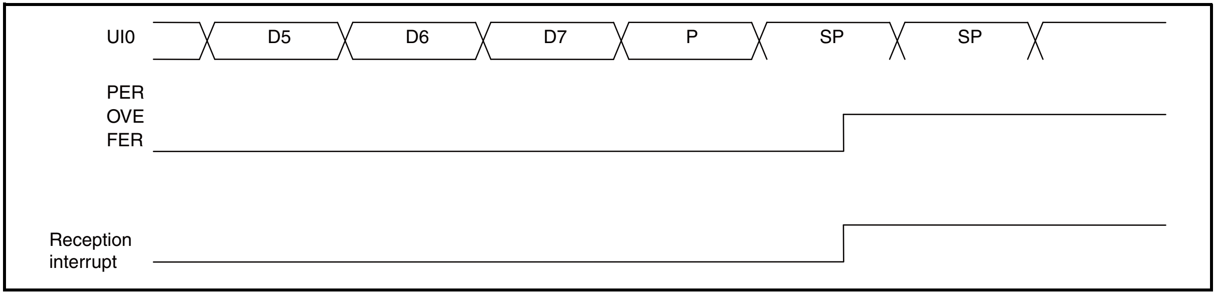
  The framing error (FER) bit is set to "1" if "1" is not detected at the position of the first stop bit in serial data received in the set character bit length (CBL) under parity control (PEN). Note that the stop bit is not checked if it appears at the second bit or later.

- Overrun error (OVE)

  Upon completion of reception of serial data, the overrun error (OVE) bit is set to "1" if the reception of the next data is performed before the previous receive data is read.

Each flag is set at the position of the first stop bit.

Figure 17.7-4 Setting Timing for Receiving Errors



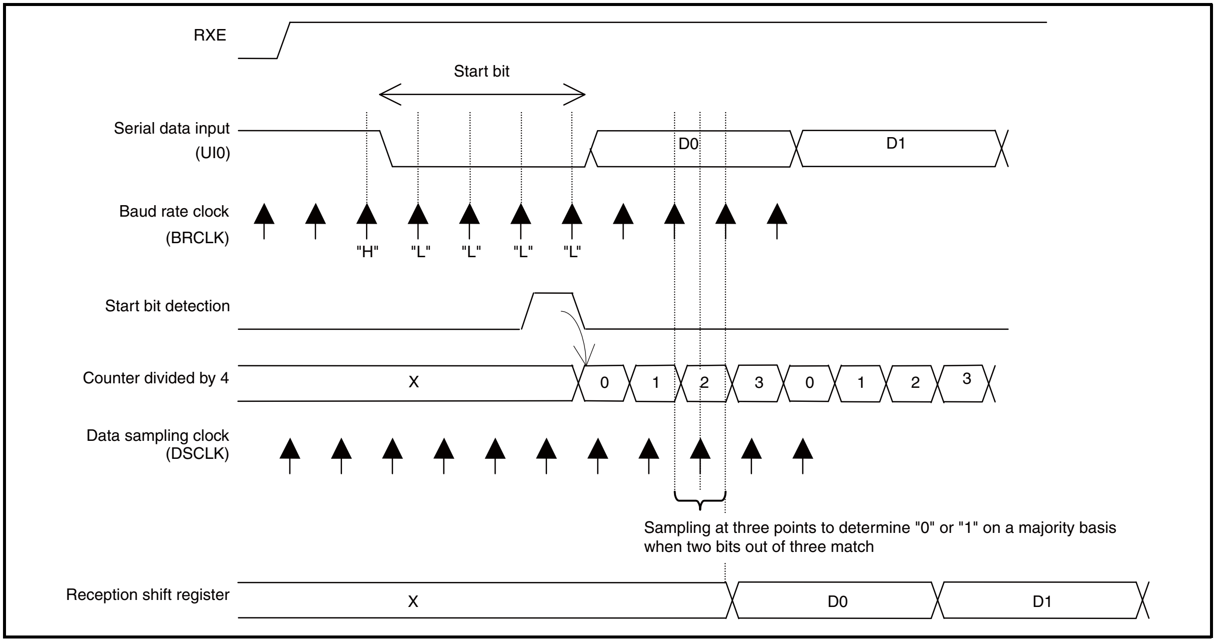
● Start bit detection and confirmation of receive data during reception

The start bit is detected by a falling of the serial input followed by a succession of three "L" levels after the serial data input is sampled according to the clock (BRCLK) signal provided by the dedicated baud rate generator with the reception operation enable bit (RXE) set to "1". When the first "H", "L", "L", "L" train is detected in a BRCLK sample, therefore, the current bit is regarded as the start bit.

The frequency-quartered circuit is activated upon detection of the start bit and serial data is input to the reception shift register at intervals of four periods of BRCLK.

When data is received, sampling is performed at three points of the baud rate clock (BRCLK) and data sampling clock (DSCLK) and received data is confirmed on a majority basis when two bits out of three match.

Figure 17.7-5 Start Bit Detection and Serial Data Input



● Transmission in asynchronous clock mode

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

Either of the following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", and then write transmit data to the serial output data register to start transmission.
- Write transmit data to the UART/SIO serial output data register, and then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

The transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, and the transmit data register empty (TDRE) is set to "1".

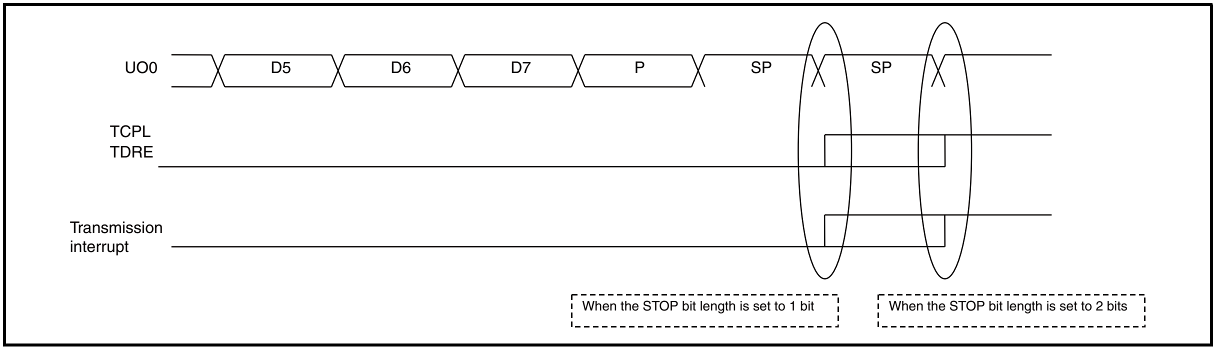
When the transmit interrupt enable bit (TIE) contains "1", a transmit interrupt occurs if the transmit data register empty (TDRE) bit is set to "1". This allows the next piece of transmit data to be written to the UART/SIO serial output data register (TDR0) by interrupt handling.

To detect the completion of serial transmission by transmit interrupt, set the transmission completion interrupt enable bits as follows: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmit interrupt occurs.

Both the transmission completion flag (TCPL) and the transmission data register empty flag (TDRE), when transmitting data consecutively, are set at the position which the transmission of the last bit was completed (it varies depending on the data length, parity enable, or stop bit length setting), as shown in Figure 17.7-6 below.

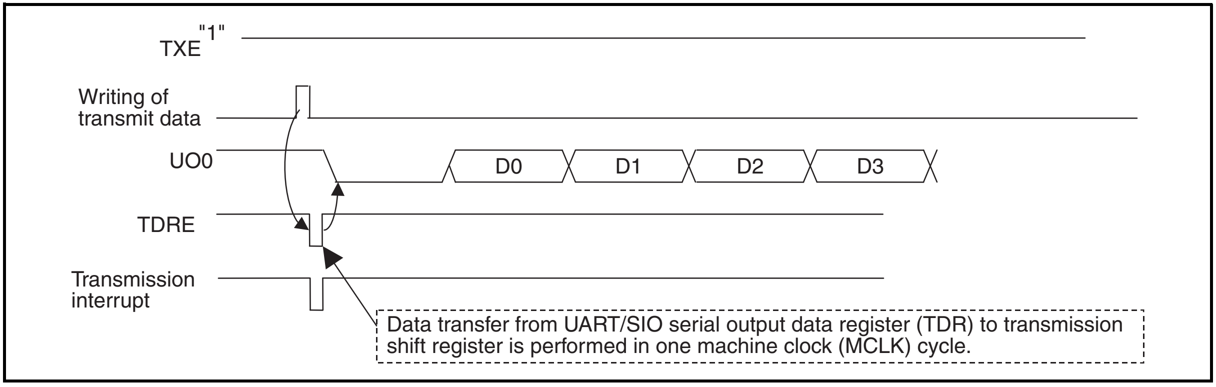
Note that modifying UART/SIO serial mode control register 1 (SMC10) during transmission may result in unpredictable operation.

Figure 17.7-6 Transmission in Asynchronous Clock Mode (UART)


UO0
D5
D6
D7
P
SP
SP
TCPL
TDRE
Transmission interrupt
When the STOP bit length is set to 1 bit
When the STOP bit length is set to 2 bits

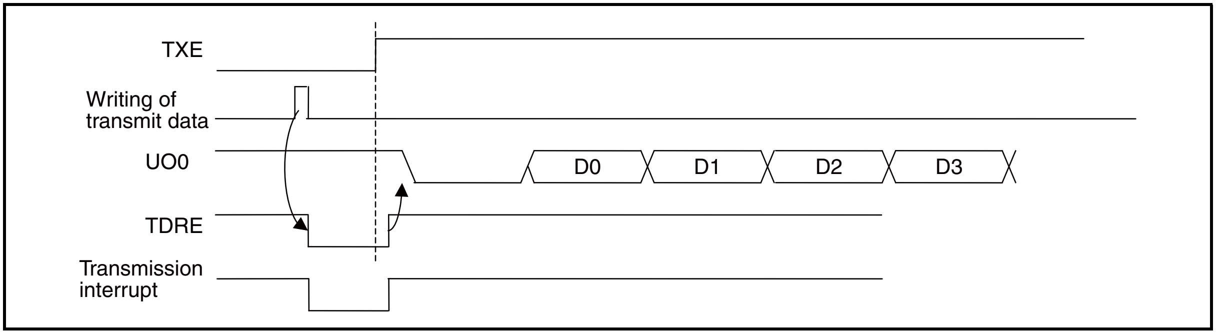
The TDRE flag is set at the point indicated in the following figure if the preceding piece of transmit data does not exist in the transmission shift register.

Figure 17.7-7 Setting Timing 1 for Transmit Data Register Empty Flag (TDRE) (When TXE is "1")



Data transfer from UART/SIO serial output data register (TDR) to transmission shift register is performed in one machine clock (MCLK) cycle.

Figure 17.7-8 Setting Timing 2 for Transmit Data Register Empty Flag (TDRE) (When TXE Is Switched from "0" to "1")



● Concurrent transmission and reception

In asynchronous clock mode (UART), transmission and reception can be performed independently. Therefore, transmission and reception can be performed at the same time or even with transmitting and receiving frames overlapping each other in shifted phases.

17.7.2 Operations in Operation Mode 1

Operation mode 1 operates in synchronous clock mode.

■ Operations in UART/SIO Operation Mode 1

Setting the MD bit in UART/SIO serial mode control register 1 (SMC10) to "1" selects synchronous clock mode (SIO).

The character bit length in synchronous clock mode (SIO) is variable between 5 bits and 8 bits.

Note, however, that parity is disabled and no stop bit is used.

The serial clock is selected by the CKS bit in the SMC10 register. Select the dedicated baud rate generator or external clock. The SIO performs shift operation using the selected serial clock as a shift clock.

To input the external clock signal, set the SCKE bit to "0".

To output the dedicated baud rate generator output as a shift clock signal, set the SCKE bit to "1". The serial clock signal is obtained by dividing clock by two, which is supplied by the dedicated baud rate generator. The baud rate in the SIO mode can be set in the following range. (For more information about the dedicated baud rate generator, also see "CHAPTER 18 UART/SIO DEDICATED BAUD RATE GENERATOR")

Table 17.7-4 Baud Rate Setting Range in SIO Mode

PSS[1:0]	BRS[7:0]
00_B to 11_B	01_H(1) to FF_H(255), 00_H(256) (The highest and lowest baud rate settings are 01_H and 00_H, respectively.)

The baud rate applied when the external clock or dedicated baud rate generator is used is obtained from the corresponding equation illustrated below.

Figure 17.7-9 Calculating Baud Rate Based on External Clock

$$\text{Baud rate value} = \frac{1}{\text{External clock*}} \text{ [bps]}$$

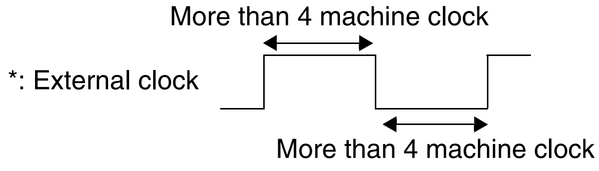


Figure 17.7-10 Baud Rate Calculation Formula for Using Dedicated Baud Rate Generator

$$\text{Baud rate value} = \frac{\text{Machine clock (MCLK)}}{2\times \boxed{\begin{array}{c}1\\2\\4\\8\end{array}} \times \boxed{\begin{array}{c}1\\:\\256\end{array}}}\ \text{[bps]}$$

UART prescaler select register (PSSR0)
Prescaler select (PSS1, PSS0)

UART baud rate setting register (BRSR0)
Baud rate setting (BRS7 to BRS0)

- Serial clock

The serial clock signal is output under control of the output for transmit data. When only reception is performed, therefore, set transmission control (TXE = 1) to write dummy transmit data to the UART/SIO serial output register. Refer to the data sheet of the MB95430H Series for the UCK clock value.

- Reception in UART/SIO operation mode 1

For reception in operation mode 1, each register is used as follows.

Figure 17.7-11 Registers Used for Reception in Operation Mode 1

SMC10 (UART/SIO serial mode control register 1)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
◎	×	×	×	◎	◎	◎	1

SMC20 (UART/SIO serial mode control register 2)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
◎	0	◎	◎	◎	◎	×	×

SSR0 (UART/SIO serial status and data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	PER	OVE	FER	RDRF	TCPL	TDRE
×	×	×	◎	×	◎	×	×

TDR0 (UART/SIO serial output data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
×	×	×	×	×	×	×	×

RDR0 (UART/SIO serial input data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
◎	◎	◎	◎	◎	◎	◎	◎

◎ : Used bit
x : Unused bit
1 : Set to "1"
0 : Set to "0"

The reception depends on whether the serial clock has been set to external or internal clock.

<When external clock is enabled>

When the reception operation enable bit (RXE) contains "1", serial data is received always at the rising edge of the external clock signal.

<When internal clock is enabled>

The serial clock signal is output in accordance with transmission. Therefore, transmission must be performed even when only performing reception. The following two procedures can be used.

- Set the transmission operation enable bit (TXE) to "1", then write transmit data to the UART/SIO serial output data register to generate the serial clock signal and start reception.
- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to generate the serial clock signal and start reception.

When 5-bit to 8-bit serial data is received by the reception shift register, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next piece of serial data can be received.

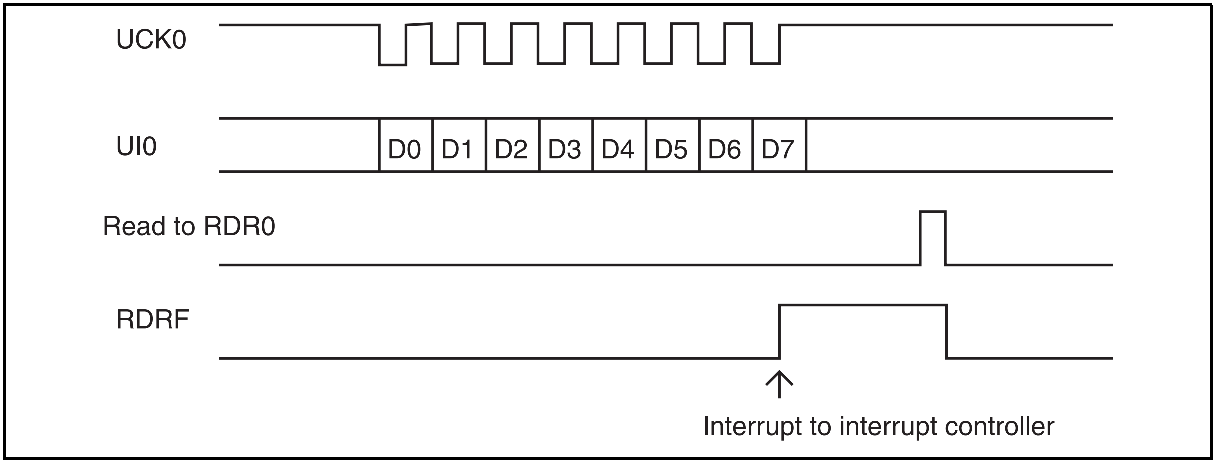
When the serial input data register stores data, the receive data register full (RDRF) bit is set to "1".

A receive interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the receive interrupt enable bit (RIE) contains "1".

To read received data, read it from the UART/SIO serial input data register after checking the error flag (OVE) in the UART/SIO serial status and data register.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

Figure 17.7-12 8-bit Reception of Synchronous CLK Mode



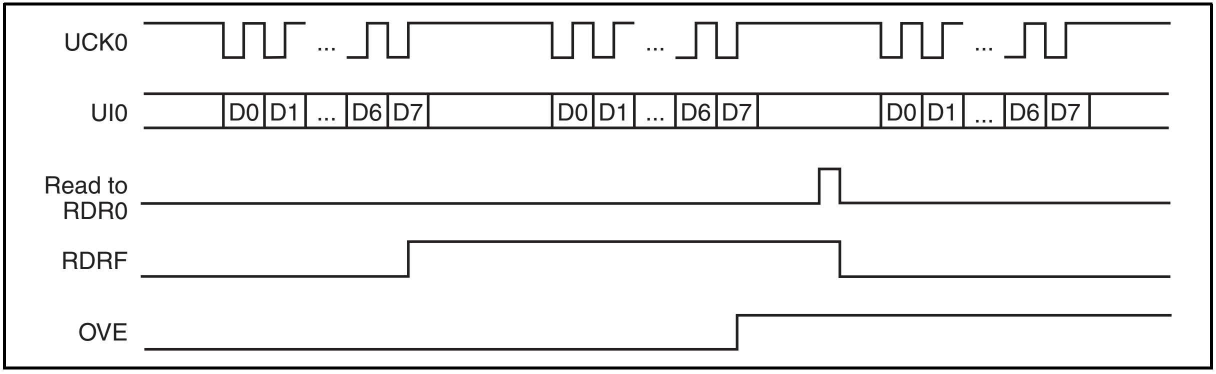
Operation when reception error occurs

When an overrun error (OVE) exists, received data is not transferred to the UART/SIO serial input data register (RDR0).

Overrun error (OVE)

Upon completion of reception for serial data, the overrun error (OVE) bit is set to "1" if the receive data register full (RDRF) bit has been set to "1" by the reception for the preceding piece of data.

Figure 17.7-13 Overrun Error



● Transmission in UART/SIO operation mode 1

For transmission in operation mode 1, each register is used as follows.

Figure 17.7-14 Registers Used for Transmission in Operation Mode 1

SMC10 (UART/SIO serial mode control register 1)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
◎	×	×	×	◎	◎	◎	1

SMC20 (UART/SIO serial mode control register 2)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
◎	0	◎	◎	◎	◎	×	×

SSR0 (UART/SIO serial status and data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	PER	OVE	FER	RDRF	TCPL	TDRE
×	×	×	◎	×	◎	×	×

TDR0 (UART/SIO serial output data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
×	×	×	×	×	×	×	×

RDR0 (UART/SIO serial input data register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
◎	◎	◎	◎	◎	◎	◎	◎

◎ : Used bit
x : Unused bit
1 : Set to "1"
0 : Set to "0"

The following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", then write transmit data to the UART/SIO serial output data register to start transmission.
- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit is set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

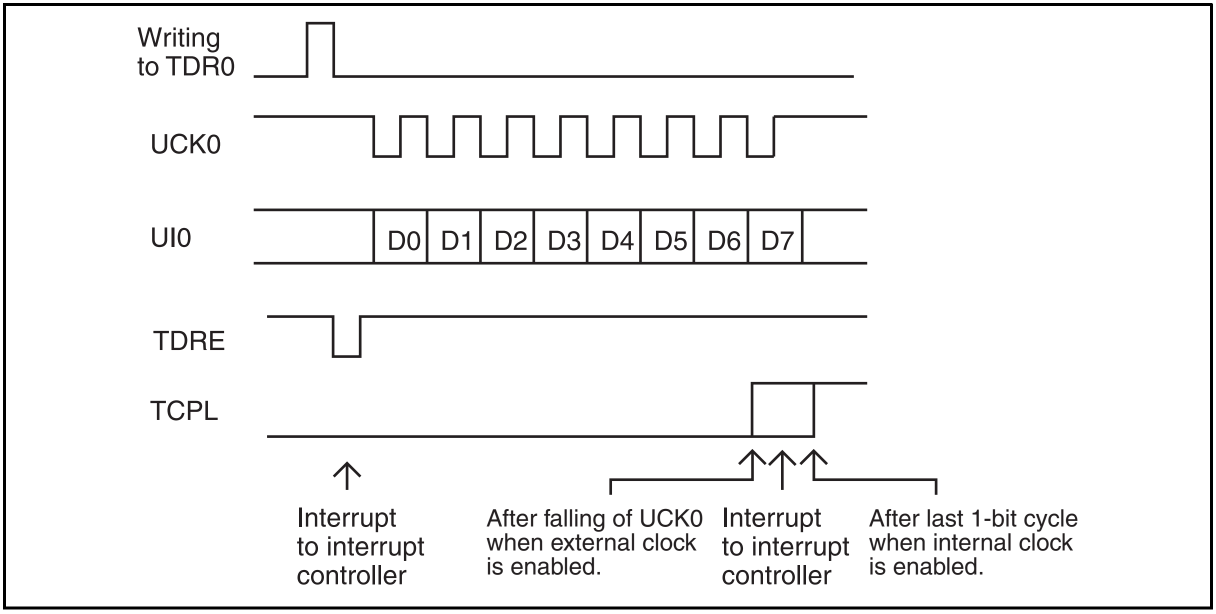
When serial transmission is started after transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, the transmit data register empty (TDRE) bit is set to "1".

When the use of the external clock signal has been set, serial data transmission starts at the fall of the first serial clock signal after the transmission process is started.

A transmission completion interrupt occurs the moment the transmit data register empty (TDRE) bit is set to "1" when the transmit interrupt enable bit (TIE) contains "1". At this time, the next piece of transmit data can be written to the UART/SIO serial output data register (TDR0). Serial transmission can be continued with the transmission operation enable bit (TXE) set to "1".

To use a transmission completion interrupt to detect the completion of serial transmission, enable transmission completion interrupt output this way: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmission completion interrupt occurs.

Figure 17.7-15 8-bit Transmission in Synchronous CLK Mode


Writing to TDR0
UCK0
UI0
D0
D1
D2
D3
D4
D5
D6
D7
TDRE
TCPL
Interrupt to interrupt controller
After falling of UCK0 when external clock is enabled.
Interrupt to interrupt controller
After last 1-bit cycle when internal clock is enabled.

● Concurrent transmission and reception

<When external clock is enabled>

Transmission and reception can be performed independently of each other. Transmission and reception can therefore be performed at the same time or even when their phases are shifted from each other and overlapping.

<When internal clock is enabled>

As the transmitting side generates a serial clock, reception is influenced.

If transmission stops during reception, the receiving side is suspended. It resumes reception when the transmitting side is restarted.

- See "17.4 Pins of UART/SIO" for operation with serial clock output and operation with serial clock input.

17.8 Sample Settings for UART/SIO

This section provides sample settings for the UART/SIO.

■ Sample Settings

● How to select the operation mode

The operation mode select bit (SMC10:MD) is used.

Operation mode		Operation mode select (MD)
Mode 0	Asynchronous clock mode (UART)	Set the bit to "0"
Mode 1	Synchronous clock mode (SIO)	Set the bit to "1"

● Operating clock types and selection method

The clock select bit (SMC10:CKS) is used.

Clock input	Clock select (CKS)
To select the dedicated baud rate generator	Set the bit to "0"
To select an external clock	Set the bit to "1"

● How to use the UCK, UI, or UO pin

The following setting is used.

	UART
To set UCK pin as an input	DDR0:P02 = 0 SMC20:SCKE = 0
To set UCK pin as an output	SMC20:SCKE = 1
To use UI pin	DDR0:P04 = 0
To use UOUI pin	SMC20:TXOE = 1

- How to enable/stop UART operation

The reception operation enable bit (SMC20:RXE) is used.

Operation	Reception operation enable bit (RXE)
To disable (stop) reception	Set the bit to "0".
To enable reception	Set the bit to "1".

The transmission operation control bit (SMC20:TXE) is used.

Operation	Transmission operation control bit (TXE)
To disable (stop) transmission	Set the bit to "0".
To enable transmission	Set the bit to "1".

- How to set parity

The parity control (SMC10:PEN) and parity polarity (SMC10:TDP) bits are used.

Operation	Parity control (PEN)	Parity polarity (TDP)
To select no parity	Set the bit to "0".	-
To select even parity	Set the bit to "1".	Set the bit to "0".
To select odd parity	Set the bit to "1".	Set the bit to "1".

- How to set the data length

The data length select bit (SMC10:CBL[1:0]) is used.

Operation	Data length select bit (CBL[1:0])
To select 5-bit length	Set the bits to "00_B".
To select 6-bit length	Set the bits to "01_B".
To select 7-bit length	Set the bits to "10_B".
To select 8-bit length	Set the bits to "11_B".

- How to select the STOP bit length

The STOP bit length control bit (SMC10:SBL) is used.

Operation	STOP bit length control (SBL)
To set the STOP bit to 1-bit length	Set SBL to "0".
To set the STOP bit to 2-bit length	Set SBL to "1".

● How to clear error flags

The reception error flag clear bit (SMC20:RERC) is used.

Operation	Reception error flag clear bit (RERC)
To clear an error flag (PER, OVE, FER)	Set the bit to "0".

● How to set the transfer direction

The serial data direction control bit (SMC10:BDS) is used.

LSB first or MSB first can be selected for the transfer direction in any operation mode.

Operation	Serial data direction control (BDS)
To select LSB first transfer (from least significant bit)	Set the bit to "0".
To select MSB first transfer (from most significant bit)	Set the bit to "1".

● How to clear the reception completion flag

The following setup is performed.

Operation	Method
To clear the reception completion flag	Read from the RDR0 register.

When the first read from the RDR0 register is performed, reception starts.

● How to clear the transmission buffer empty flag

The following setup is performed.

Operation	Method
To clear the transmission buffer empty flag	Write to the TDR0 register.

When the first write to TDR0 register is performed, transmission starts.

● How to set the baud rate

See "17.7.1 Operations in Operation Mode 0".

● Interrupt-related registers

The interrupt level setting registers shown in the following table are used to set the interrupt level.

Channel	Interrupt level setting register	Interrupt vector
ch. 0	Interrupt level register(ILR1) Address: $0007A_H$	#4 Address: $0FFF2_H$

● How to enable/disable/clear interrupts

The interrupt request enable bits (SMC20:RIE, SMC20:TCIE, SMC20:TEIE) are used to enable interrupts.

	UART reception	UART transmission	
	Receive interrupt enable bit (RIE)	Transmission completion interrupt enable bit (TCIE)	Transmission data register empty interrupt enable bit (TEIE)
To disable interrupt requests	Set the bits to "0".		
To enable interrupt requests	Set the bits to "1".		

Interrupt requests are cleared in the following setup procedure.

	UART reception	UART transmission
To clear interrupt requests	Read from the UART/SIO serial input register (RDR0) to clear the reception data register full bit (RDRF).	Write data to the UART/SIO serial output data register (TDR0) to clear the transmission data register empty bit (TDRE) to "0".
	Write "0" to the error flag clear bit (RERC) to clear error flags (PER, OVE, FER) to "0".	

CHAPTER 18

UART/SIO DEDICATED BAUD RATE GENERATOR

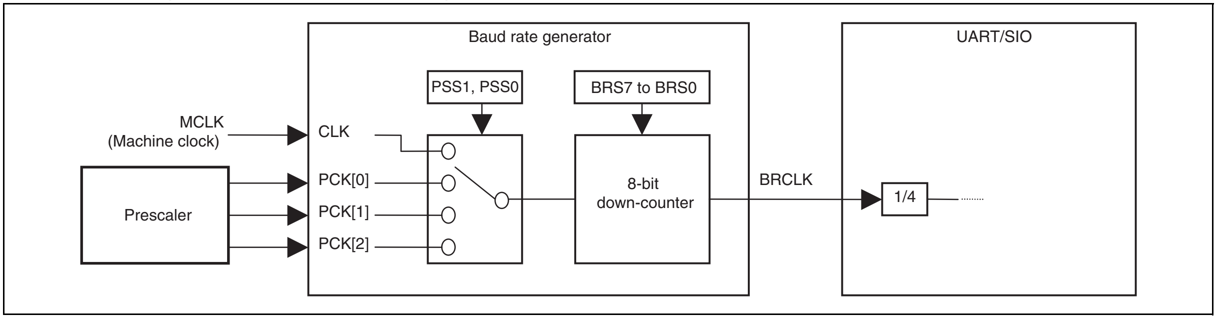
This chapter describes the functions and operations of the dedicated baud rate generator of UART/SIO.

18.1 Overview of UART/SIO Dedicated Baud Rate Generator

The UART/SIO dedicated baud rate generator generates the baud rate for the UART/SIO.
The generator consists of the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Block Diagram of UART/SIO Dedicated Baud Rate Generator

Figure 18.1-1 Block Diagram of UART/SIO Dedicated Baud Rate Generator



■ Input Clock

The UART/SIO dedicated baud rate generator uses the output clock from the prescaler or the machine clock as its input clock.

■ Output Clock

The UART/SIO dedicated baud rate generator supplies its clock to the UART/SIO.

18.2 Channel of UART/SIO Dedicated Baud Rate Generator

This section describes the channel of the UART/SIO dedicated baud rate generator.

■ Channel of UART/SIO Dedicated Baud Rate Generator

The MB95430H Series has one channel of UART/SIO dedicated baud rate generator.

Table 18.2-1 shows the registers of the UART/SIO dedicated baud rate generator.

Table 18.2-1 Registers of Dedicated Baud Rate Generator

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	PSSR0	UART/SIO dedicated baud rate generator prescaler selection register
	BRSR0	UART/SIO dedicated baud rate generator baud rate setting register

18.3 Registers of UART/SIO Dedicated Baud Rate Generator

The registers of the UART/SIO dedicated baud rate generator are namely the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Registers of UART/SIO Dedicated Baud Rate Generator

Figure 18.3-1 Registers of UART/SIO Dedicated Baud Rate Generator

UART/SIO dedicated baud rate generator prescaler selection register (PSSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBE_H$	-	-	-	-	-	BRGE	PSS1	PSS0	00000000_B
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	

UART/SIO dedicated baud rate generator baud rate setting register (BRSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBF_H$	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

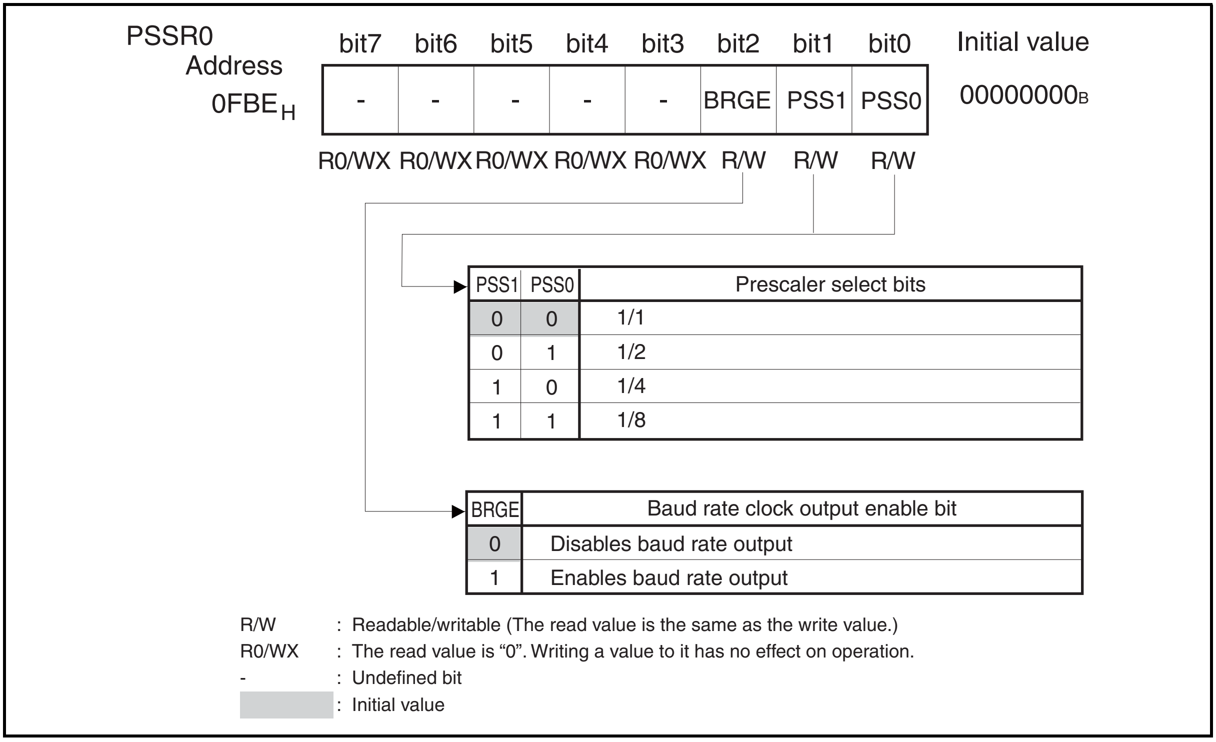
R/W	: Readable/writable (The read value is the same as the write value.)
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.
-	: Undefined bit

18.3.1 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

The UART/SIO dedicated baud rate generator prescaler register (PSSR0) controls the output of the baud rate clock and the prescaler.

■ UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

Figure 18.3-2 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)


PSSR0
Address
0FBE$_H$
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
Initial value
- - - - - BRGE PSS1 PSS0
00000000$_B$
R0/WX R0/WX R0/WX R0/WX R0/WX R/W R/W R/W

PSS1	PSS0	Prescaler select bits
0	0	1/1
0	1	1/2
1	0	1/4
1	1	1/8

BRGE	Baud rate clock output enable bit
0	Disables baud rate output
1	Enables baud rate output

R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 18.3-1 Functions of Bits in UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7
to
bit3</td><td>Undefined bits</td><td>These bits are undefined. Reading the bits always returns "0".</td></tr>
<tr><td>bit2</td><td>BRGE:
Baud rate clock output enable bit</td><td>This bit enables the output of the baud rate clock "BRCLK".
<b>Writing "1":</b> Loads BRS[7:0] to the 8-bit down-counter and outputs "BRCLK", which is supplied to the UART/SIO.
<b>Writing "0":</b> Stops the output of "BRCLK".</td></tr>
<tr><td>bit1,
bit0</td><td>PSS1, PSS0:
Prescaler select bits</td><td><table><tr><th>PSS1</th><th>PSS0</th><th>Prescaler selection</th></tr><tr><td>0</td><td>0</td><td>1/1</td></tr><tr><td>0</td><td>1</td><td>1/2</td></tr><tr><td>1</td><td>0</td><td>1/4</td></tr><tr><td>1</td><td>1</td><td>1/8</td></tr></table></td></tr>
</table>

18.3.2 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

The UART/SIO dedicated baud rate generator dedicated baud rate generator baud rate setting register (BRSR0) controls the baud rate settings.

■ UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

Figure 18.3-3 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FBF$_H$	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

This register sets the cycle of the 8-bit down-counter and can be used to set any baud rate clock. Write to the register when the UART is stopped.

Do not set BRS[7:0] to "00$_H$" or "01$_H$" in clock asynchronous mode.

18.4 Operations of UART/SIO Dedicated Baud Rate Generator

The UART/SIO dedicated baud rate generator serves as the baud rate generator for asynchronous clock mode.

■ Baud Rate Setting

The SMC10 register (CKS bit) of the UART/SIO is used to select the serial clock. This selects the UART/SIO dedicated baud rate generator.

In asynchronous CLK mode, the shift clock that is selected by the CKS bit and divided by four is used and transfers can be performed within the range from -2% to +2%. The baud rate calculation formula for the UART/SIO dedicated baud rate generator is shown below.

Figure 18.4-1 Baud Rate Calculation Formula when UART/SIO Dedicated Baud Rate Generator Is Used

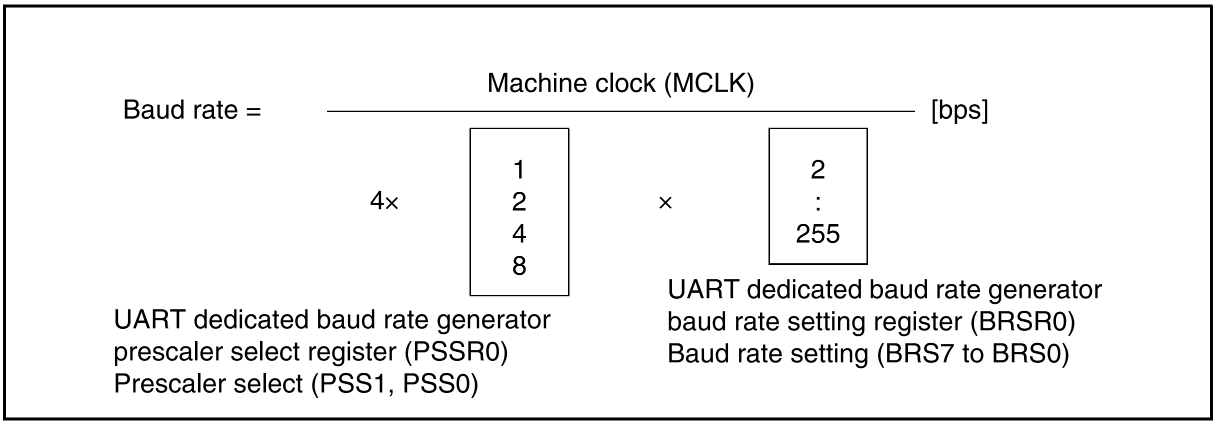

Baud rate =
Machine clock (MCLK)
[bps]
4×
1
2
4
8
×
2
:
255
UART dedicated baud rate generator
prescaler select register (PSSR0)
Prescaler select (PSS1, PSS0)
UART dedicated baud rate generator
baud rate setting register (BRSR0)
Baud rate setting (BRS7 to BRS0)

Table 18.4-1Sample Asynchronous Transfer Rates by Baud Rate Generator (Machine Clock = 10MHz, 16MHz, 16.25MHz)

UART/SIO Dedicated baud rate generator setting		UART internal division	Total division ratio (PSS × BRS x4)	Baud rate (10MHz/Total division ratio)	Baud rate (16MHz / Total division ratio)	Baud rate (16.25MHz / Total division ratio)
Prescaler select PSS[1:0]	Baud rate counter setting BRS [7:0]					
1 (Setting value: 0, 0)	20	4	80	125000	200000	203125
1 (Setting value: 0, 0)	22	4	88	113636	181818	184659
1 (Setting value: 0, 0)	44	4	176	56818	90909	92330
1 (Setting value: 0, 0)	87	4	348	28736	45977	46695
1 (Setting value: 0, 0)	130	4	520	19231	30769	31250
2 (Setting value: 0, 1)	130	4	1040	9615	15385	15625
4 (Setting value: 1, 0)	130	4	2080	4808	7692	7813
8 (Setting value: 1, 1)	130	4	4160	2404	3846	3906

The baud rate can be set in UART mode within the following range.

Table 18.4-2 Permissible Baud Rate Range in UART Mode

PSS[1:0]	BRS[7:0]
"00_B" to "11_B"	02_H (2) to FF_H (255)

CHAPTER 19

I^2C

This chapter describes functions and operations of the I^2C.

19.1 Overview of I^{2}C

The I^{2}C interface supports the I^{2}C bus specification published by Philips. The interface provides the functions of transmission and reception in master and slave modes, detection of arbitration lost, detection of slave address and general call address, generation and detection of start and stop conditions, bus error detection, and MCU standby wakeup.

■ I^{2}C Functions

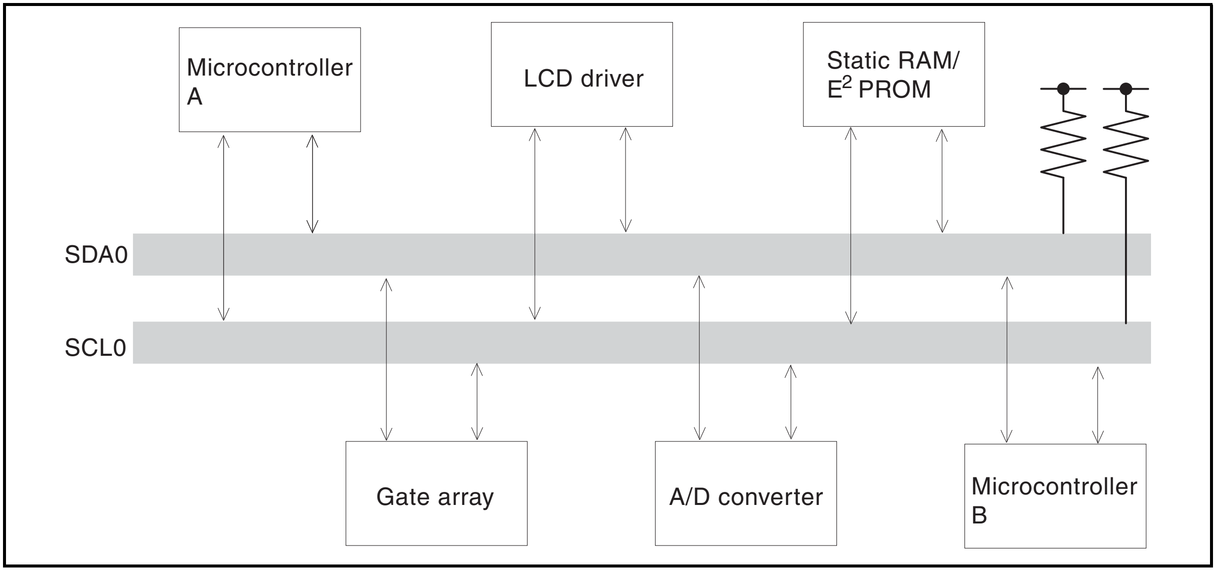
The I^{2}C interface is a two-wire, bi-directional bus consisting of a serial data line (SDA) and serial clock line (SCL). The devices connected to the bus via these two wires can exchange data, and each device can operate as a sender or receiver in accordance with their respective functions based on the unique address assigned to each device. Furthermore, the interface establishes a master/slave relationship between devices.

Also, the I^{2}C interface can connect multiple devices provided the bus capacitance does not exceed an upper limit of 400 pF. The I^{2}C interface is a true multi-master bus with collision detection and a communication control protocol that prevent loss of data even if more than one master attempts to start a data transfer at the same time.

The communication control protocol ensures that only one master is able to take control of the bus at a time, even if multiple masters attempt to take control of the bus simultaneously, without messages being lost or data being altered. Multi-master means that more than one master can attempt to take control of the bus at the same time without causing messages to be lost.

Also, the I^{2}C interface includes a function to wake up the MCU from standby mode.

Figure 19.1-1 I^{2}C Interface Configuration



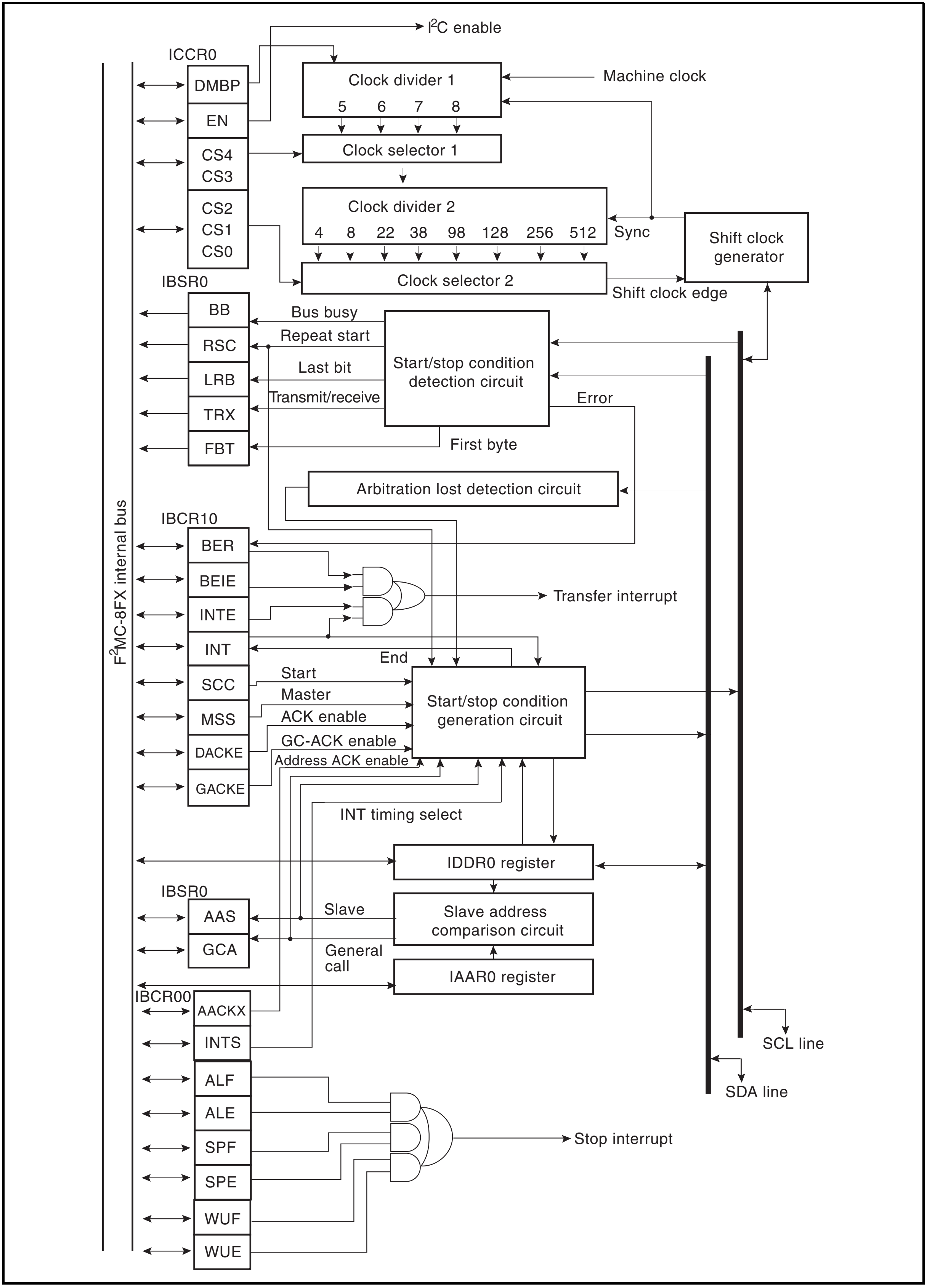
19.2 I^{2}C Configuration

I^{2}C consists of the following blocks:

- **Clock selector**
- **Clock divider**
- **Shift clock generator**
- **Start/stop condition generation circuit**
- **Start/stop condition detection circuit**
- **Arbitration lost detection circuit**
- **Slave address comparison circuit**
- **IBSR0 register**
- **IBCR registers (IBCR00, IBCR10)**
- **ICCR0 register**
- **IAAR0 register**
- **IDDR0 register**

■ Block Diagram of I^2C

Figure 19.2-1 Block Diagram of I^2C



- Clock selector, clock divider, and shift clock generator

This circuit uses the machine clock to generate the shift clock for the I^{2}C bus.

- Start/stop condition generation circuit

When a start condition is transmitted with the bus idle (SCL and SDA at the "H" level), a master starts communications. When SCL = "H", a start condition is generated by changing the SDA line from "H" to "L". The master can terminate its communication by generating a stop condition. When SCL = "H", a stop condition is generated by changing the SDA line from "L" to "H".

- Start/stop condition detection circuit

This circuit detects a start/stop condition for data transfer.

- Arbitration lost detection circuit

This interface circuit supports multi-master systems. If two or more masters attempt to transmit at the same time, the arbitration lost condition (if logic level "1" is sent when the SDA line goes to the "L" level) occurs. When the arbitration lost is detected, IBCR00:ALF is set to "1" and the master changes to a slave automatically.

- Slave address comparison circuit

The slave address comparison circuit receives the slave address after the start condition to compare it with its own slave address. The address is seven-bit data followed by a data direction (R/W) bit in the eighth bit position. If the received address matches the own slave address, the comparison circuit transmits an acknowledgment.

- IBSR0 register

The IBSR0 register shows the status of the I^{2}C interface.

- IBCR registers (IBCR00, IBCR10)

The IBCR registers are used to select the operating mode and to enable or disable interrupts, acknowledgment, general call acknowledgment, and the function to wake up the MCU from standby mode.

- ICCR0 register

The ICCR0 register is used to enable I^{2}C interface operations and select the shift clock frequency.

- IAAR0 register

The IAAR0 register is used to set the slave address.

- IDDR0 register

The IDDR0 register holds the transmit or receive shift data or address. When transmitted, the data or address written to this register is transferred from the MSB first to the bus.

■ Input Clock

I^{2}C uses the machine clock as the input clock (shift clock).

19.3 I^{2}C Channel

This section describes the I^{2}C channel.

■ I^{2}C Channel

The MB95430H Series has one channel of I^{2}C.

Table 19.3-1 and Table 19.3-2 show the pins and registers of I^{2}C respectively.

Table 19.3-1 Pins of I^{2}C

Channel	Pin name	Pin function
0	SCL SDA	I^{2}C bus I/O

Table 19.3-2 I^{2}C Registers

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	IBCR00	I^{2}C bus control register 0
	IBCR10	I^{2}C bus control register 1
	IBSR0	I^{2}C bus status register
	IDDR0	I^{2}C data register
	IAAR0	I^{2}C address register
	ICCR0	I^{2}C clock control register

19.4 I^{2}C Bus Interface Pins

This section describes the pins of the I^{2}C bus interface and gives their block diagram.

■ Pins of I^{2}C Bus Interface

The pins of the I^{2}C bus interface are SDA and SCL.

● SDA pin

The SDA pin can serve as a general-purpose I/O port, external interrupt input (hysteresis input), serial data output pin (N-ch open drain) for 8-bit serial I/O, and I^{2}C data I/O pin (SDA).

SDA: When I^{2}C is enabled (ICCR0:EN = 1), the SDA pin is automatically set as a data I/O pin to function as the SDA pin.

To use it as an input pin, enable the I^{2}C operation (ICCR0: EN = 1) and write "0" to bit0 in the corresponding port direction register (DDR).

The SDA pin can be assigned to P03 or P65 by modifying the setting of the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

● SCL pin

The SCL pin can serve as an N-ch open drain I/O port, external interrupt input (hysteresis input), serial data input (hysteresis input) for eight-bit serial I/O, or I^{2}C serial clock I/O pin (SCL).

SCL: When I^{2}C is enabled (ICCR0:EN = 1), the SCL pin is automatically set as the shift clock I/O pin to function as the SCL pin.

To use it as an input pin, enable the I^{2}C operation (ICCR0: EN = 1) and write "0" to bit1 in the the corresponding port direction register (DDR).

The SCL pin can be assigned to P04 or P12 by modifying the setting of the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

■ Block Diagram of I2C-related Pins

Figure 19.4-1 Block Diagram of I2C-related Pins SCL and SDA (P04/AN04/INT04/UI/SCL and P03/INT03/AN03/SDA)

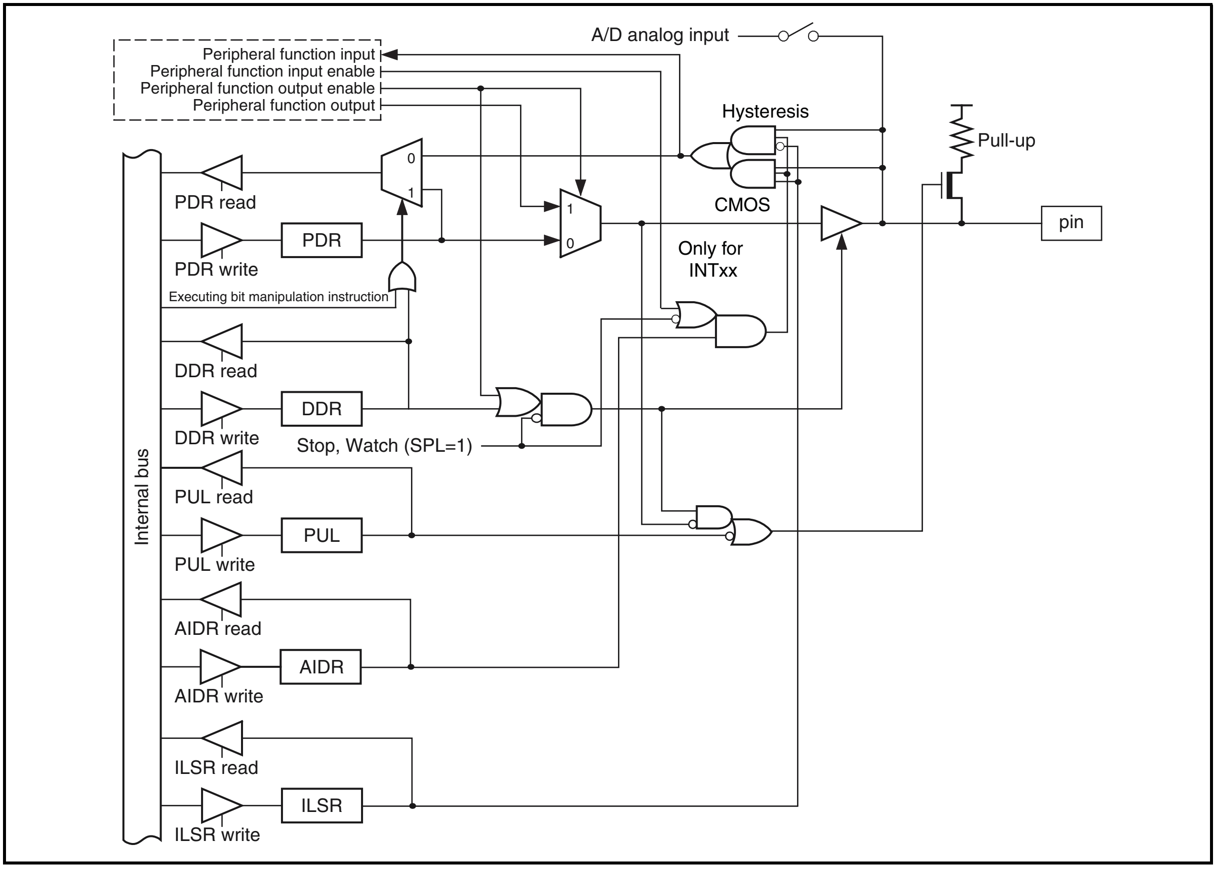


Figure 19.4-2 Block Diagram of I^{2}C-related Pin SCL (P12/EC0/UI/SCL/DBG)

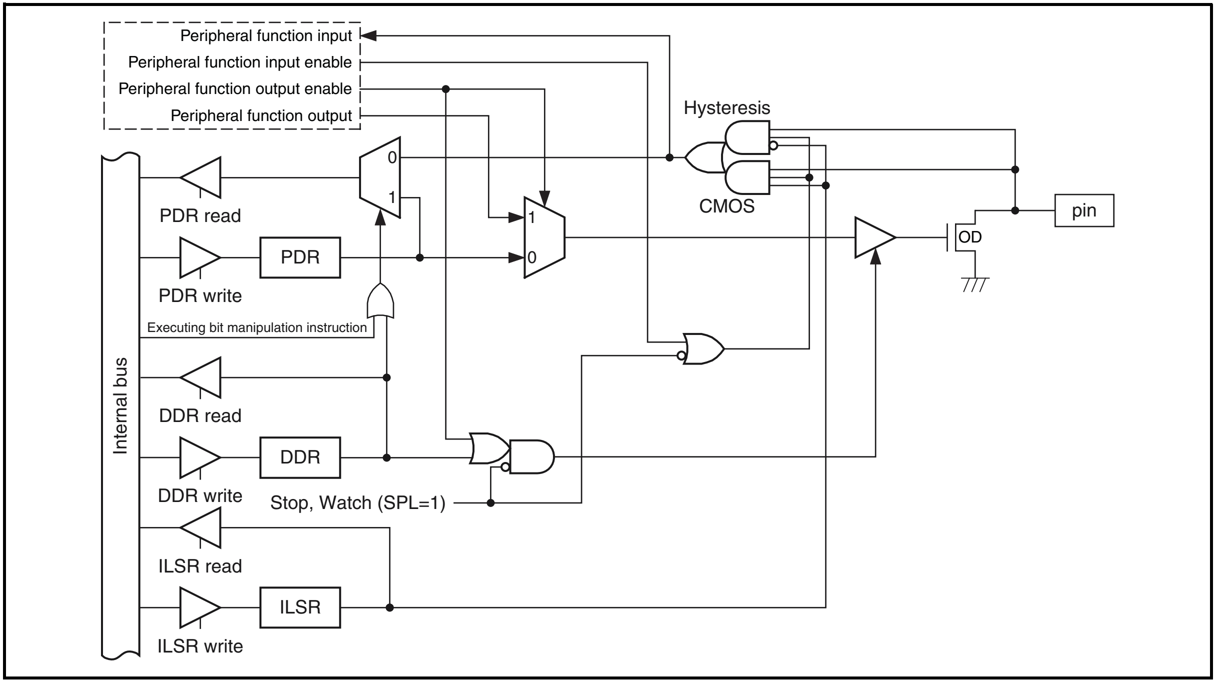
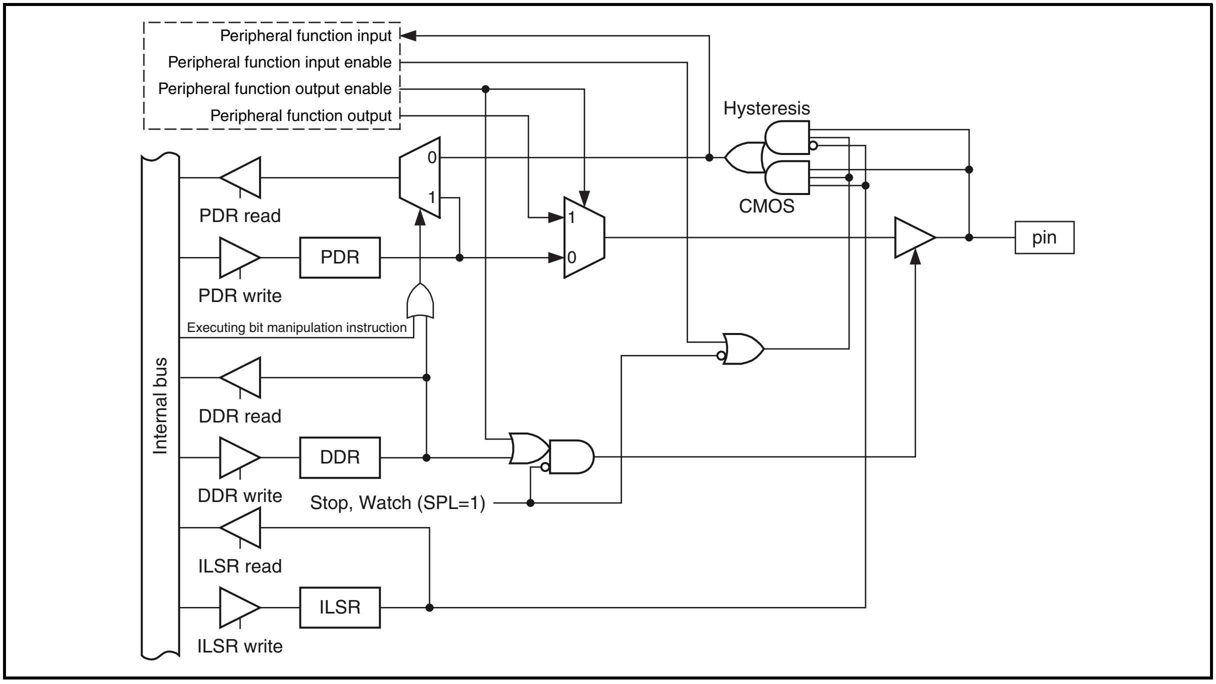

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
pin
OD
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

Figure 19.4-3 Block Diagram of I^{2}C-related Pin SDA (P65/CMP3_O/UO/SDA)


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
pin
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

MB95430H Series

19.5 Registers of I^{2}C

This section describes the registers of I^{2}C.

■ Registers of I^{2}C

Figure 19.5-1 Registers of I^{2}C

I^{2}C bus control register 0 (IBCR00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050_H	AACKX	INTS	ALF	ALE	SPF	SPE	WUF	WUE	00000000_B
	R/W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	

I^{2}C bus control register 1 (IBCR10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0051_H	BER	BEIE	SCC	MSS	DACKE	GACKE	INTE	INT	00000000_B
	R(RM1),W	R/W	R0,W	R/W	R/W	R/W	R/W	R(RM1),W	

I^{2}C bus status register (IBSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0052_H	BB	RSC	-	LRB	TRX	AAS	GCA	FBT	00000000_B
	R/WX	R/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

I^{2}C data register (IDDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053_H	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

I^{2}C address register (IAAR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0054_H	-	A6	A5	A4	A3	A2	A1	A0	00000000_B
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

I^{2}C clock control register (ICCR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0055_H	DMBP	-	EN	CS4	CS3	CS2	CS1	CS0	00000000_B
	R/W	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

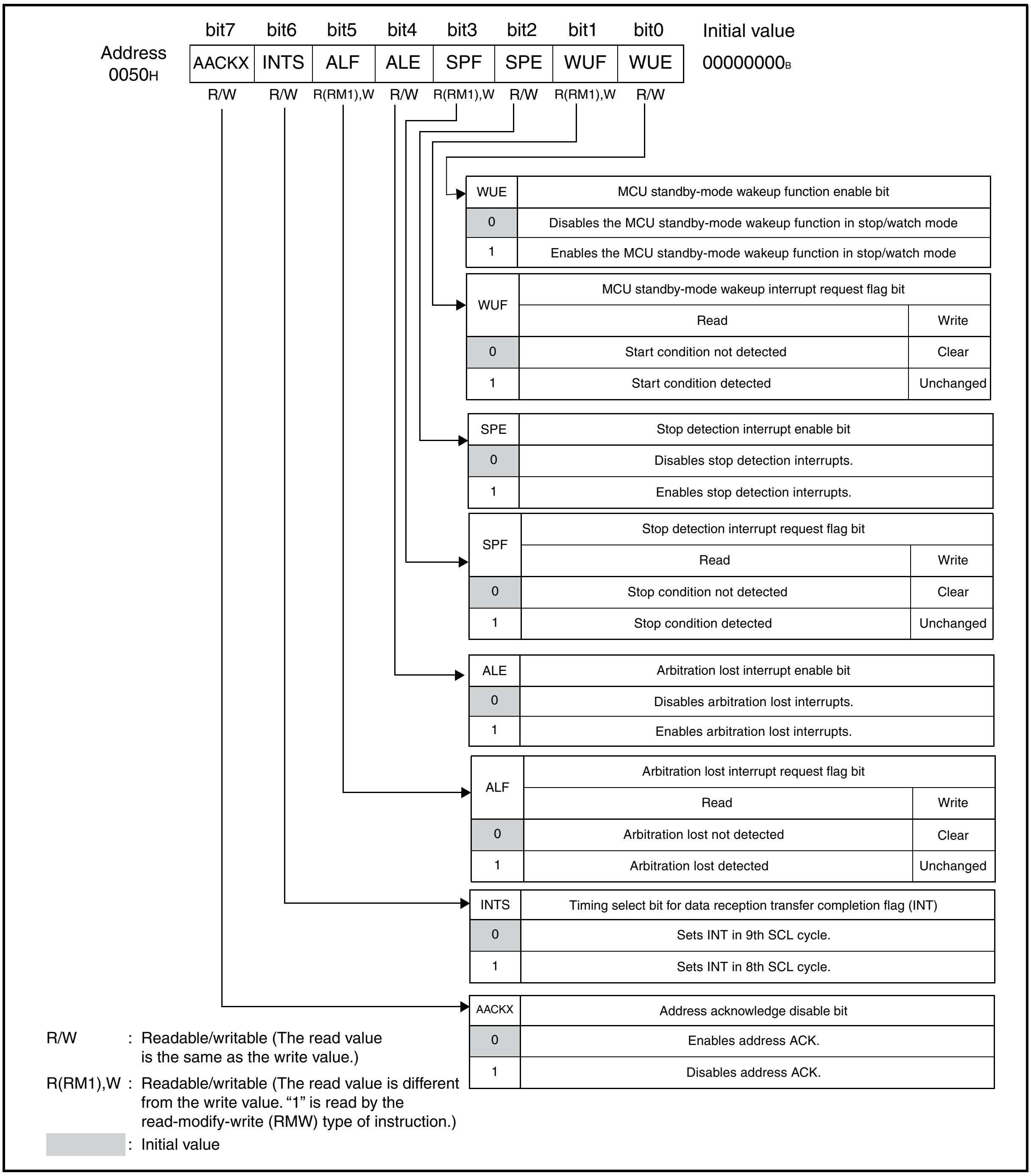
R/W	: Readable/writable (The read value is the same as the write value.)
R(RM1),W	: Readable/writable (The read value is different from write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R0,W	: Write only (Writable. The read value is "0".)
R/WX	: Read only (Readable. Writing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

19.5.1 I2C Bus Control Registers (IBCR00, IBCR10)

The I2C bus control registers are used to select the operating mode and to enable or disable interrupts, acknowledgment, general call acknowledgment, and MCU standby wakeup function.

■ I2C Bus Control Register 0 (IBCR00)

Figure 19.5-2 I2C Bus Control Register 0 (IBCR00)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0050H	AACKX	INTS	ALF	ALE	SPF	SPE	WUF	WUE	00000000B
	R/W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	

WUE	MCU standby-mode wakeup function enable bit
0	Disables the MCU standby-mode wakeup function in stop/watch mode
1	Enables the MCU standby-mode wakeup function in stop/watch mode

WUF	MCU standby-mode wakeup interrupt request flag bit	
	Read	Write
0	Start condition not detected	Clear
1	Start condition detected	Unchanged

SPE	Stop detection interrupt enable bit
0	Disables stop detection interrupts.
1	Enables stop detection interrupts.

SPF	Stop detection interrupt request flag bit	
	Read	Write
0	Stop condition not detected	Clear
1	Stop condition detected	Unchanged

ALE	Arbitration lost interrupt enable bit
0	Disables arbitration lost interrupts.
1	Enables arbitration lost interrupts.

ALF	Arbitration lost interrupt request flag bit	
	Read	Write
0	Arbitration lost not detected	Clear
1	Arbitration lost detected	Unchanged

INTS	Timing select bit for data reception transfer completion flag (INT)
0	Sets INT in 9th SCL cycle.
1	Sets INT in 8th SCL cycle.

AACKX	Address acknowledge disable bit
0	Enables address ACK.
1	Disables address ACK.

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 19.5-1 Functions of Bits in I^{2}C Bus Control Register 0 (IBCR00) (1 / 2)

Bit name		Function
bit7	AACKX: Address acknowledge disable bit	This bit controls the address ACK when the first byte is transmitted. **Writing "0":** Causes the address ACK to be output automatically (The address ACK is returned automatically if the slave address matches). **Writing "1":** Prevents the address ACK from being output. Write "1" to this bit in either of the following ways: - Write "1" to the bit in master mode. - Clear the bit to "0" after making sure that the bus busy bit is "0" (IBSR0:BB = 0). Notes: • If AACKX =1 and IBSR0:FBT =0 when an IBCR10:INT bit interrupt occurs, no address ACK is output even though the I^{2}C address matches the slave address. Clear the IBCR10:INT bit to "0" as an interrupt is generated upon completion of transfer of each byte of address/data in the same way as during addressing. • If AACKX =1 and IBSR0:FBT =1 when an IBCR10:INT bit interrupt occurs, "1" might be written to AACKX after addressing as in slave mode. Either continue normal communication after setting AACKX to "0" again or restart communication after disabling I^{2}C operation (ICCR0:EN = 0).
bit6	INTS: Timing select bit for data reception transfer completion flag (INT)	This bit selects the timing of the transfer completion interrupt (IBCR10:INT) when data is received. Change the bit only when IBSR0:TRX = 0 and IBSR0:FBT = 0. **Writing "0":** Sets the transfer completion interrupt (IBCR10:INT) in the ninth SCL cycle. **Writing "1":** Sets the transfer completion interrupt (IBCR10:INT) in the eighth SCL cycle. Notes: • The transfer completion interrupt (IBCR10:INT) is set always in the ninth SCL cycle except during data reception (IBSR0:TRX = 1 or IBSR0:FBT = 1). • If the data ACK depends on the content of the received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACKE) after writing "1" to this bit (for example, using a previous transfer completion interrupt) to read latest received data. • The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt in the ninth SCL cycle.) If ACK is read when this bit is "1", therefore, you must write "0" to this bit in the transfer completion interrupt in the eighth SCL cycle so that another transfer completion interrupt will occur in the ninth SCL cycle.
bit5	ALF: Arbitration lost interrupt request flag bit	This bit is used to detect when arbitration is lost. • An arbitration lost interrupt request is generated if this bit and the IBCR00:ALE bit are both "1". • This bit is set to "1" in the following cases: - When arbitration lost is detected during data/address transmission as a master - When "1" is written to the IBCR10:MSS bit with the bus being used by another system. However, the bit is not set when "1" is written to the MSS bit after the system returns AACK or GACK as a slave. • This bit is set to "0" in the following cases: - When "0" is written to the IBCR00:ALF bit with IBSR0:BB = 0. - When "0" is written to the IBCR10:INT bit to clear the transmission completion flag. • Writing "1" to this bit leaves its value unchanged and has no effect on the operation. • The bit returns "1" when read by a read-modify-write (RMW) instruction.
bit4	ALE: Arbitration lost interrupt enable bit	This bit enables or disables arbitration lost interrupts. An arbitration lost interrupt request is generated if this bit and the IBCR00:ALF bit are both "1". **Writing "0":** Disables arbitration lost interrupts. **Writing "1":** Enables arbitration lost interrupts.
bit3	SPF: Stop detection interrupt request flag bit	This bit is used to detect a stop condition. • A stop detection interrupt request is generated if this bit and the IBCR00:SPE bit are both "1". • This bit is set to "1" if a valid stop condition is detected when the bus is busy. **Writing "0":** Clears itself (changes the value to "0"). **Writing "1":** Leaves its value unchanged without affecting the operation. • The bit returns "1" when read by a read-modify-write (RMW) instruction.

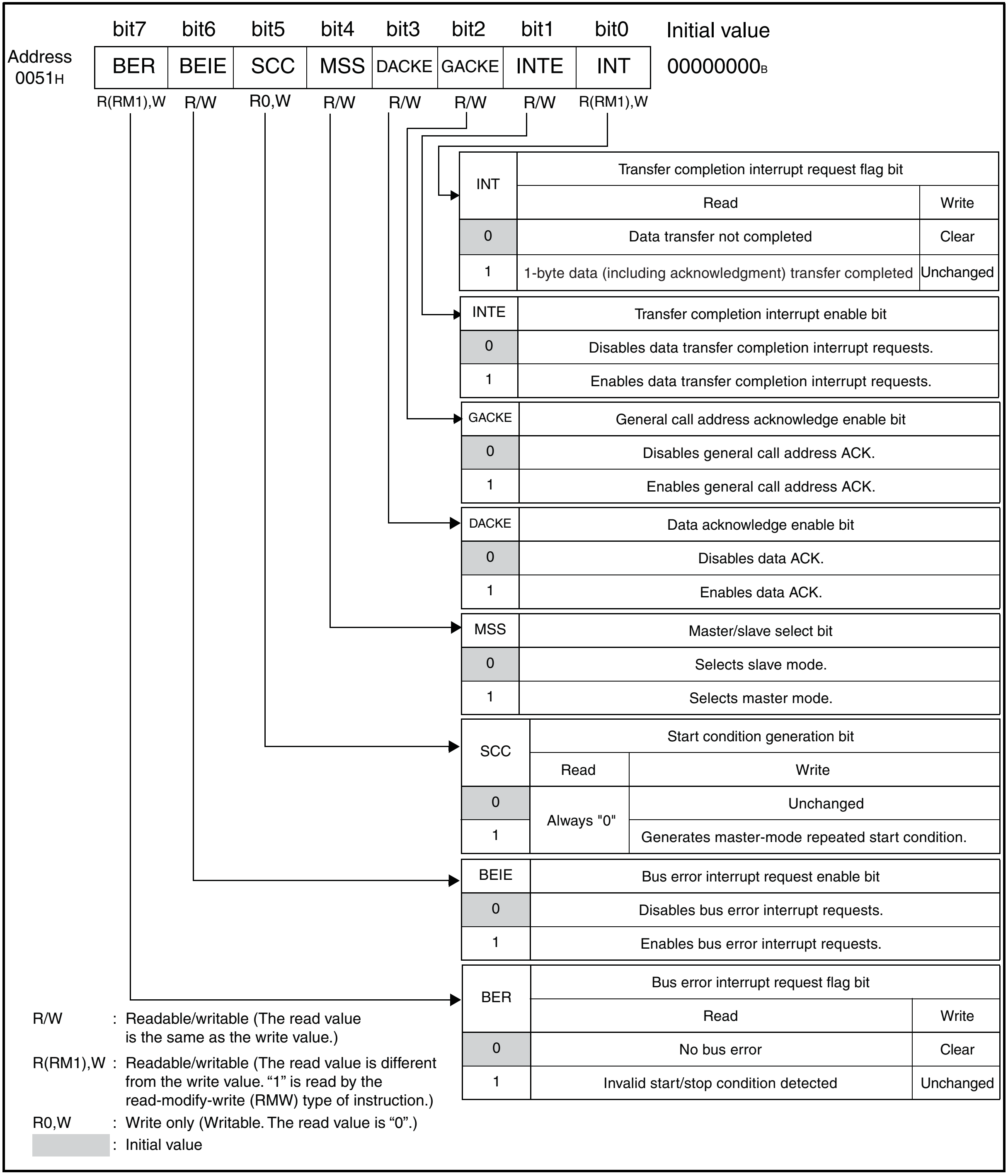
Table 19.5-1 Functions of Bits in I^{2}C Bus Control Register 0 (IBCR00) (2 / 2)

Bit name		Function
bit2	SPE: Stop detection interrupt enable bit	This bit enables or disables stop detection interrupts. A stop detection interrupt request is generated if this bit and the IBCR00:SPF bit are both "1". **Writing "0":** Disables stop detection interrupts. **Writing "1":** Enables stop detection interrupts.
bit1	WUF: MCU standby-mode wakeup interrupt request flag bit	This bit is used to detect MCU wakeup from a standby mode (stop or watch mode). • A wakeup interrupt request is generated if this bit and the IBCR00:WUE bit are both "1". • This bit is set to "1" if a start condition is detected with the wakeup function enabled (IBCR00:WUE = 1). **Writing "0":** Clears itself (changes the value to "0"). **Writing "1":** Leaves its value unchanged without affecting the operation. • The bit returns "1" when read by a read-modify-write (RMW) instruction.
bit0	WUE: MCU standby-mode wakeup function enable bit	This bit enables or disables the function to wake up the MCU from standby mode (stop or watch mode). **Writing "0":** Disables the wakeup function. **Writing "1":** Enables the wakeup function. If a start condition is detected in stop or watch mode when this bit is "1", a wakeup interrupt request is generated to start I^{2}C operation. Notes: • Write "1" to this bit immediately before the MCU enters the stop or watch mode. To ensure that I^{2}C operation can restart immediately after the MCU wakes up from stop or watch mode, clear (write "0" to) this bit as soon as possible. • When a wakeup interrupt request occurs, the MCU wakes up after the oscillation stabilization wait time elapses. To prevent the data loss immediately after wakeup, therefore, the SCL must rise as the first cycle and the first bit must be received as data after 100 µs (assuming that the minimum oscillation stabilization wait time is 100 µs) from the wakeup due to the start of I^{2}C transmission (upon detection of the falling edge of SDA). • During a MCU standby mode, the status flags, state machine, and I^{2}C bus outputs for the I^{2}C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I^{2}C bus system, make sure that IBSR0:BB = 0 before entering standby mode. • The wakeup function does not support the transition of the MCU to stop or watch mode with IBSR0:BB = 1. If the MCU enters stop or watch mode with IBSR0:BB = 1, a bus error will occur upon detection of a start condition. • The wakeup function is useful only when the MCU remains in stop/watch mode.

Note: The AACKX, INTS, and WUE bits in the IBCR00 register are set to "0" and no values can be written to them either when I^{2}C operation is disabled (ICCR:EN = 0) or when a bus error occurs (IBSR:BER = 1).

■ I2C Bus Control Register 1 (IBCR10)

Figure 19.5-3 I2C Bus Control Register 1 (IBCR10)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0051H	BER	BEIE	SCC	MSS	DACKE	GACKE	INTE	INT	00000000B
	R(RM1),W	R/W	R0,W	R/W	R/W	R/W	R/W	R(RM1),W	

INT	Transfer completion interrupt request flag bit	
	Read	Write
0	Data transfer not completed	Clear
1	1-byte data (including acknowledgment) transfer completed	Unchanged

INTE	Transfer completion interrupt enable bit
0	Disables data transfer completion interrupt requests.
1	Enables data transfer completion interrupt requests.

GACKE	General call address acknowledge enable bit
0	Disables general call address ACK.
1	Enables general call address ACK.

DACKE	Data acknowledge enable bit
0	Disables data ACK.
1	Enables data ACK.

MSS	Master/slave select bit
0	Selects slave mode.
1	Selects master mode.

SCC	Start condition generation bit	
	Read	Write
0	Always "0"	Unchanged
1		Generates master-mode repeated start condition.

BEIE	Bus error interrupt request enable bit
0	Disables bus error interrupt requests.
1	Enables bus error interrupt requests.

BER	Bus error interrupt request flag bit	
	Read	Write
0	No bus error	Clear
1	Invalid start/stop condition detected	Unchanged

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.)

R0,W : Write only (Writable. The read value is “0”.)

▨ : Initial value

Table 19.5-2 Functions of Bits in I^{2}C Bus Control Register 1 (IBCR10) (1 / 2)

Bit name		Function
bit7	BER: Bus error interrupt request flag bit	This bit is used to detect bus errors. • A bus error interrupt request is generated if this bit and the IBCR10:BEIE bit are both "1". • This bit is set to "1" when an invalid start or stop condition is detected. **Writing "0":** Clears itself (changes the value to "0"). **Writing "1":** Leaves its value unchanged without affecting the operation. • The bit returns "1" when read by a read-modify-write (RMW) instruction. • When this bit is set to "1", ICCR0:EN is set to "0", and the I^{2}C interface enters halt mode to terminate data transfer.
bit6	BEIE: Bus error interrupt request enable bit	This bit enables or disables bus error interrupts. A bus error interrupt request is generated if this bit and the IBCR10:BER bit are both "1". **Writing "0":** Disables bus error interrupts. **Writing "1":** Enables bus error interrupts.
bit5	SCC: Start condition generation bit	This bit can be used to generate a start condition repeatedly to restart communications in master mode. • Writing "1" to the bit in master mode generates a start condition repeatedly. • Writing "0" to the bit is meaningless. • When read, the bit returns "0". Notes: • Do not set IBCR10:SCC = 1 and IBCR10:MSS = 0 at the same time. • An attempt to write "1" to this bit is ignored when IBCR10:INT = 0 (no start condition is generated). If you write "1" to this bit and "0" to the IBCR10:INT bit at the same time when the IBCR10:INT = 1, this bit takes priority and generates a start condition.
bit4	MSS: Master/slave select bit	This bit selects master mode or slave mode. • Writing "1" to this bit while the I^{2}C bus is in the idle state (IBSR0:BB = 0) selects master mode, generates a start condition, and then starts address transfer. • Writing "0" to the bit while the I^{2}C bus is in the busy state (IBSR0:BB = 1) selects slave mode, generates a stop condition, and then ends data transfer. • If arbitration lost occurs during data or address transfer in master mode, this bit is cleared to "0" and the mode changes to slave mode. Notes: • Do not set IBCR10:SCC = 1 and IBCR10:MSS = 0 at the same time. • An attempt to write "0" to this bit is ignored when IBCR10:INT = 0. If you write "0" to this bit and "0" to the IBCR10:INT bit at the same time when the IBCR10:INT = 1, this bit takes priority and generates a stop condition. • The IBCR00:ALF bit is not set even though you write "1" to the MSS bit during transmission or reception in slave mode. Do not write "1" to the MSS bit during transmission or reception in slave mode.
bit3	DACKE: Data acknowledge enable bit	This bit controls data acknowledgment during data reception. **Writing "0":** Disables data acknowledge output. **Writing "1":** Enables data acknowledge output. In this case, data acknowledgment is output in the ninth SCL cycle during data reception in master mode. In slave mode, data acknowledgment is output in the ninth SCL cycle only if address acknowledgment has already been output.
bit2	GACKE: General call address acknowledge enable bit	This bit controls general call address acknowledgment. **Writing "0":** Disables output of general call address acknowledge. **Writing "1":** Causes a general call address acknowledgment to be output if a general call address (00_H) is received in master or slave mode.
bit1	INTE: Transfer completion interrupt enable bit	This bit enables or disables transfer completion interrupts. **Writing "0":** Disables transfer completion interrupts. **Writing "1":** Enables transfer completion interrupts. A transfer completion interrupt request is generated if this bit and the IBCR10:INT bit are both "1".

Table 19.5-2 Functions of Bits in I^{2}C Bus Control Register 1 (IBCR10) (2 / 2)

Bit name		Function
bit0	INT: Transfer completion interrupt request flag bit	This bit is used to detect transfer completion. • A transfer completion interrupt request is generated if this bit and the IBCR10:INTE bit are both "1". • This bit is set to "1" upon completion of transfer of 1-byte address or data (whether or not this includes an acknowledgment depends on the IBCR00:INTS setting) if any of the following four conditions is satisfied. - In bus master mode - Addressed as slave - General call address received - Arbitration lost detected • This bit is set to "0" in the following cases: - "0" written to the bit - Repeated start condition (IBCR10:SCC = 1) or stop condition (IBCR10:MSS = 0) occurred in master mode. • An attempt to write "1" to this bit leaves its value unchanged and has no effect on the operation. • The bit returns "1" when read by a read-modify-write (RMW) instruction. • The SCL line remains at "L" while this bit is "1". • Writing "0" to clear the bit (change the value to "0") releases the SCL line to enable transmission for the next byte of data. Notes: • If "1" is written to IBCR10:SCC when this bit is "0", the IBCR10:SCC bit has priority and the start condition is generated. • If "0" is written to IBCR10:MSS when this bit is "0", the IBCR10:MSS bit has priority and the stop condition is generated. • If IBCR00:INTS = 1 when data is received, this bit is set to "1" upon completion of transfer of one-byte data (including no acknowledgment). In other cases, this bit is set to "1" upon completion of transmission or reception of one-byte data/ address including an acknowledgment.

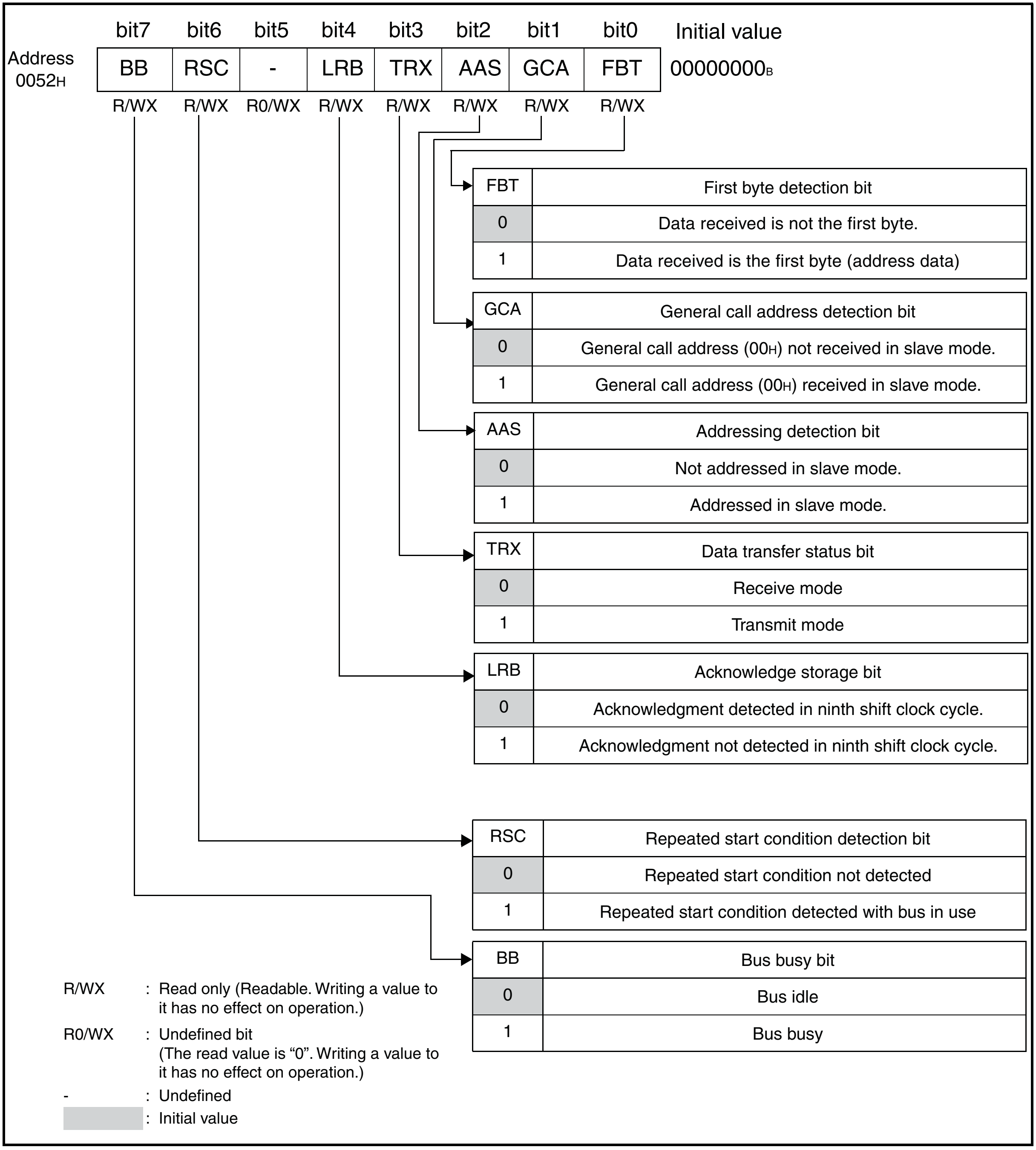
Notes:
- When clearing the interrupt request flag (IBCR10:BER) by writing "0", do not update the interrupt request enable bit (IBCR10:BEIE) at the same time.
- All the bits in IBCR10 except the BER and BEIE bits are cleared to "0" either when operation is disabled (ICCR:EN = 0) or when a bus error occurs (IBSR0:BER = 1).

19.5.2 I^{2}C Bus Status Register (IBSR0)

The IBSR0 register indicates the status of the I^{2}C interface.

■ I^{2}C Bus Status Register (IBSR0)

Figure 19.5-4 I^{2}C Bus Status Register (IBSR0)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0052$_H$	BB	RSC	-	LRB	TRX	AAS	GCA	FBT	00000000$_B$
	R/WX	R/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

FBT	First byte detection bit
0	Data received is not the first byte.
1	Data received is the first byte (address data)

GCA	General call address detection bit
0	General call address (00$_H$) not received in slave mode.
1	General call address (00$_H$) received in slave mode.

AAS	Addressing detection bit
0	Not addressed in slave mode.
1	Addressed in slave mode.

TRX	Data transfer status bit
0	Receive mode
1	Transmit mode

LRB	Acknowledge storage bit
0	Acknowledgment detected in ninth shift clock cycle.
1	Acknowledgment not detected in ninth shift clock cycle.

RSC	Repeated start condition detection bit
0	Repeated start condition not detected
1	Repeated start condition detected with bus in use

BB	Bus busy bit
0	Bus idle
1	Bus busy

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.)

- : Undefined

▨ : Initial value

Table 19.5-3 Functions of Bits in I^{2}C Bus Status Register (IBSR0)

Bit name		Function
bit7	BB: Bus busy bit	This bit indicates the bus status. • This bit is set to "1" when a start condition is detected. • This bit is set to "0" when a stop condition is detected.
bit6	RSC: Repeated start condition detection bit	This bit is used to detect repeated start conditions. • This bit is set to "1" when a repeated start condition is detected. • This bit is set to "0" in the following cases: - When "0" is written to IBCR10:INT. - When the slave address does not match the address set in IAAR0 in slave mode. - When the slave address matches the address set in IAAR0 but IBCR00:AACKX = 1 in slave mode. - When the general call address is received but IBCR10:GACKE = 0 in slave mode. - When a stop condition is detected.
bit5	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit4	LRB: Acknowledge storage bit	This bit saves the value of the SDA line in the ninth shift clock cycle during data byte transfer. • This bit is set to "1" when no acknowledgment is detected (SDA = "H"). • This bit is set to "0" in the following cases: - When acknowledgment is detected (SDA = "L") - When a start or stop condition is detected. Note: It follows from the above that this bit must be read after ACK (Read the value in response to the transfer completion interrupt in the ninth SCL cycle). Accordingly, if ACK is read when the IBCR00:INTS bit is "1", you must write "0" to the IBCR00:INTS bit in the transfer completion interrupt triggered by the eighth SCL cycle so that another transfer completion interrupt will be triggered by the ninth SCL cycle.
bit3	TRX: Data transfer status bit	This bit indicates the data transfer mode. • This bit is set to "1" when data transfer is performed in transfer mode. • This bit is set to "0" in the following cases: - Data is transferred in receive mode. - NACK is received in slave transmit mode.
bit2	AAS: Addressing detection bit	This bit indicates that the MCU has been addressed in slave mode. • This bit is set to "1" if the MCU is addressed in slave mode. • This bit is set to "0" when a start or stop condition is detected.
bit1	GCA: General call address detection bit	This bit is used to detect a general call address. • This bit is set to "1" in the following cases: - When the general call address (00_H) is received in slave mode. - When the general call address (00_H) is received in master mode with IBCR10:GACKE = 1. - When arbitration lost is detected during transmission of the second byte of the general call address in master mode. • This bit is set to "0" in the following cases: - When a start or stop condition is detected. - When arbitration lost is not detected during transmission of the second byte of the general call address in master mode.
bit0	FBT: First byte detection bit	This bit is used to detect first byte. • This bit is set to "1" when a start condition is detected. • This bit is set to "0" in the following cases: - When "0" is written to the IBCR10:INT bit. - When the slave address does not match the address set in IAAR0 in slave mode. - When the slave address matches the address set in IAAR0 but IBCR00:AACKX = 1 in slave mode. - When the general call address is received with IBCR10:GACKE = 0 in slave mode.

19.5.3 I^{2}C Data Register (IDDR0)

The IDDR0 register is used to set the data or address to send and to hold the data or address received.

■ I^{2}C Data Register (IDDR0)

Figure 19.5-5 I^{2}C Data Register (IDDR0)

I^{2}C data register (IDDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053_H	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

In transmit mode, each bit of the data or address value written to the register is shifted to the SDA line, starting with the MSB. The write side of this register is double-buffered, where if the bus is in use (IBSR0:BB=1), the write data is loaded to the 8-bit shift register either when the current data transfer completion interrupt is cleared (writing "0" to the IBCR10:INT bit) or when a repeated start condition is generated (writing "1" to the IBCR10:SCC bit). Each bit of the shift register data is output (shifted) to the SDA line.

Note that writing to this register has no effect on the current data transfer. In slave mode, however, data is transferred to the shift register after the address is determined.

The received data or address can be read from this register during the transfer completion interrupt (IBCR10:INT = 1). When it is read, however, the serial transfer register is directly read from, the receive data is valid only while IBCR10:INT = 1.

19.5.4 I^{2}C Address Register (IAAR0)

The IAAR0 register is used to set the slave address.

■ I^{2}C Address Register (IAAR0)

Figure 19.5-6 I^{2}C Address Register (IAAR0)

I^{2}C address register (IAAR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0054_H	-	A6	A5	A4	A3	A2	A1	A0	00000000_B
	R0/WX	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.)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
- : Undefined bit

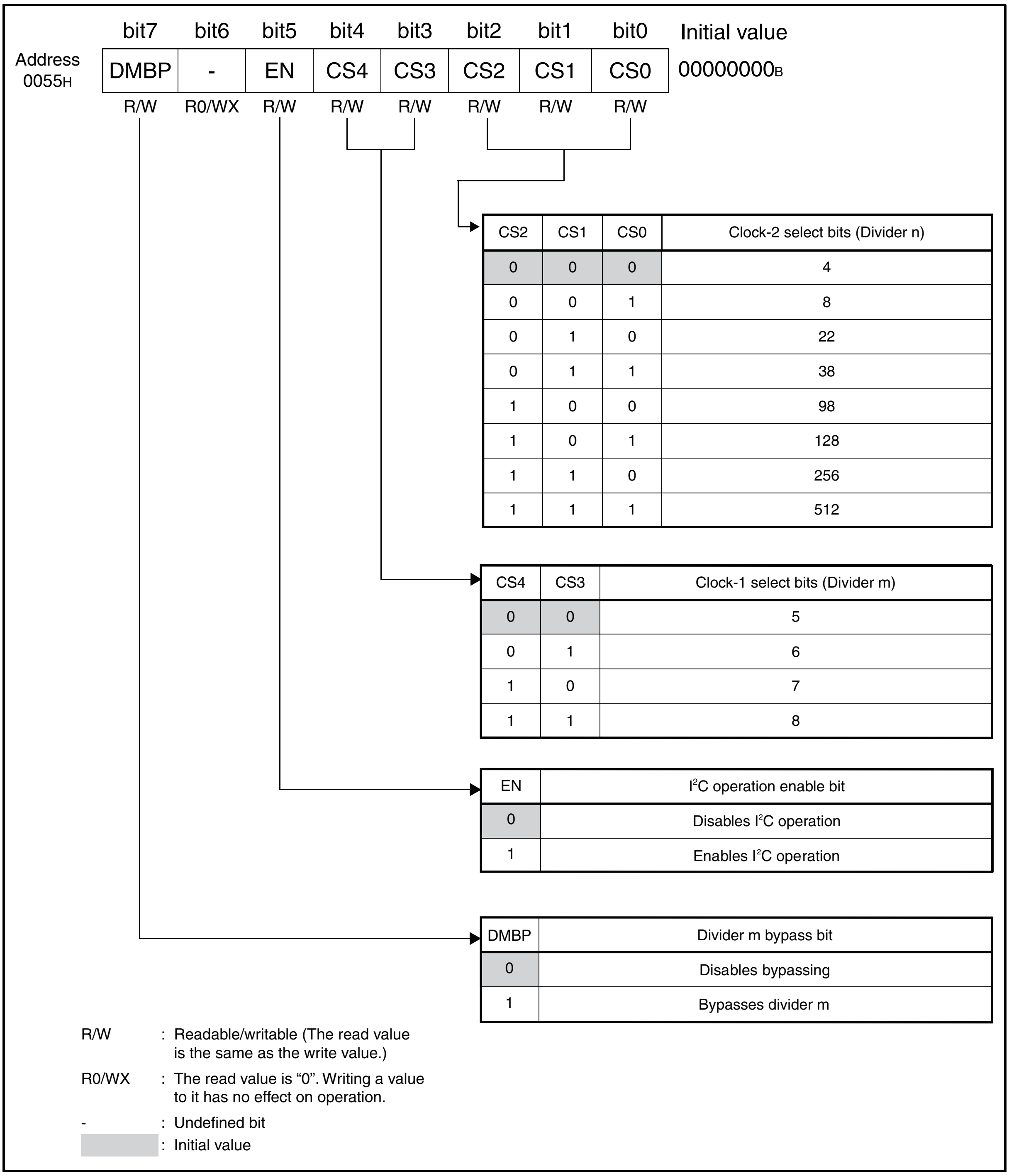
The I^{2}C address register (IAAR0) is used to set the slave address. In slave mode, address data from the master is received and then compared with the value of the IAAR0 register.

19.5.5 I^{2}C Clock Control Register (ICCR0)

The ICCR0 register is used to enable I^{2}C operation and select the shift clock frequency.

■ I^{2}C Clock Control Register (ICCR0)

Figure 19.5-7 I^{2}C Clock Control Register (ICCR0)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0055H	DMBP	-	EN	CS4	CS3	CS2	CS1	CS0	00000000B
	R/W	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

CS2	CS1	CS0	Clock-2 select bits (Divider n)
0	0	0	4
0	0	1	8
0	1	0	22
0	1	1	38
1	0	0	98
1	0	1	128
1	1	0	256
1	1	1	512

CS4	CS3	Clock-1 select bits (Divider m)
0	0	5
0	1	6
1	0	7
1	1	8

EN	I^{2}C operation enable bit
0	Disables I^{2}C operation
1	Enables I^{2}C operation

DMBP	Divider m bypass bit
0	Disables bypassing
1	Bypasses divider m

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

R0/WX : The read value is “0”. Writing a value to it has no effect on operation.

- : Undefined bit

▨ : Initial value

Table 19.5-4 Functions of Bits in I^{2}C Clock Control Register (ICCR0)

Bit name		Function
bit7	DMBP: Divider-m bypass bit	This bit is used to bypass the divider m to generate the shift clock frequency. **Writing "0":** Sets the value set in CS3 and CS4 as the divider m value (m = ICCR0:CS4, CS3). **Writing "1":** Bypasses the divider m. Note: Do not set this bit to "1" when divider n = 4 (ICCR0:CS2 to CS0 = 000_B).
bit6	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit5	EN: I^{2}C operation enable bit	• This bit enables I^{2}C interface operation. **Writing "0":** Disables operation of the I^{2}C interface and clears the following bits to "0". - AACKX, INTS, and WUE bits in the IBCR00 register - All the bits in the IBCR10 register except the BER and BEIE bits - All bits in the IBSR0 register **Writing "1":** Enables operation of the I^{2}C interface. • This bit is set to "0" in the following cases: - When "0" is written to this bit. - When IBCR10:BER is "1".
bit4, bit3	CS4, CS3: Clock-1 select bits (Divider m)	These bits set the shift clock frequency. Shift clock frequency (Fsck) is set as shown by the following equation: $\text{Fsck} = \dfrac{\phi}{(m \times n + 2)}$ ϕ represents the machine clock frequency (MCLK).
bit2 to bit0	CS2, CS1, CS0: Clock-2 select bits (Divider n)	

Note: If the standby mode wakeup function is not used, disable I^{2}C operation before switching the MCU to stop or watch mode.

19.6 I^{2}C Interrupts

The I^{2}C interface has a transfer interrupt and a stop interrupt which are triggered by the following events.

- **Transfer interrupt**
  A transfer interrupt occurs either upon completion of data transfer or when a bus error occurs.
- **Stop interrupt**
  A stop interrupt occurs upon detection of a stop condition or arbitration lost or upon access to the I^{2}C interface in stop/watch mode.

■ Transfer Interrupt

Table 19.6-1 shows the transfer interrupt control bits and I^{2}C interrupt sources.

Table 19.6-1 Transfer Interrupt Control Bits and I^{2}C Interrupt Sources

Item	End of transfer	Bus error
Interrupt request flag bit	IBCR10:INT =1	IBCR10:BER =1
Interrupt request enable bit	IBCR10:INTE =1	IBCR10:BEIE =1
Interrupt source	Data transfer complete	Bus error occurred

- Interrupt upon completion of transfer

  An interrupt request is output to the CPU upon completion of data transfer if the transfer completion interrupt request enable bit has been set to enable (IBCR10:INTE = 1). In the interrupt service routine, write "0" to the transfer completion interrupt request flag bit (IBCR10:INT) to clear the interrupt request. When data transfer is completed, the IBCR10:INT bit is set to "1" regardless of the value of the IBCR10:INTE bit.

- Interrupt in response to a bus error

  When the following conditions are met, a bus error is deemed to have occurred, and the I^{2}C interface will be stopped.

  - When a stop condition is detected in master mode.
  - When a start or stop condition is detected during transmission or reception of the first byte.
  - When a start or stop condition is detected during transmission or reception of data (excluding the start, first data, and stop bits).

In these cases, an interrupt request is output to the CPU if the bus error interrupt request enable bit has been set to enable (IBCR10:BEIE = 1). In the interrupt service routine, write "0" to the bus error interrupt request flag bit (IBCR10:BER) to clear the interrupt request. When a bus error occurs, the IBCR10:BER bit is set to "1" regardless of the value of the IBCR10:BEIE bit.

■ Stop Interrupt

Table 19.6-2 shows the stop interrupt control bits and I^{2}C interrupt sources (trigger events).

Table 19.6-2 Stop Interrupt Control Bits and I^{2}C Interrupt Sources

Item	Detection of stop condition	Detection of arbitration lost	MCU wakeup from stop/watch mode
Interrupt request flag bit	IBCR00:SPF =1	IBCR00:ALF =1	IBCR00:WUF =1
Interrupt request enable bit	IBCR00:SPE =1	IBCR00:ALE =1	IBCR00:WUE =1
Interrupt source	Stop condition detected	Arbitration lost detected	Start condition detected

- Interrupt upon detection of a stop condition

  A stop condition is considered to be valid if all of the following conditions are satisfied when the stop condition is detected.

  - The bus is busy (state which the start condition is detected).
  - IBCR10:MSS = 0
  - After transfer of one byte of data completes, including the acknowledgment.

In this case, an interrupt request is output to the CPU if the stop condition detection interrupt request enable bit has been set to enable (IBCR00:SPE =1). In the interrupt service routine, write "0" to the IBCR00:SPF bit to clear the interrupt request.

The IBCR00:SPF bit is set to "1" when a valid stop condition occurs regardless of the value of the IBCR00:SPE bit.

- Interrupt upon detection of arbitration lost

  When arbitration lost is detected, an interrupt request is output to the CPU if the arbitration lost detection interrupt request enable bit has been set to enable (IBCR00:ALE = 1). Either write "0" to the arbitration lost interrupt request flag bit (IBCR00:ALF) while the bus is idle or write "0" to the IBCR10:INT bit from the interrupt service routine while the bus is busy to clear the interrupt request.

  When arbitration lost occurs, the IBCR00:ALF bit is set to "1" regardless of the value for the IBCR00:ALE bit.

- Interrupt for MCU wakeup from stop/watch mode

  When a start condition is detected, an interrupt request is output to the CPU if the function to wake up the MCU from stop or watch mode has been enabled (IBCR00:WUE = 1).

  In the interrupt service routine, write "0" to the MCU standby mode wakeup interrupt request flag bit (IBCR00:WUF) to clear the interrupt request.

■ Register and Vector Table Addresses Related to I^{2}C Interrupts

Table 19.6-3 Register and Vector Table Addresses Related to I^{2}C Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
I^{2}C*	IRQ16	ILR4	L16	FFDA$_H$	FFDB$_H$

*: The I^{2}C shares the interrupt request number and vector table addresses mentioned in the table with 16-bit reload timer ch. 1 and MPG (write timing/compare clear).

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

19.7 Operations of I^{2}C and Setting Procedure Examples

This section describes the operations of I^{2}C.

■ Operations of I^{2}C

● I^{2}C interface

The I^{2}C interface is an eight-bit serial interface synchronized with a shift clock. It conforms to the I^{2}C bus specification defined by Philips.

● MCU standby mode wakeup function

The wakeup function wakes up the MCU upon detection of a start condition, from low power consumption mode such as stop or watch mode.

■ Setting Procedure Example

Below is an example of procedure for setting I^{2}C:

● Initialization

1) Set the port for input (a corresponding DDR register).
2) Set the interrupt level (ILR4).
3) Set the slave address (IAAR0).
4) Select the clock and enable I^{2}C operation (ICCR0).
5) Enable bus error interrupt requests (IBCR00:BEIE = 1).

● Interrupt processing

1) Arbitrary processing
2) Clear the bus error interrupt request flag (IBCR00:BER = 0).

19.7.1 I^{2}C Interface

The I^{2}C interface is an eight-bit serial interface synchronized with the shift clock. It conforms to the I^{2}C bus specification defined by Philips.

■ I^{2}C System

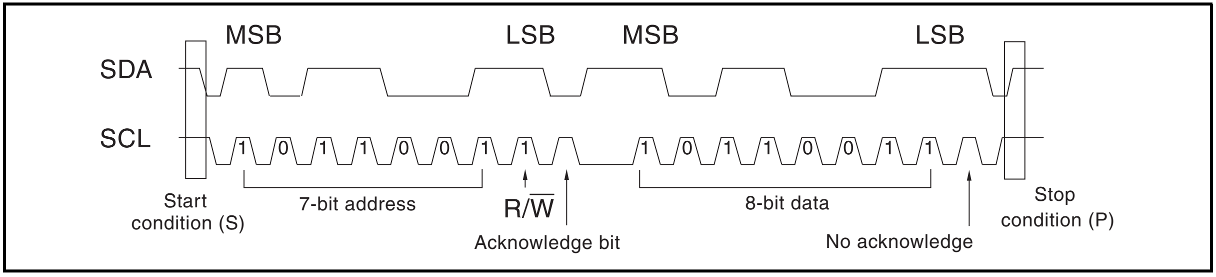
The I^{2}C bus system uses the serial data line (SDA) and serial clock line (SCL) for data transfers. All the devices connected to the bus require open drain or open collector outputs which must be connected with a pull-up resistor.

Each of the devices connected to the bus has a unique address which can be set up using software. The devices always operate in a simple master/slave relationship, where the master functions as the master transmitter or master receiver. The I^{2}C interface is a true multi-master bus with a collision detection function and arbitration function to prevent data from being lost if more than one master attempts to start data transfer at the same time.

■ I^{2}C Protocol

Figure 19.7-1 shows the format required for data transfer.

Figure 19.7-1 Data Transfer Example



The slave address is transmitted after a start condition (S) is generated. This address is seven bits followed by the data direction bit (R/$\overline{W}$) in the eighth bit position. Data is transmitted after the address. The data is eight bits followed by an acknowledgment.

Data can be transmitted continuously to the same slave address in consecutive units of eight bits plus acknowledgment.

Data transfer is always ended in the master stop condition (P). However, the repeated start condition (S) can be used to transmit the address which indicates a different slave without generating a stop condition.

■ Start Conditions

While the bus is idle (SCL and SDA are both at the logical "H" level), the master generates a start condition to start transmission. As shown in Figure 19.7-1, a start condition is triggered when the SDA line is changed from "H" to "L" while SCL = "H". This starts a new data transfer and commences master/slave operation.

A start condition can be generated in either of the following two ways.

- By writing "1" to the IBCR10:MSS bit while the I^2C bus is not in use (IBCR10:MSS = 0, IBSR0:BB = 0, IBCR10:INT = 0, and IBCR00:ALF = 0). (Next, IBSR0:BB is set to "1" to indicate that the bus is busy.)
- By writing "1" to the IBCR10:SCC bit during an interrupt while in bus master mode (IBCR10:MSS = 1, IBSR0:BB = 1, IBCR10:INT = 1, and IBCR00:ALF = 0). (This generates a repeated start condition.)

Writing "1" to the IBCR10:MSS or IBCR10:SCC bit is ignored in other than the above cases. If another system is using the bus when "1" is written to the IBCR10:MSS bit, the IBCR00:ALF bit is set to "1".

■ Addressing

● Slave addressing in master mode

In master mode, IBSR0:BB and IBSR0:TRX are set to "1" after the start condition is generated, and the slave address in the IDDR0 register is output to the bus starting with the MSB. The address data consists of eight bits: the 7-bit slave address and the data transfer direction $R/\overline{W}$ bit (bit0 of IDDR0).

The acknowledgment from the slave is received after the address data is sent. SDA goes to "L" in the ninth clock cycle and the acknowledge bit from the receiving device is received (See Figure 19.7-1). In this case, the $R/\overline{W}$ bit (IDDR0:bit0) is inverted logically and stored in the IBSR0:TRX bit as "1" if the SDA level is "L".

● Addressing in slave mode

In slave mode, after the start condition is detected, IBSR0:BB is set to "1" and IBSR0:TRX is set to "0", and the data received from the master is stored in the IDDR0 register. After the address data is received, the IDDR0 and IAAR0 registers are compared. If the addresses match, IBSR0:AAS is set to "1" and an acknowledgment is sent to the master. Next, bit0 of the receive data (bit0 of the IDDR0 register) is saved in the IBSR0:TRX bit.

■ Data Transfer

If the MCU is addressed as a slave, data can be sent or received byte by byte with the direction determined by the $R/\overline{W}$ bit sent by the master.

Each byte to be output on the SDA line is fixed at eight bits. As shown in Figure 19.7-1, the receiver sends an acknowledgment to the sender by forcing the SDA line to the stable "L" level while the acknowledge clock pulse is "H". Data is transferred at one clock pulse per bit with MSB at the head. Sending and receiving an acknowledgment is required after each byte is transferred. Accordingly, nine clock pulses are required to transfer one complete data byte.

■ Acknowledgment

An acknowledgment is sent by the receiver in the ninth clock cycle for data byte transfer by the sender based on the following conditions.

An address acknowledgment is generated in the following cases.

- The received address matches the address set in IAAR0, and the address acknowledgment is output automatically (IBCR00:AACKX = 0).
- A general call address (00_H) is received and the general call address acknowledgment output is enabled (IBCR10:GACKE = 1).

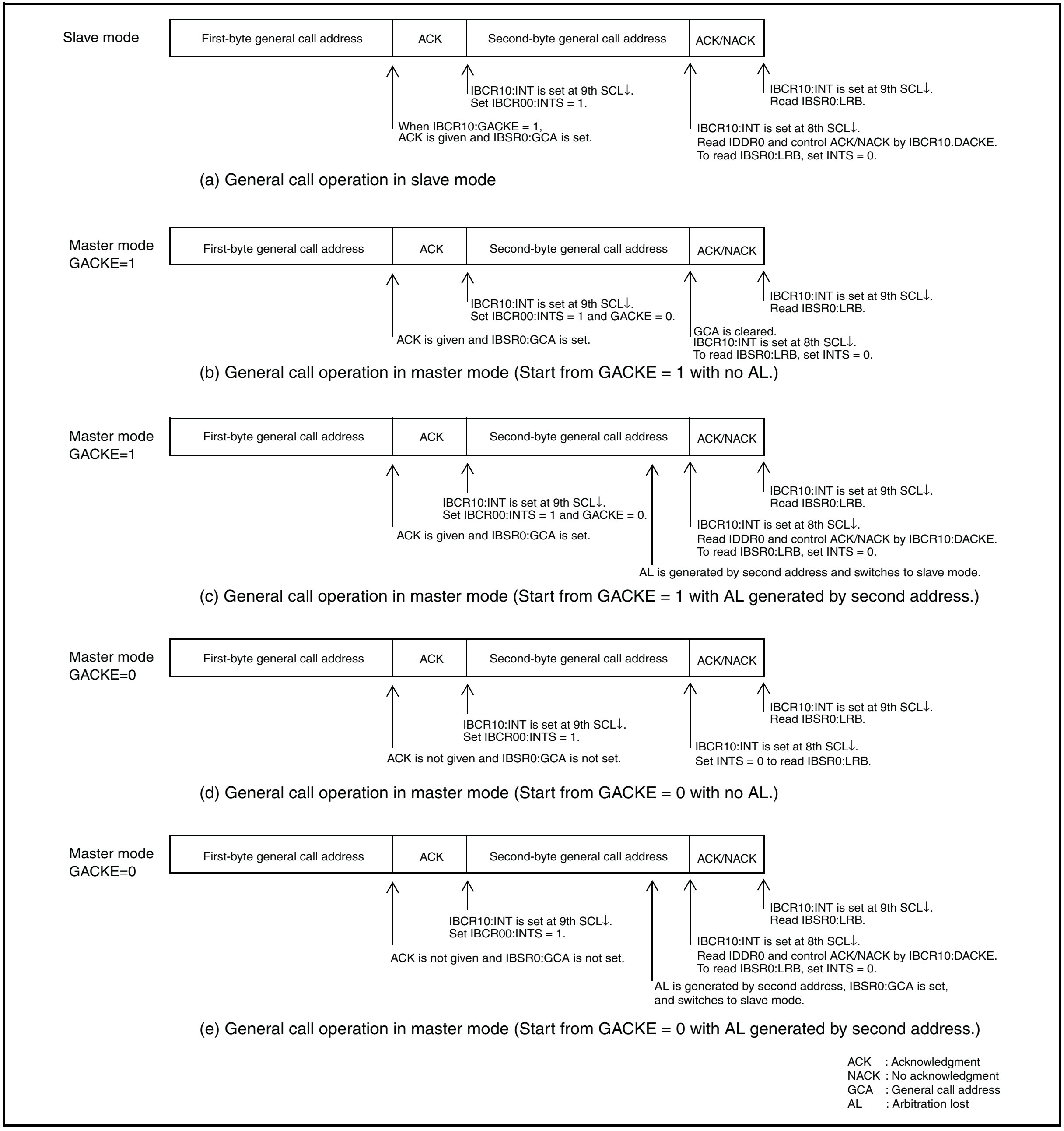
A data acknowledge bit used when data is received can be enabled or disabled by the IBCR10:DACKE bit. In master mode, a data acknowledgment is generated if IBCR10:DACKE = 1. In slave mode, a data acknowledgment is generated if an address acknowledgment has already been generated and IBCR10:DACKE = 1. The received acknowledgment is saved in IBSR0:LRB in the ninth SCL cycle.

- If the data ACK depends on the content of received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACKE) after writing "1" to the IBCR00:INTS bit (for example, by a previous transfer completion interrupt) so that the latest received data can be read.
- The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt triggered by the ninth SCL cycle). Accordingly, if ACK is read when the IBCR00:INTS bit is "1", you must write "0" to this bit in the transfer completion interrupt triggered by the eighth SCL cycle so that another transfer completion interrupt will be triggered by the ninth SCL cycle.

■ General Call Address

A general call address consists of the start address byte (00_H) and the second address byte that follows. To use a general call address, you must set IBCR10:GACKE=1 before the acknowledge of the first byte general call address. Also, the acknowledgment for the second address byte can be controlled as shown below.

Figure 19.7-2 General Call Operation


Slave mode
First-byte general call address
ACK
Second-byte general call address
ACK/NACK
IBCR10:INT is set at 9th SCL↓.
Set IBCR00:INTS = 1.
When IBCR10:GACKE = 1,
ACK is given and IBSR0:GCA is set.
IBCR10:INT is set at 9th SCL↓.
Read IBSR0:LRB.
IBCR10:INT is set at 8th SCL↓.
Read IDDR0 and control ACK/NACK by IBCR10.DACKE.
To read IBSR0:LRB, set INTS = 0.

(a) General call operation in slave mode

Master mode
GACKE=1
First-byte general call address
ACK
Second-byte general call address
ACK/NACK
IBCR10:INT is set at 9th SCL↓.
Set IBCR00:INTS = 1 and GACKE = 0.
ACK is given and IBSR0:GCA is set.
GCA is cleared.
IBCR10:INT is set at 8th SCL↓.
To read IBSR0:LRB, set INTS = 0.
IBCR10:INT is set at 9th SCL↓.
Read IBSR0:LRB.

(b) General call operation in master mode (Start from GACKE = 1 with no AL.)

Master mode
GACKE=1
First-byte general call address
ACK
Second-byte general call address
ACK/NACK
IBCR10:INT is set at 9th SCL↓.
Set IBCR00:INTS = 1 and GACKE = 0.
ACK is given and IBSR0:GCA is set.
IBCR10:INT is set at 9th SCL↓.
Read IBSR0:LRB.
IBCR10:INT is set at 8th SCL↓.
Read IDDR0 and control ACK/NACK by IBCR10:DACKE.
To read IBSR0:LRB, set INTS = 0.
AL is generated by second address and switches to slave mode.

(c) General call operation in master mode (Start from GACKE = 1 with AL generated by second address.)

Master mode
GACKE=0
First-byte general call address
ACK
Second-byte general call address
ACK/NACK
IBCR10:INT is set at 9th SCL↓.
Set IBCR00:INTS = 1.
ACK is not given and IBSR0:GCA is not set.
IBCR10:INT is set at 9th SCL↓.
Read IBSR0:LRB.
IBCR10:INT is set at 8th SCL↓.
Set INTS = 0 to read IBSR0:LRB.

(d) General call operation in master mode (Start from GACKE = 0 with no AL.)

Master mode
GACKE=0
First-byte general call address
ACK
Second-byte general call address
ACK/NACK
IBCR10:INT is set at 9th SCL↓.
Set IBCR00:INTS = 1.
ACK is not given and IBSR0:GCA is not set.
IBCR10:INT is set at 9th SCL↓.
Read IBSR0:LRB.
IBCR10:INT is set at 8th SCL↓.
Read IDDR0 and control ACK/NACK by IBCR10:DACKE.
To read IBSR0:LRB, set INTS = 0.
AL is generated by second address, IBSR0:GCA is set,
and switches to slave mode.

(e) General call operation in master mode (Start from GACKE = 0 with AL generated by second address.)

ACK : Acknowledgment
NACK : No acknowledgment
GCA : General call address
AL : Arbitration lost

If this module sends a general call address at the same time as another device, you can determine whether the module successfully seized control of the bus by checking whether arbitration lost was detected when the second address byte was transferred. If arbitration lost was detected, the module goes to slave mode and continues to receive data from the master.

■ Stop Condition

The master can release the bus and end communications by generating a stop condition. Changing the SDA line from "L" to "H" while SCL is "H" generates a stop condition. This signals to the other devices on the bus that the master has finished communications (referred to below as "bus free"). However, the master can continue to generate start conditions without generating a stop condition. This is called a repeated start condition.

Writing "0" to the IBCR10:MSS bit during an interrupt while in bus master mode (IBCR10:MSS = 1, IBSR0:BB = 1, IBCR10:INT = 1, and IBCR00:ALF = 0) generates a stop condition and changes to slave mode. In other cases, writing "0" to the IBCR10:MSS bit is ignored.

■ Arbitration

The interface circuit is a true multi-master bus able to connect multiple master devices. Arbitration occurs when another master within the system simultaneously transfers data during a master transfer.

Arbitration occurs on the SDA line while the SCL line is at the "H" level. When the send data is "1" and the data on the SDA line is "L" at the master, this is treated as arbitration lost. In this case, data output is halted and IBCR00:ALF is set to "1". If this occurs, an interrupt is generated if arbitration lost interrupts have been enabled (IBCR00:ALE = 1). If IBCR00:ALF is set to "1", the module sets IBCR10:MSS = 0 and IBSR0:TRX = 0, clears TRX, and goes to slave receive mode.

If IBCR00:ALF is set to "1" when IBSR0:BB = 0, IBCR00:ALF is cleared only by writing "0". If IBCR00:ALF is set to "1" when IBSR0:BB = 1, IBCR00:ALF is cleared only by clearing IBCR10:INT to "0".

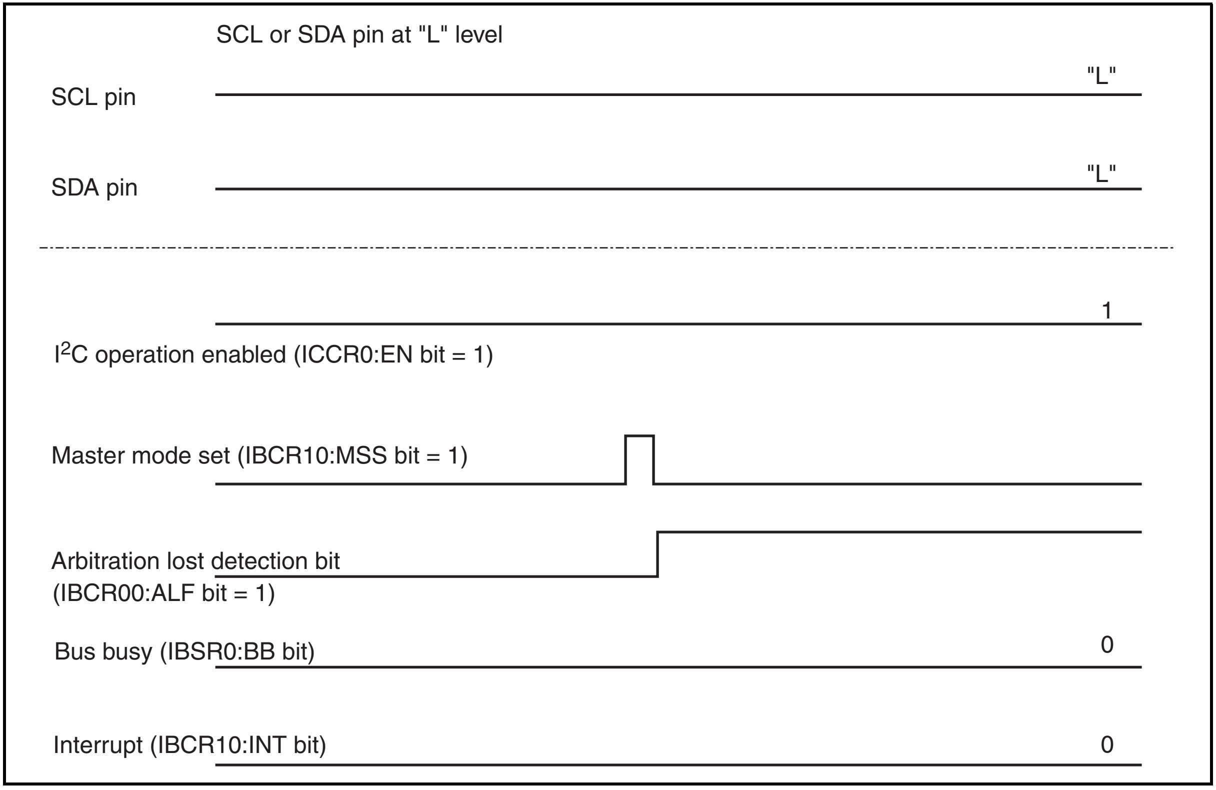
● Conditions for generating an arbitration lost interrupt when IBSR0:BB = 0

When a start condition is generated by the program (by setting the IBCR10:MSS bit to "1") at the timing shown in Figure 19.7-3 or Figure 19.7-4, interrupt generation (IBCR10:INT bit = 1) is prohibited by arbitration lost detection (IBCR00:ALF = 1).

- Conditions (1) in which no interrupt is generated due to arbitration lost

If the program triggers a start condition (by setting the IBCR10:MSS bit to "1") when no start condition has been detected (IBSR0:BB bit = 0) and the SDA and SCL line pins are at the "L" level.

Figure 19.7-3 Timing Diagram with No Interrupt Generated with IBCR00:ALF = 1

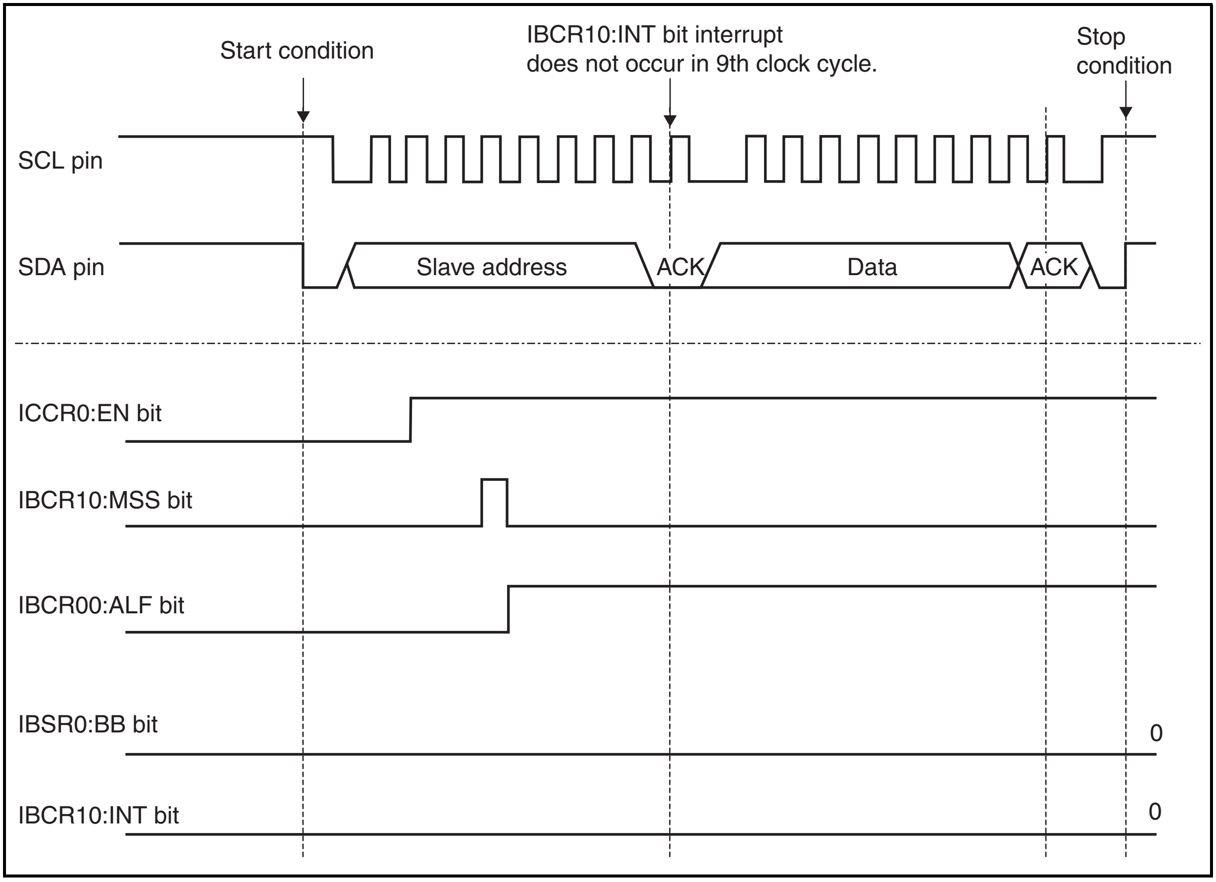

SCL or SDA pin at "L" level
SCL pin
"L"
SDA pin
"L"
1
I2C operation enabled (ICCR0:EN bit = 1)
Master mode set (IBCR10:MSS bit = 1)
Arbitration lost detection bit
(IBCR00:ALF bit = 1)
Bus busy (IBSR0:BB bit)
0
Interrupt (IBCR10:INT bit)
0

- Conditions (2) in which no interrupt is generated due to arbitration lost

If the program enables I^{2}C operation (by setting the ICCR0:EN bit to "1") and triggers a start condition (by setting the IBCR10:MSS bit to "1") when the I^{2}C bus is in use by another master.

This is because, as shown in Figure 19.7-4, this I^{2}C module cannot detect the start condition (IBSR0:BB bit= 0) if another master starts communications on the I^{2}C bus when the operation of this I^{2}C module has been disabled (ICCR0:EN bit = 0).

Figure 19.7-4 Timing Diagram with No Interrupt Generated with IBCR00:ALF = 1



If this situation can occur, use the following procedure to set up the module from the software.

1) Trigger a start condition from the program (by setting the IBCR10:MSS bit to "1").

2) Check the IBCR00:ALF and IBSR0:BB bits in the arbitration lost interrupt.

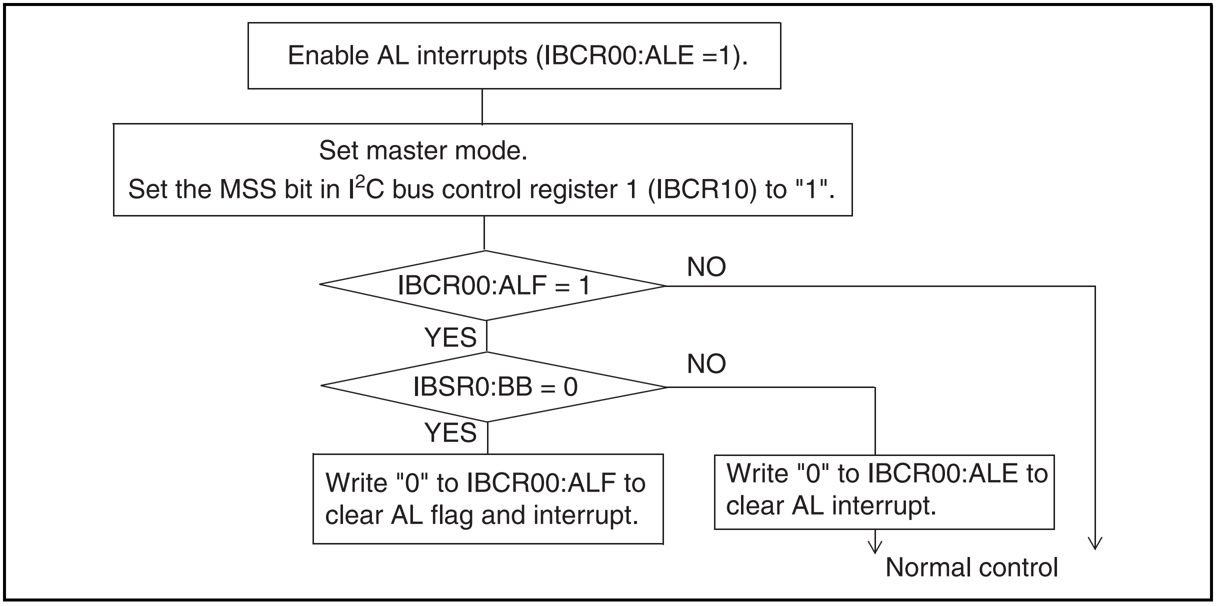
If IBCR00:ALF = 1 and IBSR0:BB = 0, clear the IBCR00:ALF bit to "0".

If IBCR00:ALF = 1 and IBSR0:BB = 1, clear the IBCR00:ALE bit to "0" and perform control as normal. (Normal control means writing "0" to the IBCR00:INT bit in the INT interrupt to clear IBCR00:ALF.)

In other cases, perform control as normal (Normal control means writing "0" to the IBCR00:INT bit in the INT interrupt to clear IBCR00:ALF.)

The following sample flow chart illustrates the procedure:

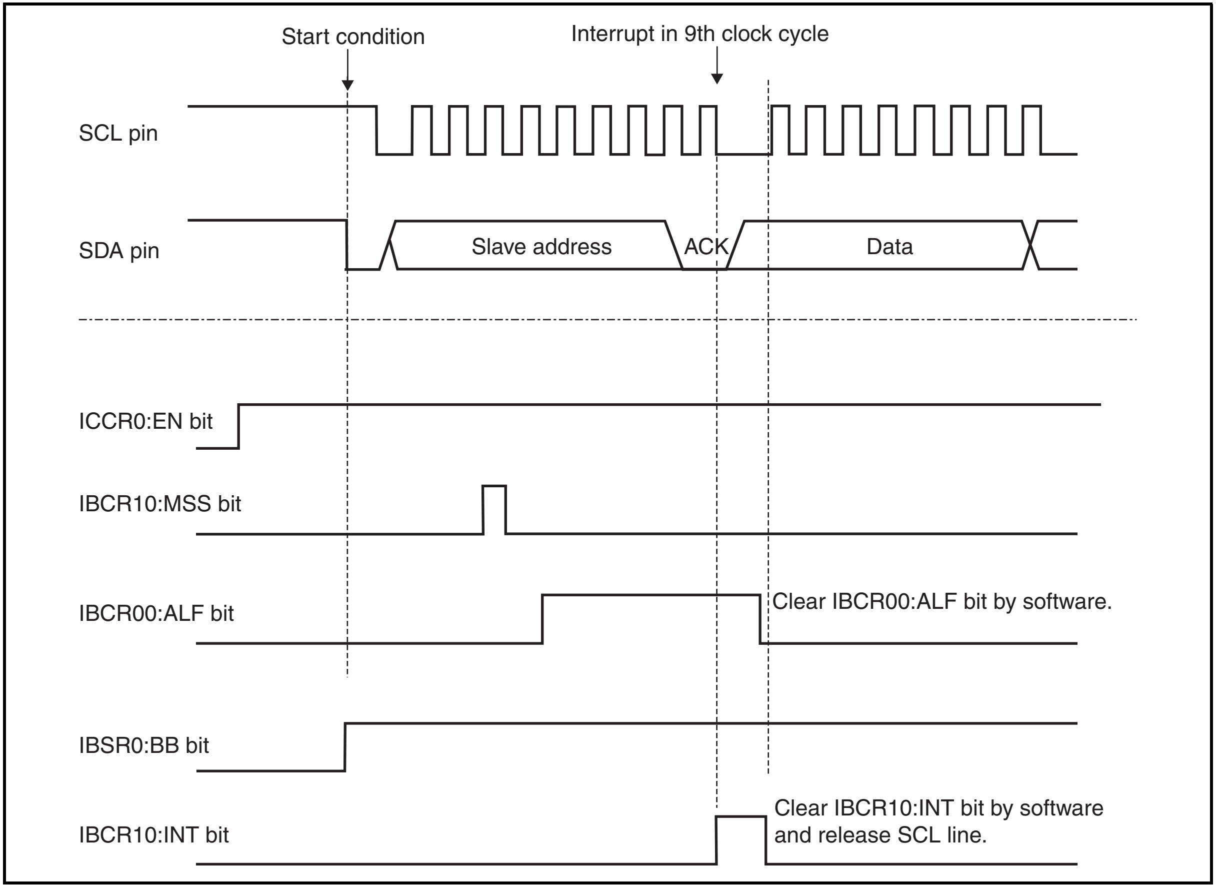
Figure 19.7-5 Sample Flow Chart 1


Enable AL interrupts (IBCR00:ALE =1).
Set master mode.
Set the MSS bit in I2C bus control register 1 (IBCR10) to "1".
IBCR00:ALF = 1
NO
YES
IBSR0:BB = 0
NO
YES
Write "0" to IBCR00:ALF to clear AL flag and interrupt.
Write "0" to IBCR00:ALE to clear AL interrupt.
Normal control

- Example of generating an interrupt (IBCR10:INT bit = 1) with "IBCR00:ALF bit = 1" detected

If a start condition is generated by the program (by setting the IBCR10:MSS bit to "1") with the bus busy (IBSR0:BB bit = 1) and arbitration lost detected, a IBCR10:INT bit interrupt occurs upon detection of "IBCR00:ALF bit = 1".

Figure 19.7-6 Timing Diagram with Interrupt Generated with "IBCR00:ALF Bit = 1" Detected


Start condition
Interrupt in 9th clock cycle
SCL pin
SDA pin
Slave address
ACK
Data
ICCR0:EN bit
IBCR10:MSS bit
IBCR00:ALF bit
Clear IBCR00:ALF bit by software.
IBSR0:BB bit
IBCR10:INT bit
Clear IBCR10:INT bit by software and release SCL line.

19.7.2 Function to Wake up the MCU from Standby Mode

The wakeup function enables the I2C macro to be accessed while the MCU is in stop or watch mode.

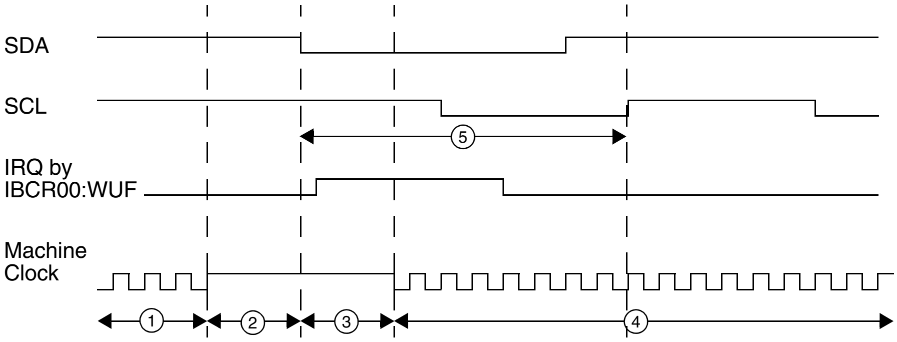
■ Function to Wake Up the MCU from Standby Mode

The I2C macro includes a function to wake up the MCU from standby mode. The function is enabled by writing "1" to the IBCR00:WUE bit.

When the MCU is in stop/watch mode with the IBCR00:WUE bit containing "1", if a start condition is detected on the I2C bus, the wakeup interrupt request flag bit (IBCR00:WUF) is set to "1" and the wakeup interrupt request is generated to wake up the MCU from stop/watch mode.

- Set IBCR00:WUE to "1" immediately prior to setting the MCU to stop or watch mode. Similarly, clear IBCR00:WUE (by writing "0") after the MCU wakes up from stop or watch mode so that I2C operation can restart as soon as possible.
- The wakeup function only applies to the MCU stop and watch modes.

Figure 19.7-7 Comparison of Normal I2C Operation and Wakeup Operation



① Set the IBCR00:WUE bit to "1" immediately before entering stop/watch mode and make sure that IBSR0:BB = 0.

② Set the MCU to stop/watch mode and the machine clock stops.

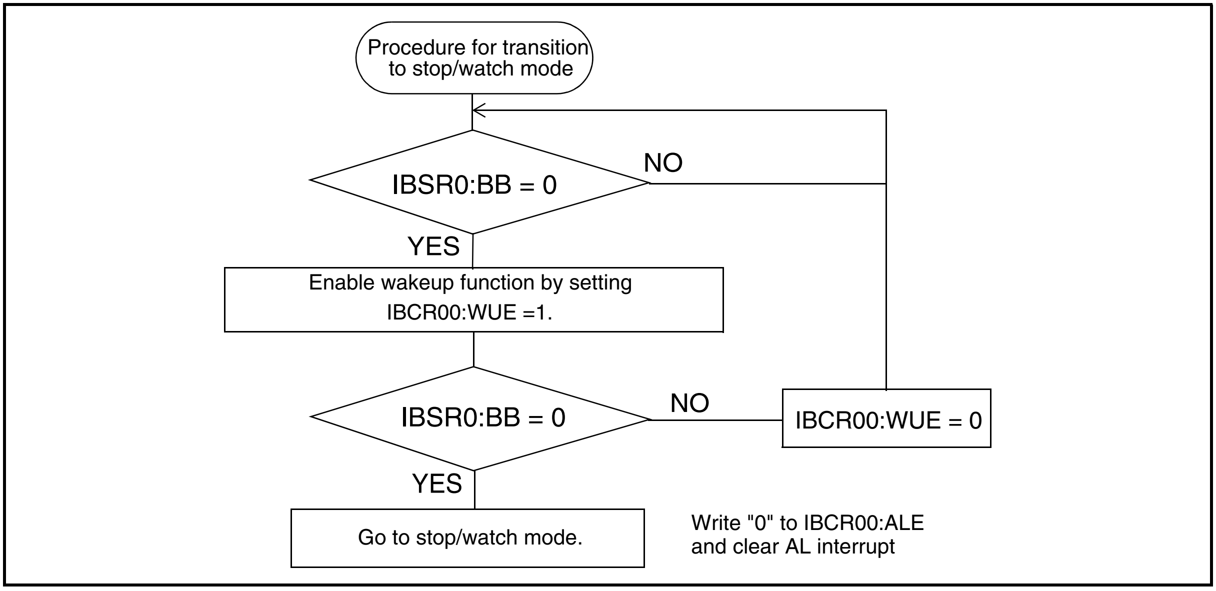
③ Detect a start condition in stop/watch mode. IBCR00:WUF is set to 1 and a wakeup IRQ is generated. After the oscillation stabilization wait time, the MCU wakes up and enters main clock mode.

④ Clear the IBCR00:WUE bit to "0" so that I2C can restart the normal operation, and clear the IBCR00:WUF bit to "0" to clear the wakeup interrupt.

⑤ To receive the data byte correctly, the SCL must be released in the first cycle after 100 µs (assuming a minimum oscillation stabilization wait time of 100 µs) from the start of I2C transmission (falling edge detection of SDA).

The following sample flow chart illustrates the wakeup function.

Figure 19.7-8 Sample Flow Chart 2


Procedure for transition to stop/watch mode
IBSR0:BB = 0
NO
YES
Enable wakeup function by setting IBCR00:WUE =1.
IBSR0:BB = 0
NO
IBCR00:WUE = 0
YES
Go to stop/watch mode.
Write "0" to IBCR00:ALE and clear AL interrupt

19.8 Notes on Using I^{2}C

This section provides notes on using I^{2}C.

■ Notes on Using I^{2}C

● Notes on setting I^{2}C interface registers

- Operation of the I^{2}C interface must be enabled (ICCR0:EN) before setting the I^{2}C bus control registers (IBCR00 and IBCR10).
- Setting the master/slave select bit (IBCR10:MSS) (by writing "1") starts data transfer.

● Notes on setting the shift clock frequency

- The shift clock frequency can be calculated by determining the m, n, and DMBP values using the Fsck equation in Table 19.5-4.
- "DMBP=1" may not be selected if the value of n is 4 (ICCR0:CS2 = CS1 = CS = 0).

● Notes on priority for simultaneous writes

- Contention between next byte transfer and stop condition
  When "0" is written to IBCR10:MSS with IBCR10:INT cleared, the MSS bit takes priority and a stop condition develops.
- Contention between next byte transfer and start condition
  When "1" is written to IBCR10:SCC with IBCR10:INT cleared, the SCC bit takes priority and a start condition develops.

● Notes on setting up using software

- Do not select a repeated start condition (IBCR10:SCC=1) and slave mode (IBCR10:MSS=0) simultaneously.
- Execution cannot return from interrupt processing if the interrupt request enable bit is enabled (IBCR10:BEIE=1/IBCR10:INTE=1) with the interrupt request flag bit (IBCR10:BER/IBCR10:INT) containing "1". Be sure to clear the IBCR10:BER/IBCR10:INT bit.
- The following bits are cleared to "0" when I^{2}C operation is disabled (ICCR0:EN=0):
  - AACKX, INTS, and WUE bits in the IBCR00 register
  - All the bits in the IBCR10 register except the BER and BEIE bits
  - All bits in the IBSR0 register

● Notes on data acknowledgment

In slave mode, a data acknowledgment is generated in either of the following cases:

- When the received address matches the value in the address register (IAAR0) and IBCR00:AACKX = 0.
- When a general call address (00_H) is received and IBCR10:GACKE = 1.

● Notes on selecting the transfer complete timing

- The transfer complete timing select bit (IBCR00:INTS) is valid only during data reception

(IBSR0:TRX = 0 and IBSR0:FBT = 0).

- In cases other than data reception (IBSR0:TRX = 1 or IBSR0:FBT = 1), the transfer completion interrupt (IBCR10:INT) is always generated in the ninth SCL cycle.
- If the data ACK depends on the content of the received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACKE) after writing "1" to the IBCR00:INTS bit (for example, using a previous transfer completion interrupt) to read latest received data.
- The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt in the ninth SCL cycle.) If ACK is read when the IBCR0:INTS bit is "1", therefore, you must write "0" to the IBCR00:INTS bit in the transfer completion interrupt in the eighth SCL cycle so that another transfer completion interrupt will occur in the ninth SCL cycle.

● Notes on using the MCU standby mode wakeup function

- Set IBCR00:WUE to "1" immediately prior to setting the MCU to stop or watch mode. Similarly, clear IBCR00:WUE (by writing "0") after the MCU wakes up from stop or watch mode so that I^{2}C operation can restart as soon as possible.
- When a wakeup interrupt request occurs, the MCU wakes up after the oscillation stabilization wait time elapses. To prevent the data loss immediately after wakeup, design the system so that the SCL rises as the first cycle and the first bit must be transmitted as data after 100 µs (assuming a minimum oscillation stabilization wait time of 100 µs) from the wakeup due to start of I^{2}C transmission (upon detection of the falling edge of SDA).
- During a MCU standby mode, the status flags, state machine, and I^{2}C bus outputs for the I^{2}C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I^{2}C bus system, make sure that IBSR0:BB = 0 before entering standby mode.
- The wakeup function does not support the transition of the MCU to stop or watch mode with IBSR0:BB = 1. If the MCU enters stop or watch mode with IBSR0:BB = 1, a bus error will occur upon detection of a start condition.
- To ensure correct operation of the I^{2}C interface, always clear IBCR00:WUE to "0" after the MCU wakes up from stop or watch mode, regardless of whether this occurs due to the I^{2}C wakeup function or the wakeup function for some other resource (such as an external interrupt).

19.9 Sample Settings for I^{2}C

This section provides sample settings for the I^{2}C interface.

■ Sample Settings

● Enabling/disabling I^{2}C operation

Use the I^{2}C operation enable bit (ICCR0:EN).

Operation	I^{2}C operation enable bit (EN)
To disable I^{2}C operation	Set the bit to "0".
To enable I^{2}C operation	Set the bit to "1".

● Selecting the I^{2}C master or slave mode

Use the master/slave select bit (IBCR10:MSS).

Operation	Master/slave select bit (MSS)
To select master mode	Set the bit to "1".
To select slave mode	Set the bit to "0".

● Selecting the shift clock

Use the clock select bits (ICCR0:CS4/CS3/CS2/CS1/CS0).

● Bypassing the divider-m when the shift clock frequency is generated

Use the divider-m bypass bit (ICCR0:DMBP).

Operation	Divider-m bypass bit (DMBP)
To bypass divider m	Set the bit to "1".

● Controlling I^{2}C address acknowledgment

Use the address acknowledge disable bit (IBCR00:AACKX).

Operation	Address acknowledge disable bit (AACKX)
To enable address acknowledge output	Set the bit to "0".
To disable address acknowledge output	Set the bit to "1".

● Controlling I^{2}C data acknowledgment

Use the data acknowledge enable bit (IBCR10:DACKE).

Operation	Data acknowledge enable bit (DACKE)
To enable data acknowledge output	Set the bit to "1".
To disable data acknowledge output	Set the bit to "0".

● Controlling I^{2}C general call address acknowledgment

Use the general call address acknowledge enable bit (IBCR10:GACKE).

Operation	General call address acknowledge enable bit (GACKE)
To enable general call address acknowledge output	Set the bit to "1".
To disable general call address acknowledge output	Set the bit to "0".

● Restarting I^{2}C communication

Use the start condition generation bit (IBCR10:SCC).

Operation	Start condition generation bit (SCC)
To restart communication	Set the bit to "1".

● Selecting the I^{2}C data reception transfer completion flag (INT)

Use the timing select bit (IBCR00:INTS) for the data reception transfer completion flag (INT).

Operation	Timing select bit (INTS) for data reception transfer completion flag (INT)
To generate a transfer interrupt in the 9th SCL cycle	Set the bit to "0".
To generate a transfer interrupt in the 8th SCL cycle	Set the bit to "1"

● Interrupt related register

To set the interrupt level, use the following interrupt level setting register.

Interrupt source	Interrupt level setting register	Interrupt vector
ch. 0	Interrupt level register (ILR4) Address: 0007D$_H$	#16 Address: 0FFDA$_H$

● Enabling, disabling, and clearing interrupts

- Transfer interrupt
  (Data transfer completion interrupt)
  To enable interrupts, use the interrupt request enable bit (IBCR10:INTE).

Operation	Interrupt request enable bit (INTE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear an interrupt request, use the interrupt request flag (IBCR10:INT).

Operation	Interrupt request flag (INT)
To clear an interrupt request	Set the bit to "0".

(Bus error generation interrupt)

To enable interrupts, use the interrupt request enable bit (IBCR10:BEIE).

Operation	Interrupt request enable bit (BEIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear an interrupt request, use the interrupt request flag (IBCR10:BER).

Operation	Interrupt request flag (BER)
To clear an interrupt request	Set the bit to "0".

- Stop interrupt
  (Stop condition detection interrupt)
  To enable interrupts, use the interrupt request enable bit (IBCR00:SPE).

Operation	Interrupt request enable bit (SPE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear an interrupt request, use the interrupt request flag (IBCR00:SPF).

Operation	Interrupt request flag (SPF)
To clear an interrupt request	Set the bit to "0".

(Arbitration lost detection interrupt)

To enable interrupts, use the interrupt request enable bit (IBCR00:ALE).

Operation	Interrupt request enable bit (ALE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear an interrupt request, use the interrupt request flag (IBCR00:ALF).

Operation	Interrupt request flag (ALF)
To clear an interrupt request	Set the bit to "0".

(Start condition detection interrupt)

To enable interrupts, use the interrupt request enable bit (IBCR00:WUE).

Operation	Interrupt request enable bit (WUE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear an interrupt request, use the interrupt request flag (IBCR00:WUF).

Operation	Interrupt request flag (WUF)
To clear an interrupt request	Set the bit to "0".

CHAPTER 20

8/10-BIT A/D CONVERTER

This chapter describes the functions and operations of the 8/10-bit A/D converter.

20.1 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 an 8-bit or 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
- Continuous activation using the external pin ADTG

20.2 Configuration of 8/10-bit A/D Converter

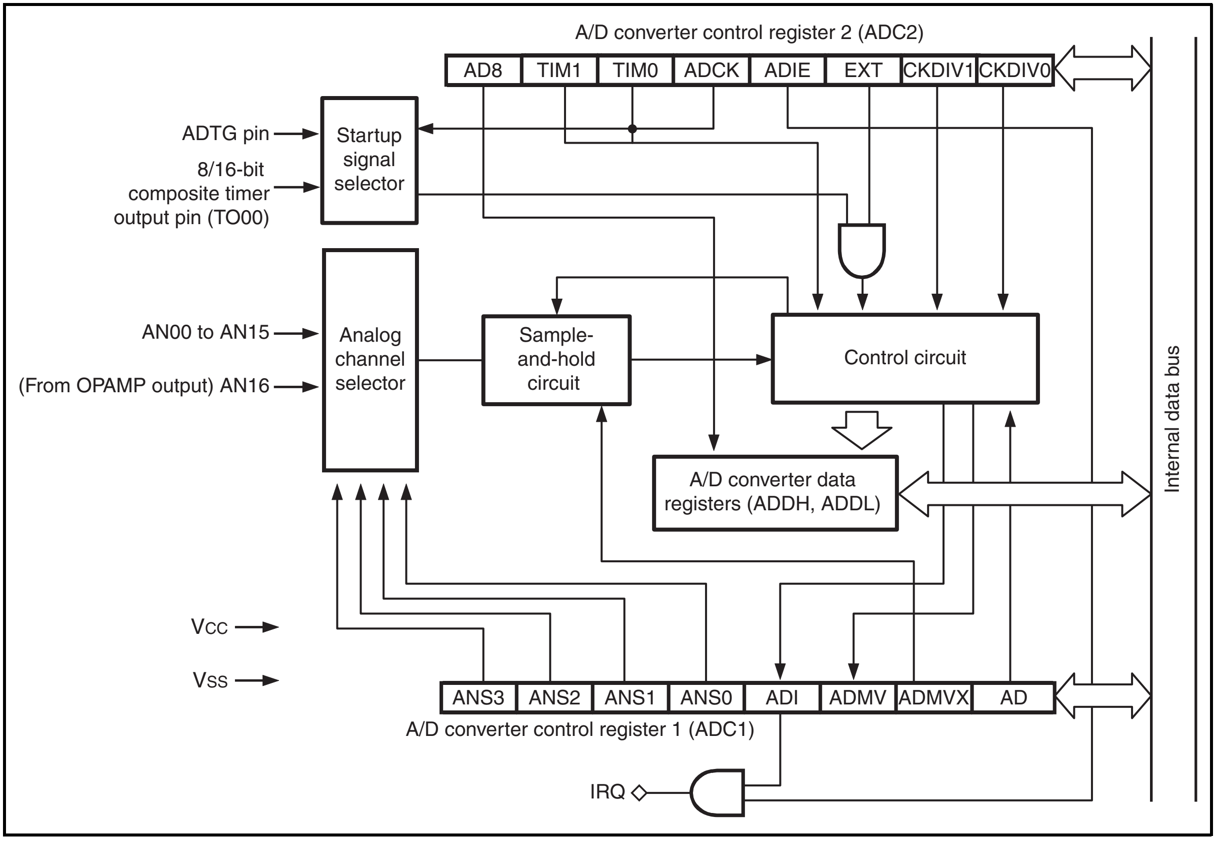
The 8/10-bit A/D converter consists of the following blocks:

- **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 20.2-1 is the block diagram of the 8/10-bit A/D converter.

Figure 20.2-1 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 conversion result 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 conversion result 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 control 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).

20.3 Pins of 8/10-bit A/D Converter

This section describes the pins of the 8/10-bit A/D converter.

■ Pins of 8/10-bit A/D Converter

The MB95430H Series has 16 channels of analog input pin.

The analog input pins are also used as general-purpose I/O ports.

● AN15 pin to AN00 pin

AN15 to AN00: When using the A/D conversion function, input to one of these pins the analog voltage to be converted. A pin of AN15 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 OPAMP output can be used as the AN16 pin of the A/D converter by modifying the setting of the OPADSEL bit in the SYSC2 register. For details, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

● ADTG pin

ADTG: This pin is used to activate the A/D conversion function by an external trigger. To use ADTG as an external trigger pin for the A/D conversion function, set it as an input port by using bit1 in the port G direction register (PDRG:bit1).

■ Block Diagram of Pins of 8/10-bit A/D Converter

Figure 20.3-1 Block Diagram of Pins AN00 and AN07 (P00/INT00/AN00 and P071/INT07/AN07/EC0) of 8/10-bit A/D Converter

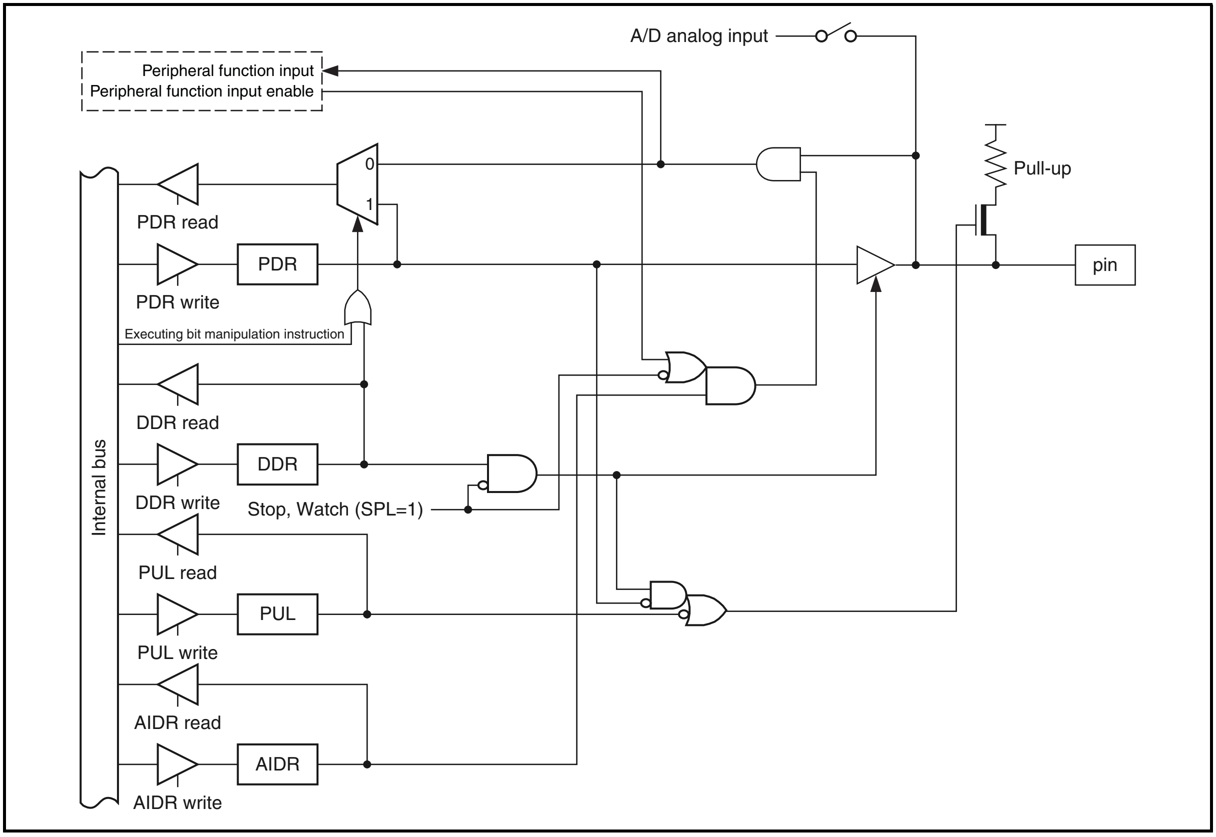


Figure 20.3-2 Block Diagram of Pins AN03 and AN04 (P03/INT03/AN03/UO and P04/INT04/AN04/UI) of 8/10-bit A/D Converter

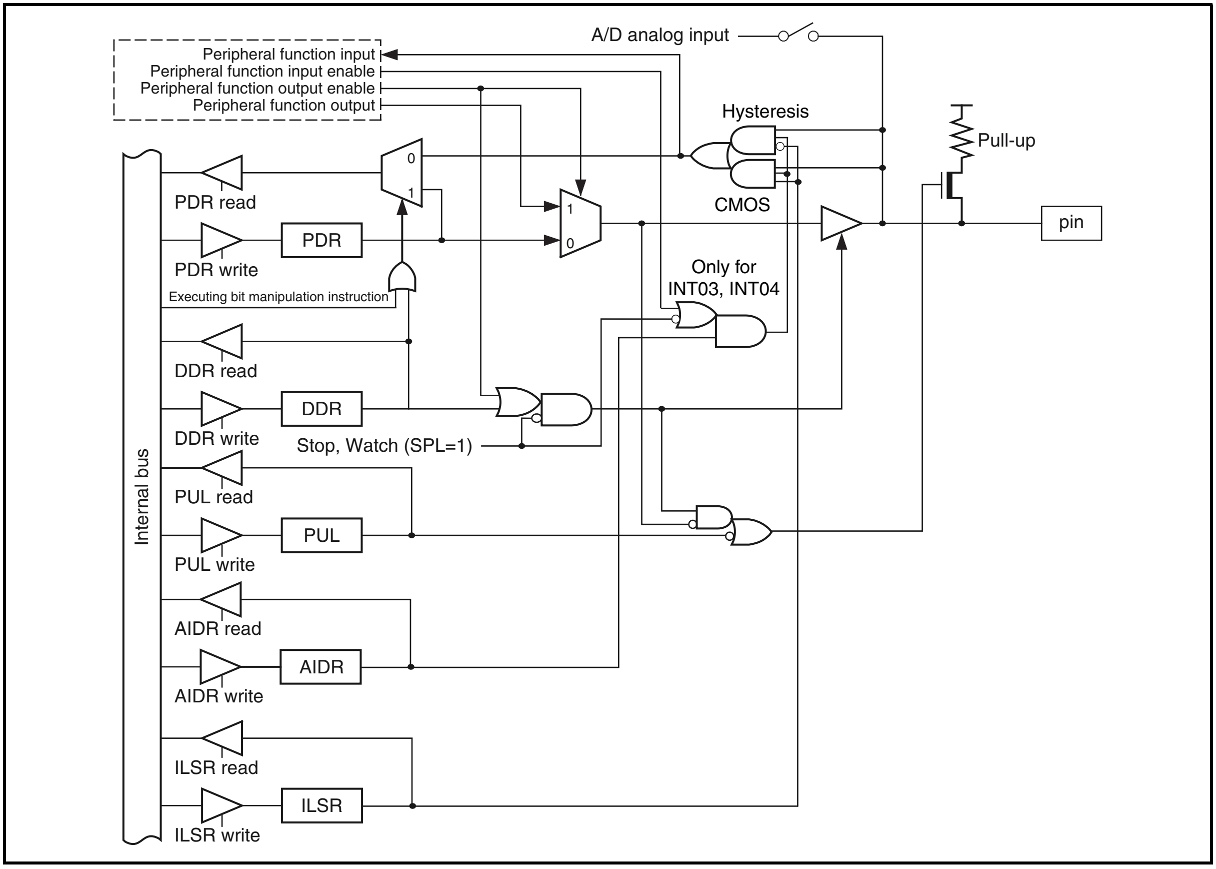


Figure 20.3-3 Block Diagram of Pins AN01, AN02, AN05 and AN06 (P01/INT01/AN01/BZ, P02/INT02/AN02/UCK, P05/INT05/AN05/TO0 and P06/INT06/AN06/TO1) of 8/10-bit A/D Converter

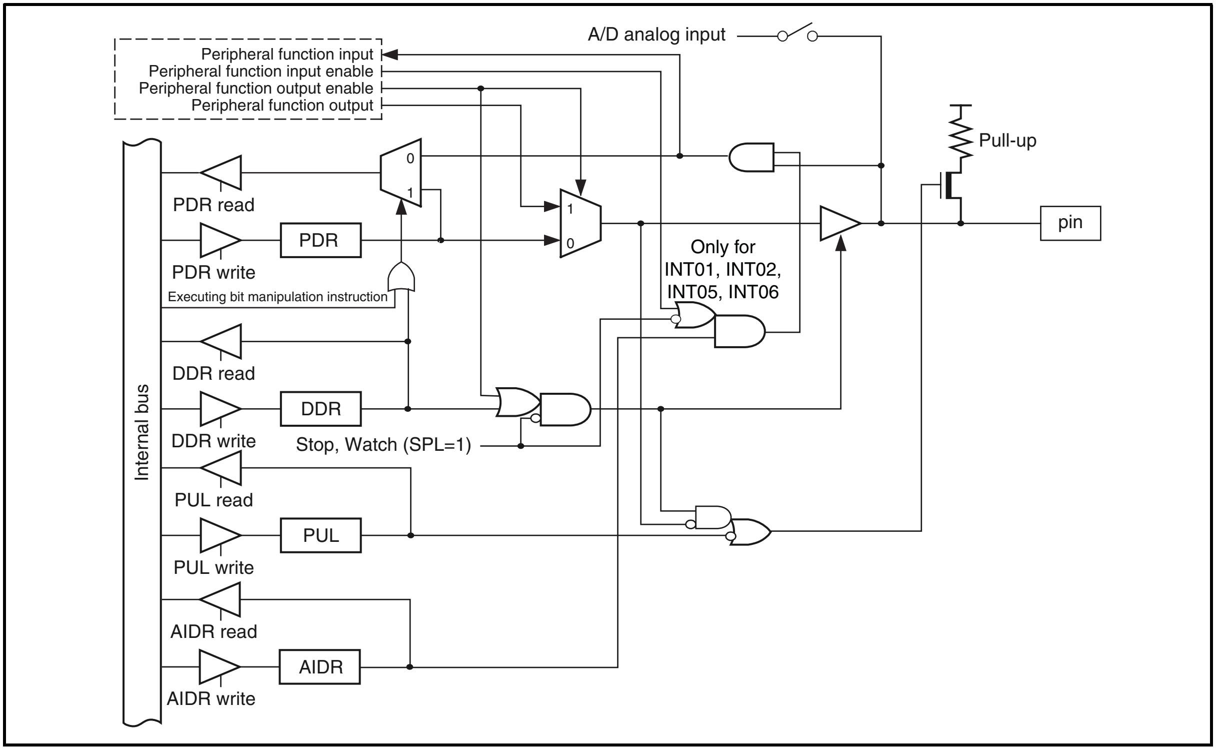


Figure 20.3-4 Block Diagram of Pins AN08, AN09, AN10, AN11, AN12, AN13, AN14 and AN15 (P71/CMP0_P/AN08, P72/CMP0_N/AN09, P74/CMP1_P/AN10, P75/CMP1_N/AN11, P63/CMP2_P/AN12, P64/CMP2_N/AN13, P66/CMP3_P/AN14 and P67/CMP3_N/AN15) of 8/10-bit A/D Converter

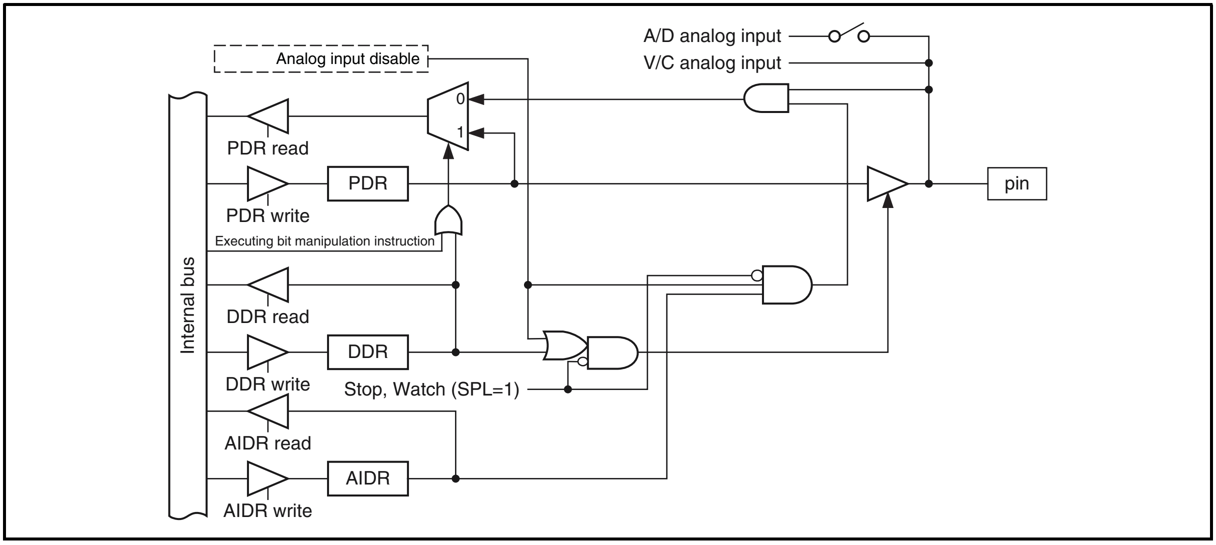
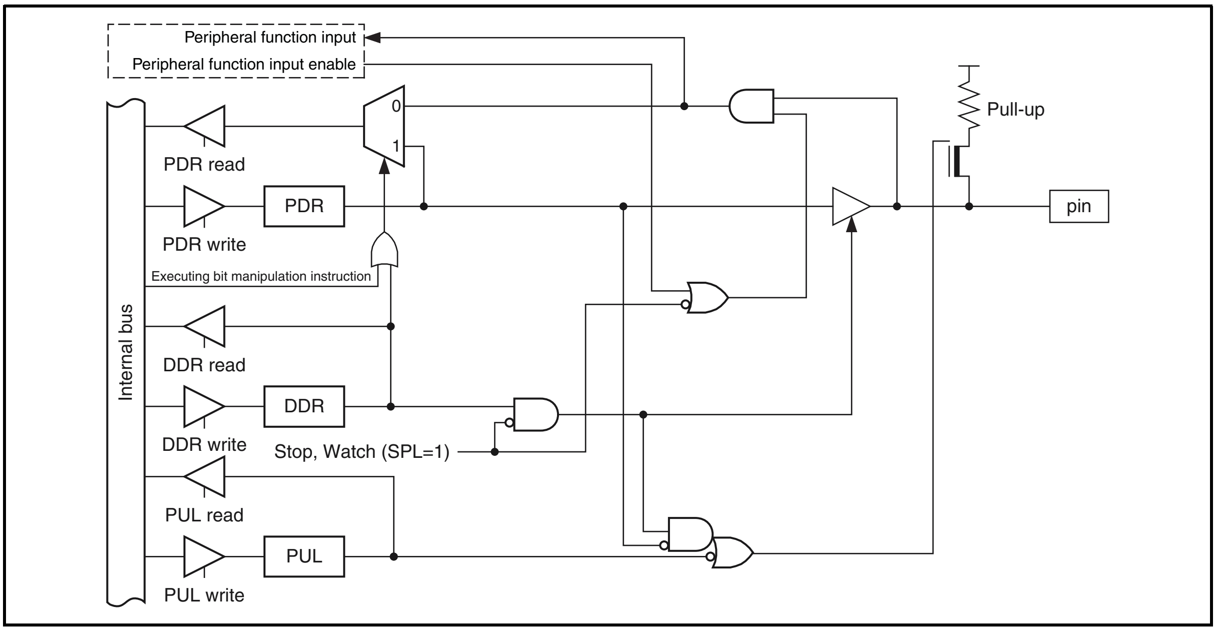


Figure 20.3-5 Block Diagram of Pin ADTG (PG1/TRG0/ADTG/OUT0/BZ/X0A) of 8/10-bit A/D Converter



20.4 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 20.4-1 lists the registers of the 8/10-bit A/D converter.

Figure 20.4-1 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	

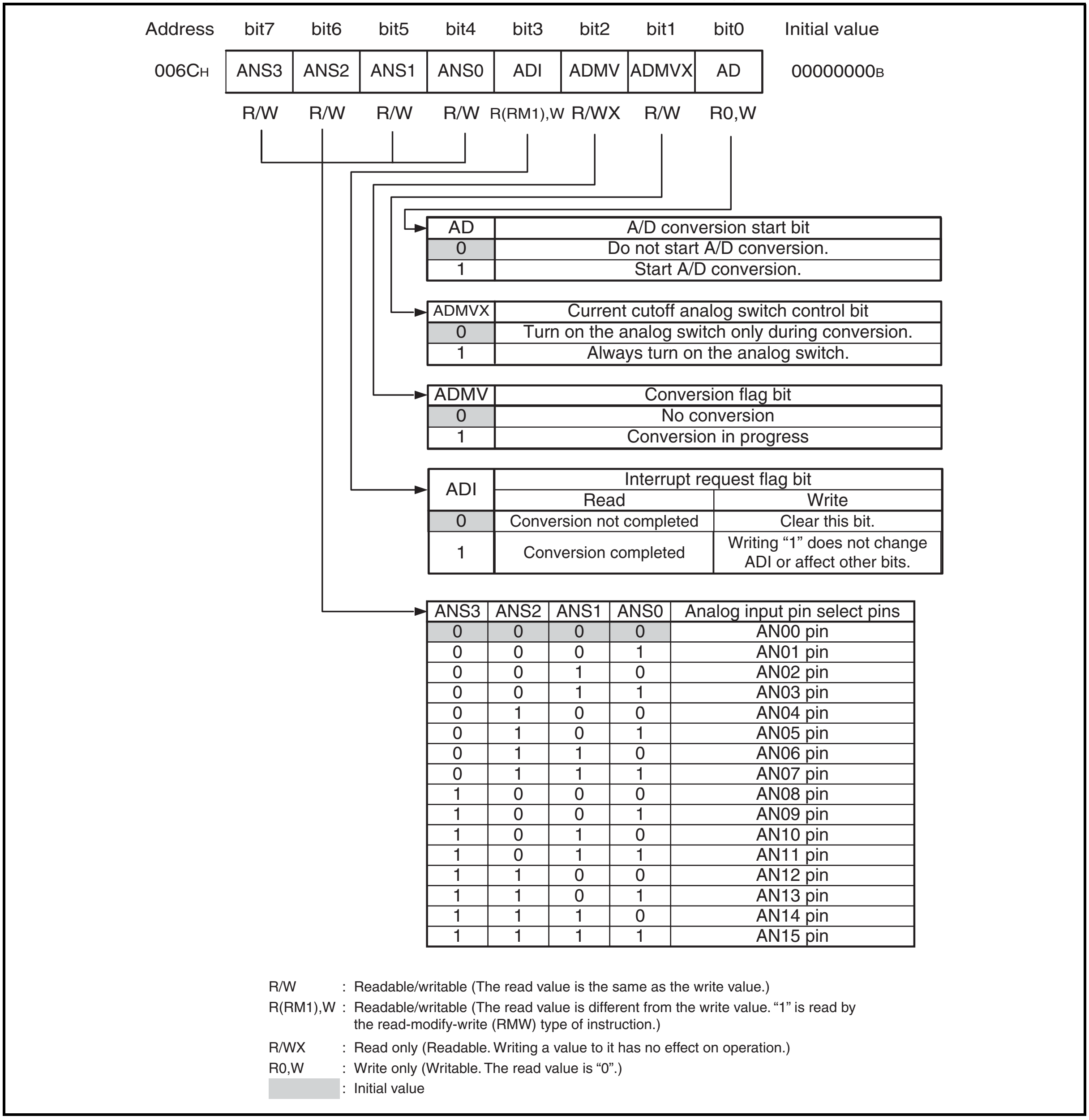
R/W	: Readable/writable (The read value is the same as the write value.)
R(RM1), W	: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/WX	: Read only (Readable. Writing a value to it has no effect on operation.)
R0,W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

20.4.1 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 20.4-2 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.

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 pins
0	0	0	0	AN00 pin
0	0	0	1	AN01 pin
0	0	1	0	AN02 pin
0	0	1	1	AN03 pin
0	1	0	0	AN04 pin
0	1	0	1	AN05 pin
0	1	1	0	AN06 pin
0	1	1	1	AN07 pin
1	0	0	0	AN08 pin
1	0	0	1	AN09 pin
1	0	1	0	AN10 pin
1	0	1	1	AN11 pin
1	1	0	0	AN12 pin
1	1	0	1	AN13 pin
1	1	1	0	AN14 pin
1	1	1	1	AN15 pin

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.)
R0,W : Write only (Writable. The read value is "0".)
▨ : Initial value

Do not select an unusable pin for the MB95430H Series with the analog input pin select bits (ANS3 to ANS0).

Table 20.4-1 Functions of Bits in 8/10-bit A/D Converter Control Register 1 (ADC1)

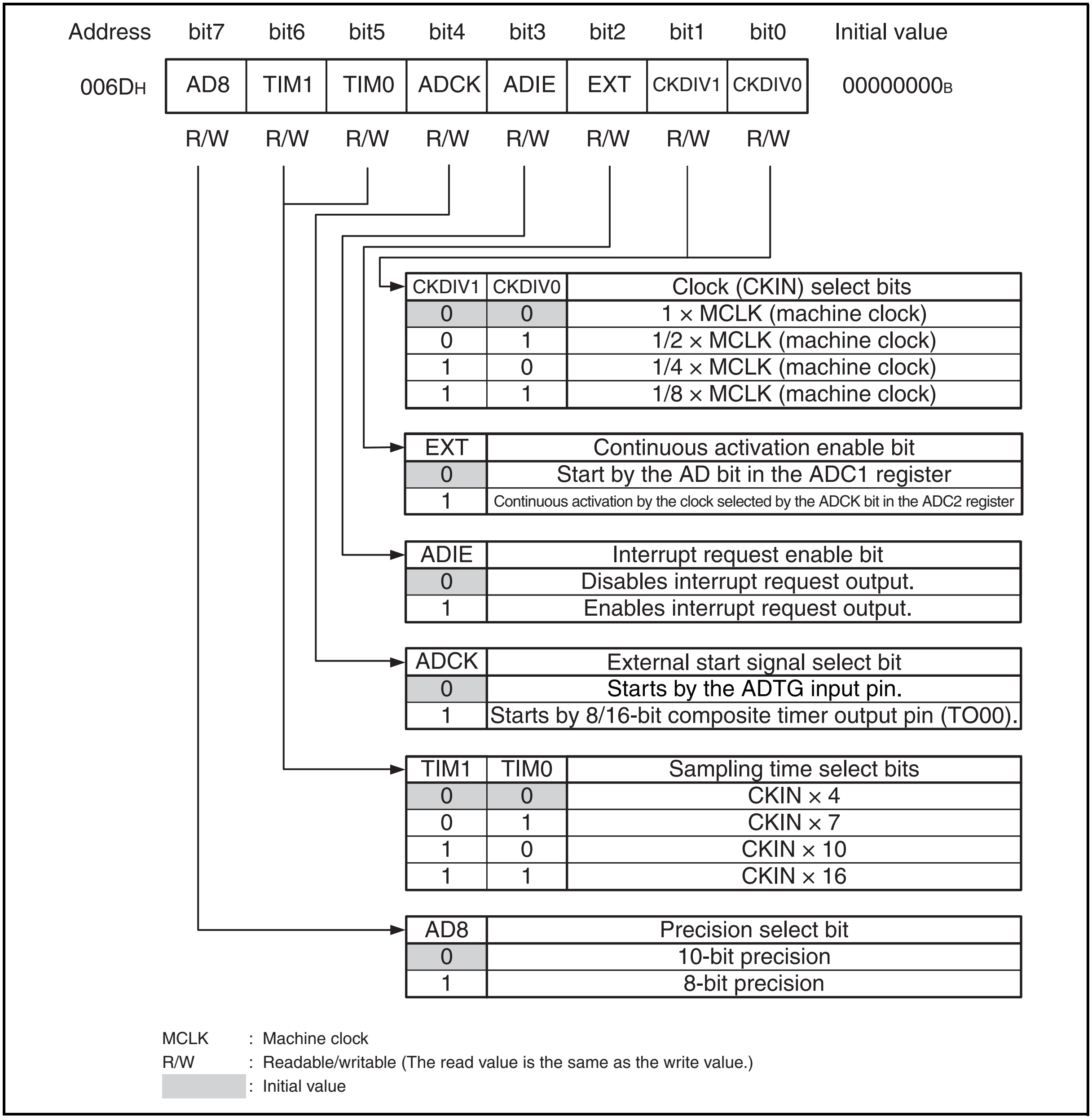
Bit name		Function
bit7 to bit4	ANS3, ANS2, ANS1, ANS0: Analog input pin select bits	These bits select an analog input pin to be used from AN00 to AN15. When A/D conversion is started (AD = 1) by the software (ADC2: EXT = 0), these bits can be modified simultaneously. Note: When the ADMV bit is "1", 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.

20.4.2 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 20.4-3 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 (machine clock)
0	1	1/2 × MCLK (machine clock)
1	0	1/4 × MCLK (machine clock)
1	1	1/8 × MCLK (machine clock)

EXT	Continuous activation enable bit
0	Start by the AD bit in the 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	Enables interrupt request output.

ADCK	External start signal select bit
0	Starts by the ADTG input pin.
1	Starts by 8/16-bit composite timer output pin (TO00).

TIM1	TIM0	Sampling time select bits
0	0	CKIN × 4
0	1	CKIN × 7
1	0	CKIN × 10
1	1	CKIN × 16

AD8	Precision select bit
0	10-bit precision
1	8-bit precision

MCLK : Machine clock
R/W : Readable/writable (The read value is the same as the write value.)
▨ : Initial value

Table 20.4-2 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.

20.4.3 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 20.4-4 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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	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 : The read value is "0". Writing a value to it has no effect on operation.
- : Undefined bit

The upper two bits of 10-bit A/D conversion result correspond to bit1 and bit0 in the ADDH register and the lower eight bits bit7 to bit0 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 A/D conversion result.

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 interrupt request flag bit (ADI) in the ADC1 register. During A/D conversion, the values of the ADDH and ADDL registers are results of the last A/D conversion.

20.5 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 20.5-1 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 converter	IRQ18	ILR4	L18	FFD6$_H$	FFD7$_H$

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

20.6 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 20.6-1 are required for activating the A/D conversion function with the software.

Figure 20.6-1 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 20.6-2 are required for continuous activation of the A/D conversion function.

Figure 20.6-2 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	
ADDL	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).

Note:

- 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).

20.7 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

● Note 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 AN16) 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 AN16), or after disconnecting analog input (AN00 to AN16).

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 x (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 MB95430H Series.

20.8 Example of Setting 8/10-bit A/D Converter

This section describes an example of setting the 8/10-bit A/D converter.

■ Example of Setting Methods

● 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 for current cutoff control bit (ADC1:ADMVX) to control the analog switch for cutting off internal reference power supply.

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)
To select the ADTG input pin	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 AN00 pin	Set the bits to "0000_B".
To use the AN01 pin	Set the bits to "0001_B".
To use the AN02 pin	Set the bits to "0010_B".
To use the AN03 pin	Set the bits to "0011_B".
To use the AN04 pin	Set the bits to "0100_B".
To use the AN05 pin	Set the bits to "0101_B".
To use the AN06 pin	Set the bits to "0110_B".
To use the AN07 pin	Set the bits to "0111_B".
To use the AN08 pin	Set the bits to "1000_B".
To use the AN09 pin	Set the bits to "1001_B".
To use the AN10 pin	Set the bits to "1010_B".
To use the AN11 pin	Set the bits to "1011_B".
To use the AN12 pin	Set the bits to "1100_B".
To use the AN13 pin	Set the bits to "1101_B".
To use the AN14 pin	Set the bits to "1110_B".
To use the AN15 pin	Set the bits to "1111_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 "0".

CHAPTER 21

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 only available in MB95F432K/F433K/F434K.)

21.1 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 MB95F432K/F433K/F434K only).

■ Low-voltage Detection Reset Circuit

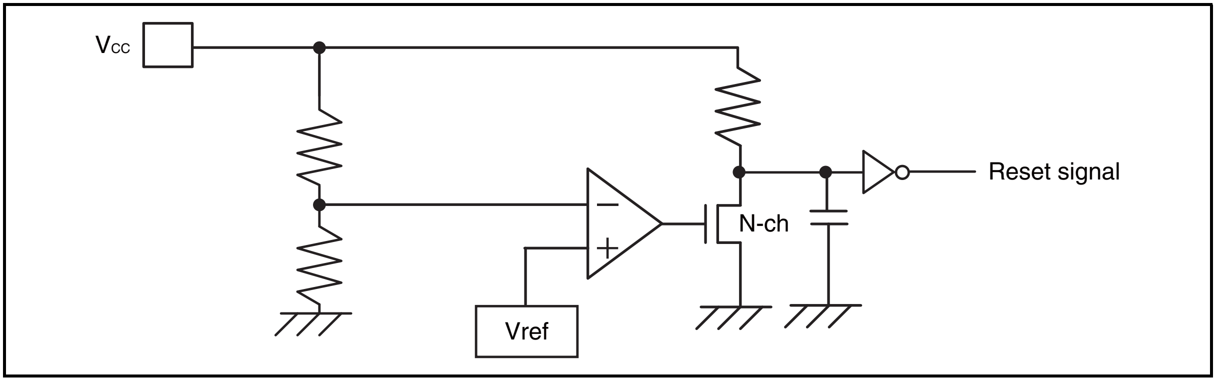
This circuit monitors power supply voltage and generates a reset signal if the power supply voltage drops below the detection voltage level. The circuit is available in MB95F432K/F433K/F434K only. Refer to the data sheet of the MB95430H Series for details of the electrical characteristics.

21.2 Configuration of Low-voltage Detection Reset Circuit

Figure 21.2-1 is the block diagram of the low-voltage detection reset circuit.

■ Block Diagram of Low-voltage Detection Reset Circuit

Figure 21.2-1 Block Diagram of Low-voltage Detection Reset Circuit



21.3 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.

● $\overline{\text{RST}}$ pin

The low-voltage detection reset signal is output inside the microcontroller and to this pin.

21.4 Operation of Low-voltage Detection Reset Circuit

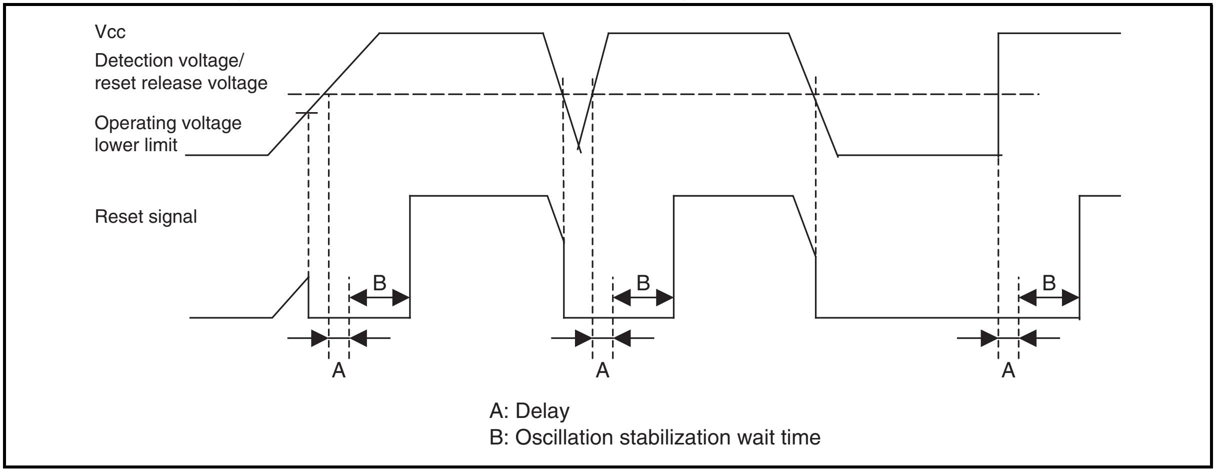
The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage.

■ 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 MB95430H Series.

Figure 21.4-1 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 22

CLOCK SUPERVISOR COUNTER

This chapter describes the functions and operations of the clock supervisor counter.

22.1 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 at specific times according to the time-base timer interval selected from eight options, 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).

22.2 Configuration of Clock Supervisor Counter

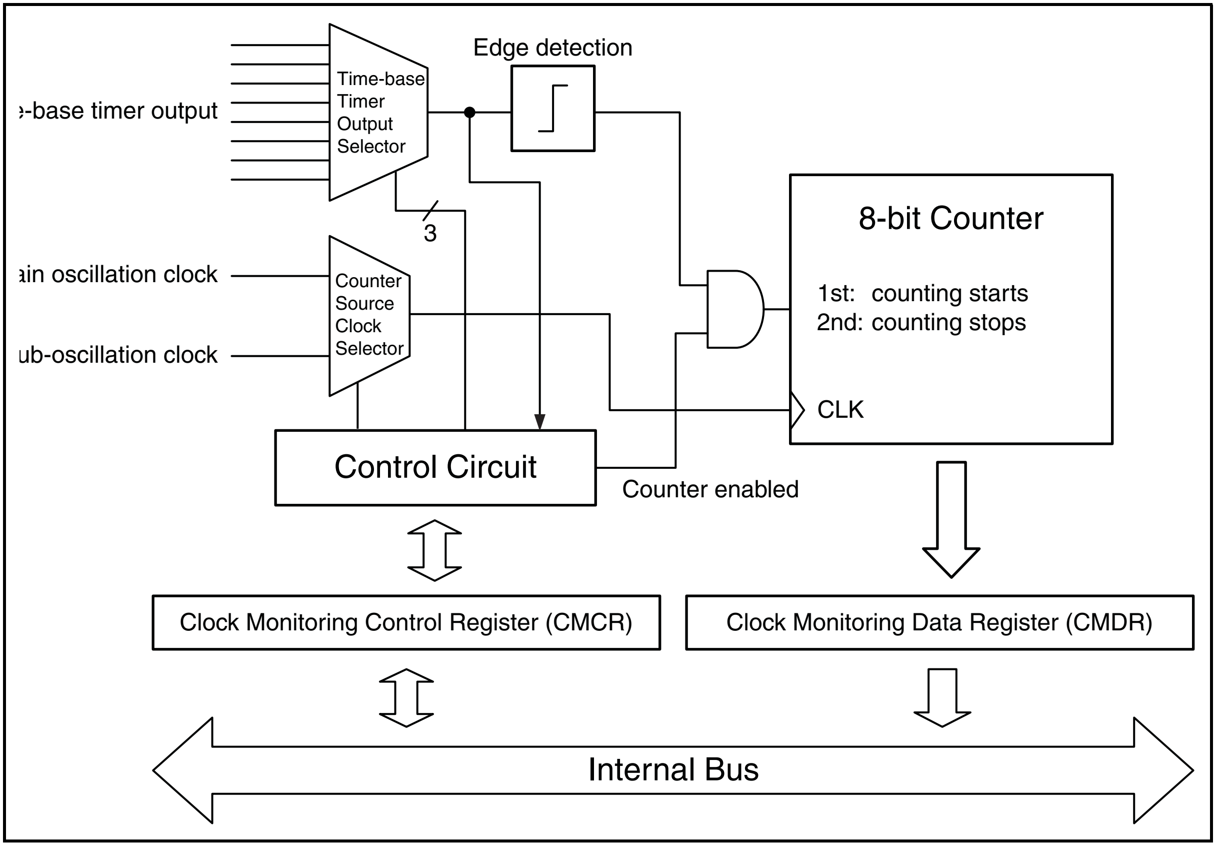
The clock supervisor counter consists of the following blocks:

- **Control circuit**
- **Clock Monitoring Control Register (CMCR)**
- **Clock Monitoring Data Register (CMDR)**
- **Time-base timer output selector**
- **Counter source clock selector**

■ Block Diagram of Clock Supervisor Counter

Figure 22.2-1 is the block diagram of the clock supervisor counter.

Figure 22.2-1 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 a counter source clock, select a counter enable period from eight different time-base 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.

● Time-base timer interval selector

This block is used to select the counter enable period from eight different time-base 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.

22.3 Registers of Clock Supervisor Counter

This section describes the registers of the clock supervisor counter.

■ Registers of Clock Supervisor Counter

Figure 22.3-1 shows the registers of the clock supervisor counter.

Figure 22.3-1 Registers of Clock Supervisor Counter

Clock monitoring data register (CMDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FEA_H$	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0	00000000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Clock monitoring control register (CMCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FE9_H$	-	-	Reserved	CMCSEL	TBTSEL2	TBTSEL1	TBTSEL0	CMCEN	00000000_B
	R0/WX	R0/WX	R/W0	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/W0	: The write value is "0"; the read value is the same as the write value.
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.
-	: Undefined bit

22.3.1 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 22.3-2 Clock Monitoring Data Register (CMDR)

Clock monitoring data register (CMDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FEA$_H$	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0	00000000$_B$
	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.)

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 time-base timer interval selected.

Table 22.3-1 Functions of Bits in Clock Monitoring Data Register (CMDR)

Bit name		Function
bit7 to bit0	CMDR to 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 time-base timer clock is detected twice (see Figure 22.5-2).

Note:

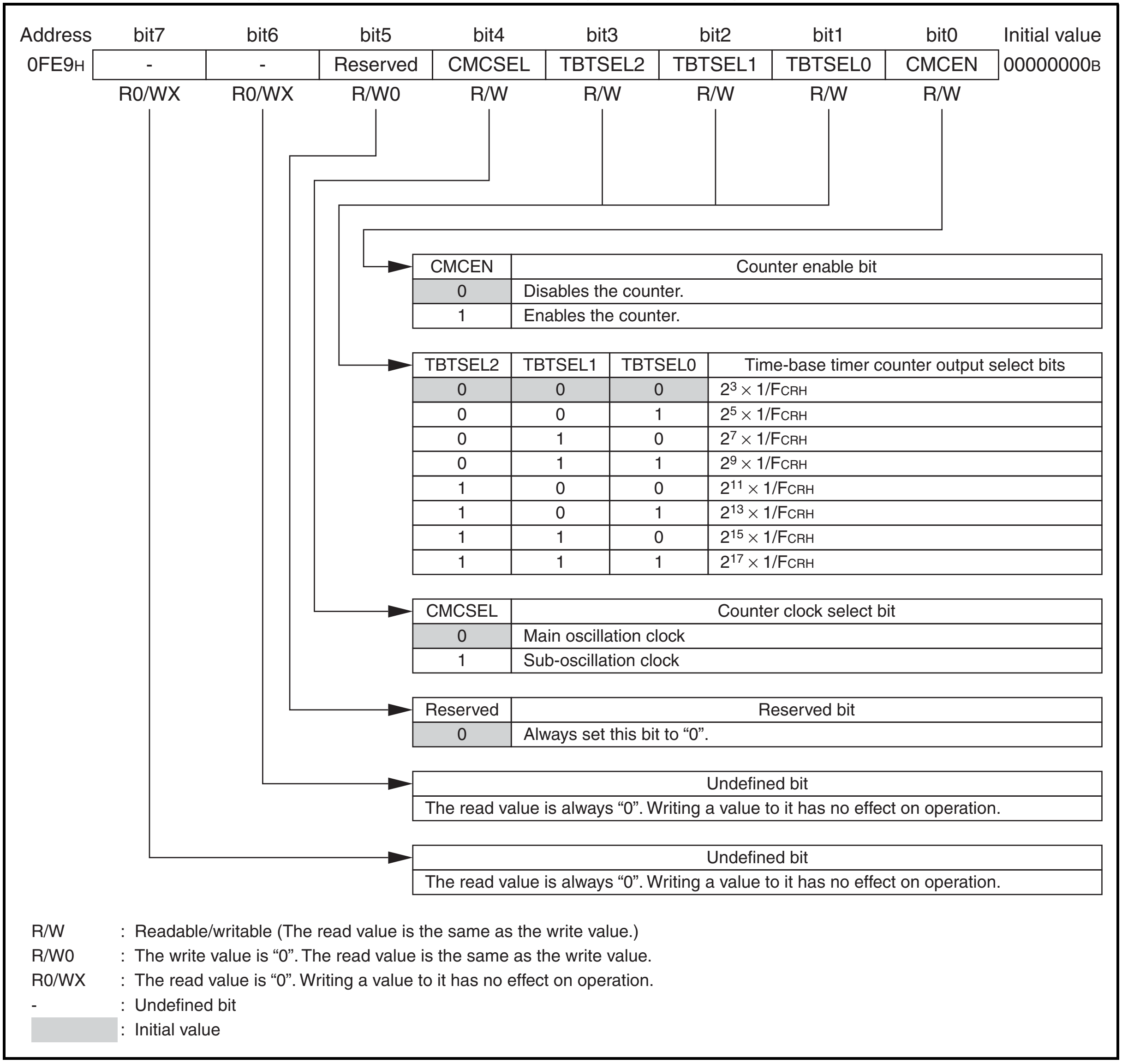
This register is "0" as long as the counter is operating (CMCEN = 1).

22.3.2 Clock Monitoring Control Register (CMCR)

The clock monitoring control register (CMCR) is used to select the counter source clock, select the time-base 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 22.3-3 Clock Monitoring Control Register (CMCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE9H	-	-	Reserved	CMCSEL	TBTSEL2	TBTSEL1	TBTSEL0	CMCEN	00000000B
	R0/WX	R0/WX	R/W0	R/W	R/W	R/W	R/W	R/W	

CMCEN	Counter enable bit
0	Disables the counter.
1	Enables the counter.

TBTSEL2	TBTSEL1	TBTSEL0	Time-base timer counter output 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

Reserved	Reserved bit
0	Always set this bit to “0”.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “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.)
R/W0	:	The write value is “0”. The read value is the same as the write value.
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	Initial value

Table 22.3-2 Functions of Bits in Clock Monitoring Control Register (CMCR)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7,
bit6</td><td>Undefined bits</td><td>The read value is always "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit5</td><td>Reserved bit</td><td>This bit is a reserved bit.
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.
<b>Writing "0":</b> Selects the external main oscillation clock as the source clock of the counter.
<b>Writing "1":</b> 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:
Time-base timer counter output select bit</td><td>These bits select the time-base timer interval.
The operation of the clock supervisor counter is enabled and disabled at specific times according to the time-base 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>Time-base timer counter output select bits</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.
<b>Writing "0":</b> Stops the counter and clears the CMDR register.
<b>Writing "1":</b> Enables the counter. The counter starts counting when detecting the rising edge of the time-base 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>

Note:

- Do not modify the CMCSEL bit when CMCEN = 1.
- Do not modify the TBTSEL[2:0] bits when CMCEN = 1.

22.4 Operations of Clock Supervisor Counter

This section describes the operations of the clock supervisor counter.

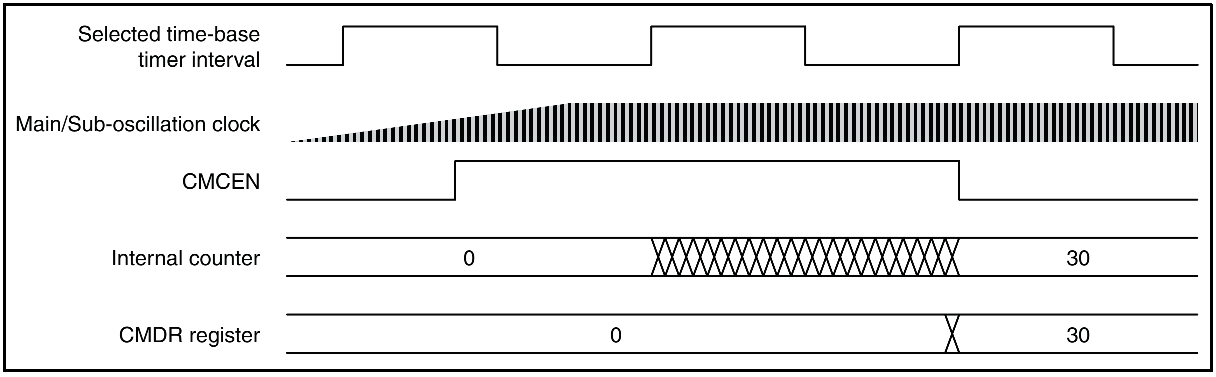
■ Clock Supervisor Counter

● 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 time-base timer interval selected from eight options by the TBTSEL [2:0] bits. Between two rising edges of the time-base 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.

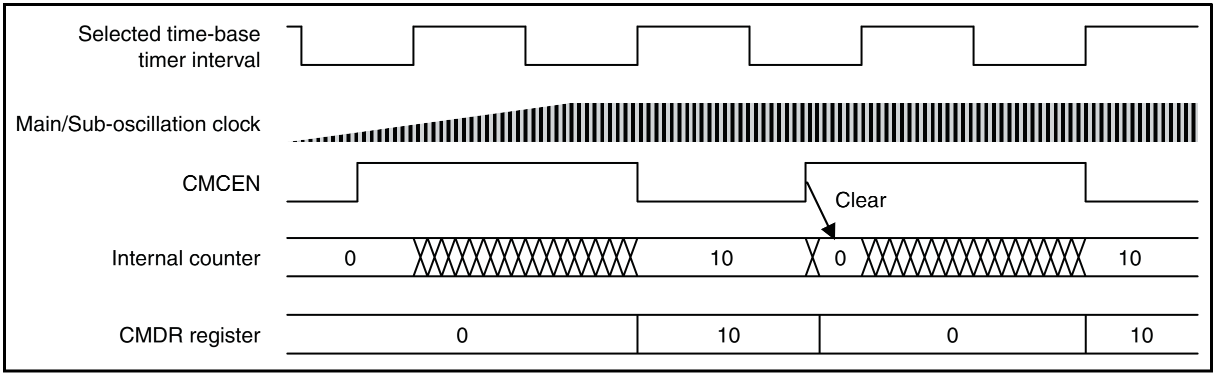
Figure 22.4-1 Clock Supervisor Counter Operation 1



● Clock Supervisor Counter Operation 2

The CMDR register is cleared when the CMCEN bit changes from "0" to "1".

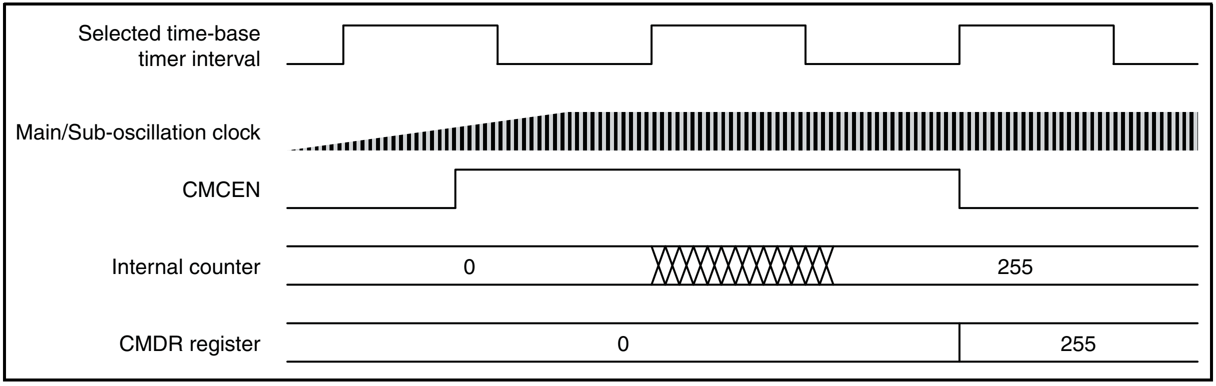
Figure 22.4-2 Clock Supervisor Counter Operation 2



● Clock Supervisor Counter Operation 3

The counter stops counting if it reaches "255". It cannot count further than "255".

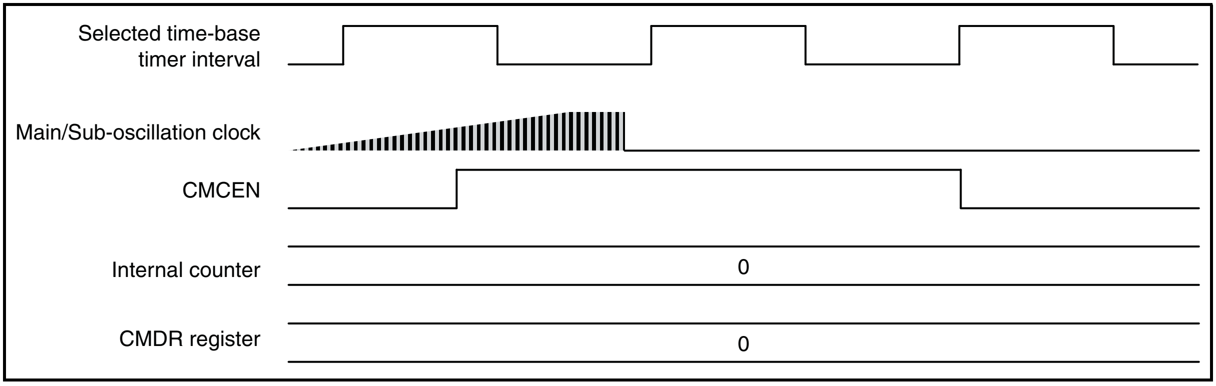
Figure 22.4-3 Clock Supervisor Counter Operation 3



● 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.

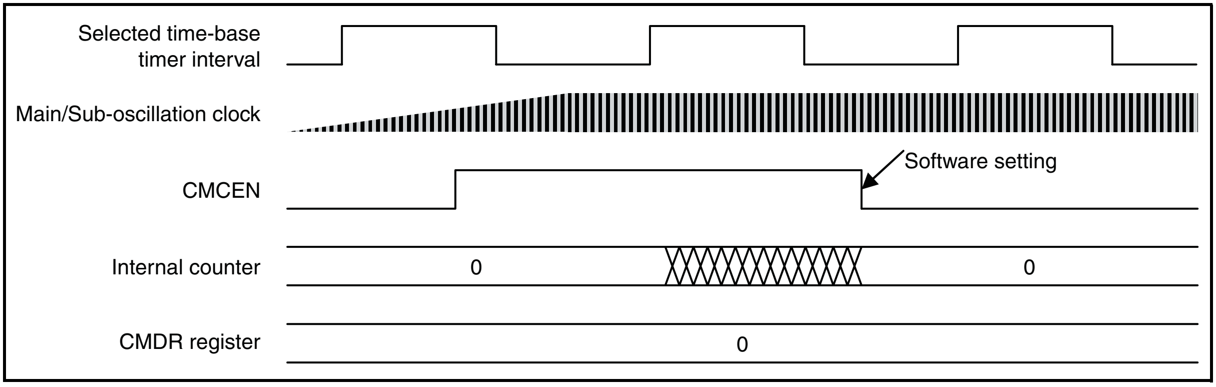
Figure 22.4-4 Clock Supervisor Counter Operation 4



● 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 22.4-5 Clock Supervisor Counter Operation 5



■ Table of Time-base Timer Intervals & Clock Supervisor Counter Values

Table 22.4-1 shows time-base timer intervals suitable for using different main CR clock frequency to measure different external clocks.

Table 22.4-1 Table of Counter Values in Relation to TBTSEL Settings

Main CR (FCRH) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measurement error	TBTSEL2 - TBTSEL0							
				"000"	"001"	"010"	"011"	"100"	"101"	"110"	"111"
				(2^{3}×1/FCRH)	(2^{5}×1/FCRH)	(2^{7}×1/FCRH)	(2^{9}×1/FCRH)	(2^{11}×1/FCRH)	(2^{13}×1/FCRH)	(2^{15}×1/FCRH)	(2^{17}×1/FCRH)
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
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

Table 22.4-1 Table of Counter Values in Relation to TBTSEL Settings

Main CR (F_{CRH}) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measurement 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})
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	33313
	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

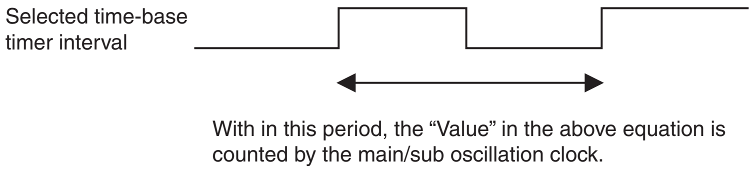
░ : Recommended setting

▓ : The counter value becomes "0" or "255".

Table 22.4-1 is calculated by the following equation:

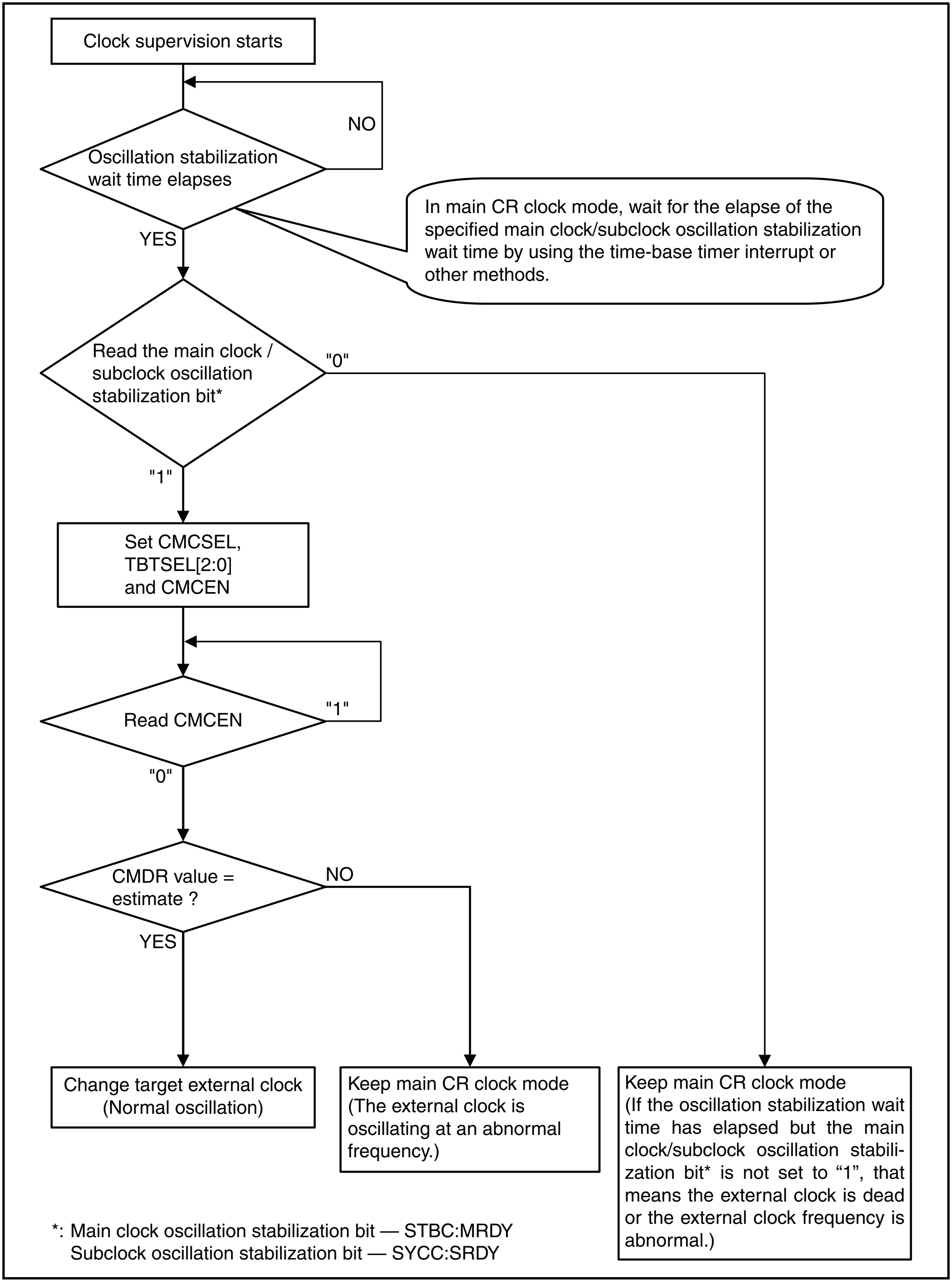
$$\text{Counter value} = \frac{\left\{\begin{array}{l} 2^{3} \times 1/F_{CRH}(\text{TBTSEL=000}) \\ 2^{5} \times 1/F_{CRH}(\text{TBTSEL=001}) \\ 2^{7} \times 1/F_{CRH}(\text{TBTSEL=010}) \\ 2^{9} \times 1/F_{CRH}(\text{TBTSEL=011}) \\ 2^{11} \times 1/F_{CRH}(\text{TBTSEL=100}) \\ 2^{13} \times 1/F_{CRH}(\text{TBTSEL=101}) \\ 2^{15} \times 1/F_{CRH}(\text{TBTSEL=110}) \\ 2^{17} \times 1/F_{CRH}(\text{TBTSEL=111}) \end{array}\right\} \times \text{Main/Sub-Oscillation Clock Frequency}}{2} \pm 1 \text{ (Measurement error)}$$

*Omit the decimal places of “Value”.


Selected time-base timer interval
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 22.4-6 Sample Operation Flow Chart of Clock Supervisor


Clock supervision starts
NO
Oscillation stabilization wait time elapses
YES
In main CR clock mode, wait for the elapse of the specified main clock/subclock oscillation stabilization wait time by using the time-base timer interrupt or other methods.
Read the main clock / subclock oscillation stabilization bit*
"0"
"1"
Set CMCSEL, TBTSEL[2:0] and CMCEN
Read CMCEN
"1"
"0"
CMDR value = estimate ?
NO
YES
Change target external clock (Normal oscillation)
Keep main CR clock mode (The external clock is oscillating at an abnormal frequency.)
Keep main CR clock mode (If the oscillation stabilization wait time has elapsed but the main clock/subclock oscillation stabilization bit* is not set to "1", that means the external clock is dead or the external clock frequency is abnormal.)
*: Main clock oscillation stabilization bit — STBC:MRDY
Subclock oscillation stabilization bit — SYCC:SRDY

If the time-base timer interrupt is used to make the clock supervisor counter wait for the oscillation stabilization time, please satisfy the following condition:

Time-base Timer Interval > Main/Sub-oscillation Stabilization Time × 1.05

e.g. F_{CH} = 4 MHz, F_{CRH} = 1 MHz, MWT[3:0] = 1111 (in WATR register)

$$\text{Time-base Timer Interval} > \frac{(2^{14} - 2)}{4 \times 10^6} \times 1.05 \approx 4.3[\text{ms}]$$

↓

TBC[3:0] = 0110 ($2^{13} \times 1/F_{CRH}$)

Note:

- See "10.1 Overview of Time-base Timer" for time-base timer interval settings.
- See "6.4 Oscillation Stabilization Wait Time Setting Register (WATR)" for main/sub-oscillation stabilization time settings.

22.5 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 time-base timer stops, the internal counter stops working. Do not clear the time-base timer while the clock supervisor counter is counting with the external clock.
- Select a time-base timer interval that is sufficiently long for the clock supervisor counter to operate. See Table 22.4-1 for time-base 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 time-base timer interval selected. If the machine clock cycle is longer than half the time-base timer interval selected, CMCEN may remain "1" even after the clock supervisor counter stops.

Table 22.5-1 below shows the appropriate clock gear setting for each TBTSEL setting.

Table 22.5-1 Appropriate Clock Gear Setting for Respective TBTSEL Settings

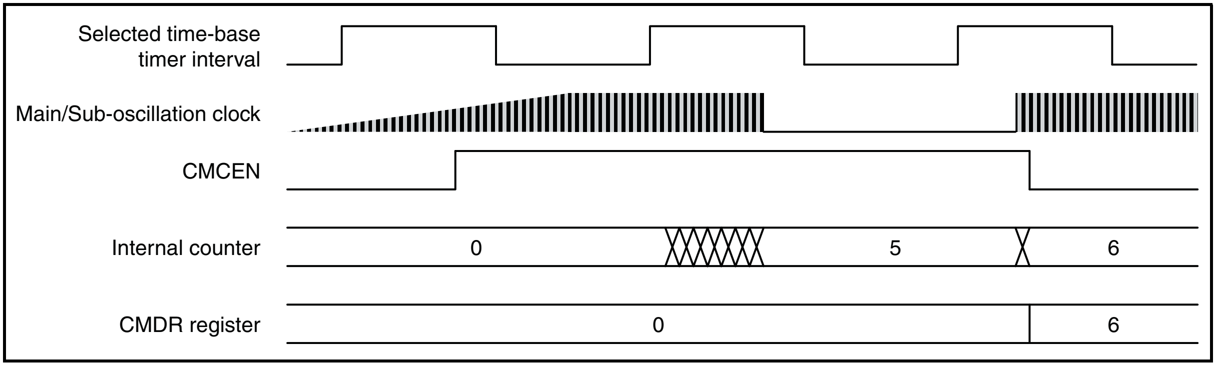
DIV (clock gear setting)	TBTSEL2 - TBTSEL0		
	000	001	010 - 111
	$2^3 \times 1/F_{CRH}$	$2^5 \times 1/F_{CRH}$	$2^7 \times 1/F_{CRH}$ - $2^{17} \times 1/F_{CRH}$
00 ($1 \times 1/F_{CRH}$)	❍	❍	❍
01 ($4 \times 1/F_{CRH}$)	x	❍	❍
10 ($8 \times 1/F_{CRH}$)	x	❍	❍
11 ($16 \times 1/F_{CRH}$)	x	x	❍

❍: Recommended

x: Prohibited

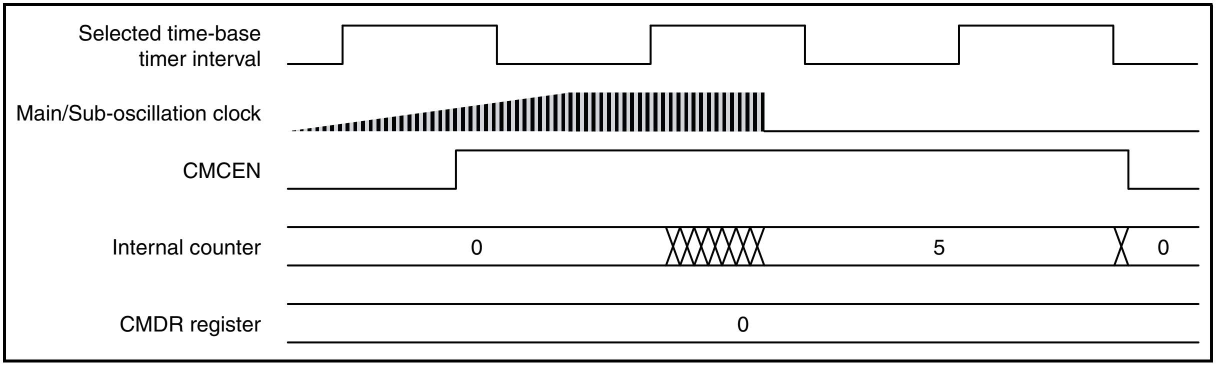
- If the external clock stops while the clock supervisor counter is operating, and it restarts after the second rising edge of the time-base timer interval selected, CMCEN is set to "0" after the external clock restarts.

Figure 22.5-1 Clock Supervisor Counter Operation 1



- After the clock supervisor counter stops, CMCEN is set to "0" when a falling edge of the time-base timer interval selected is detected after the second rising edge of the same interval. The counter is cleared at the same falling edge.

Figure 22.5-2 Clock Supervisor Counter Operation 2



CHAPTER 23

16-BIT PPG TIMER

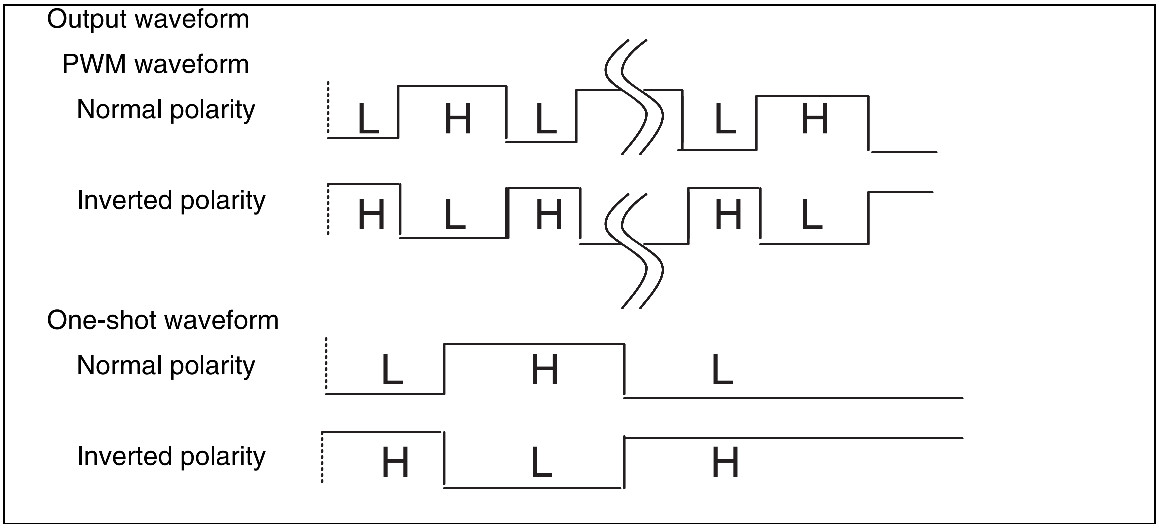
This chapter describes the functions and operations of the 16-bit PPG timer.

23.1 Overview of 16-bit PPG Timer

The 16-bit PPG timer can generate a PWM (Pulse Width Modulation) output or one-shot (square wave) output, and the period and duty of the output waveform can be changed by software freely. The timer can also generate an interrupt when a start trigger occurs or on the rising or falling edge of the output waveform.

■ 16-bit PPG Timer

The 16-bit PPG timer can output the PWM output and the one shot. The output wave form can be reversed by setting the register (Normal polarity ↔ Inverted polarity).



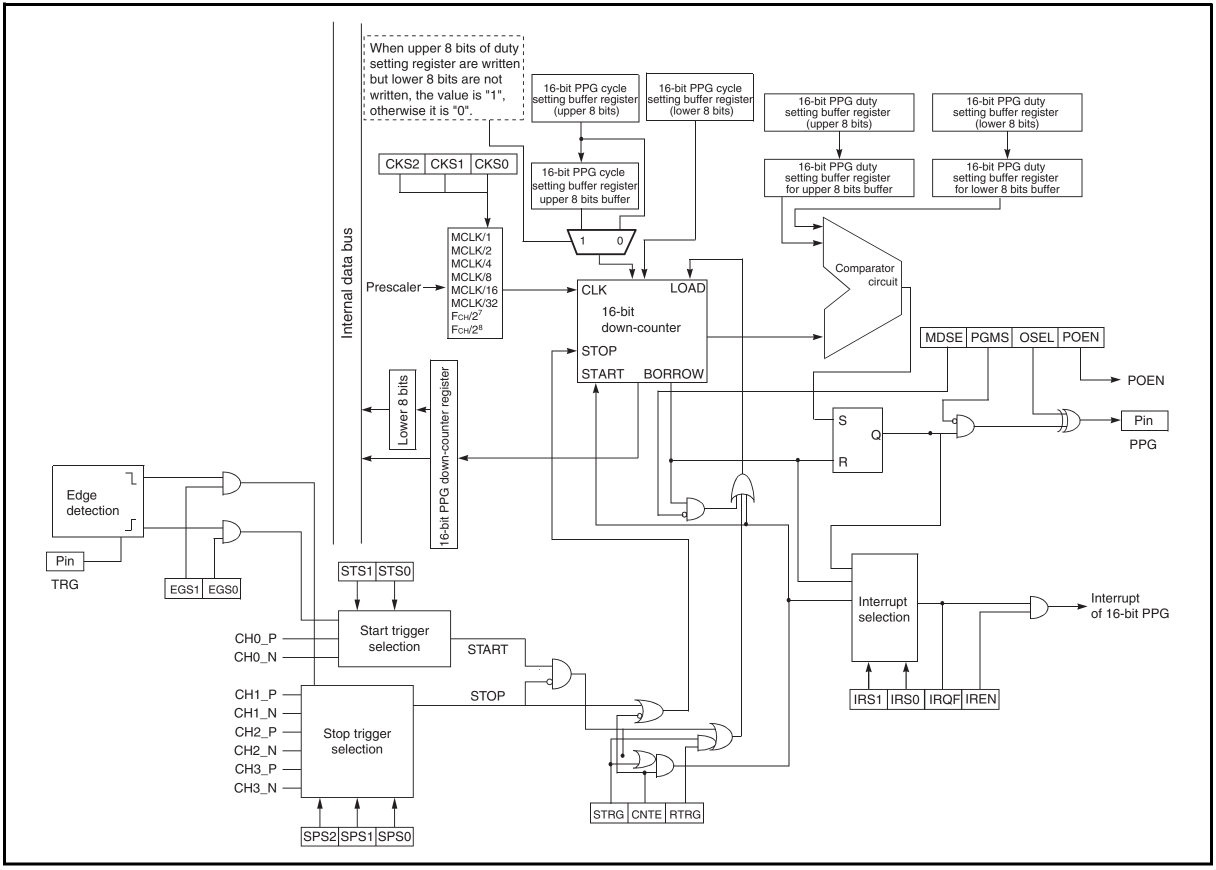
- The count operation clock can be selected from eight different clock sources (MCLK/1, MCLK/2, MCLK/4, MCLK/8, MCLK/16, MCLK/32, $F_{CH}/2^7$, or $F_{CH}/2^8$). (MCLK: Machine clock, F_{CH}: Main clock)
- Interrupt can be selectively triggered by the following four conditions:
  - Occurrence of a start trigger in the PPG timer
  - Occurrence of a counter borrow in the 16-bit down-counter (cycle match).
  - Rising edge of PPG in normal polarity or falling edge of PPG in inverted polarity
  - Counter borrow, rising edge of PPG in normal polarity, or falling edge of PPG in inverted polarity

23.2 Configuration of 16-bit PPG Timer

Shown below is the block diagram of the 16-bit PPG timer.

■ Block Diagram of 16-bit PPG Timer

Figure 23.2-1 Block Diagram of 16-bit PPG Timer


When upper 8 bits of duty setting register are written but lower 8 bits are not written, the value is "1", otherwise it is "0".
16-bit PPG cycle setting buffer register (upper 8 bits)
16-bit PPG cycle setting buffer register (lower 8 bits)
16-bit PPG duty setting buffer register (upper 8 bits)
16-bit PPG duty setting buffer register (lower 8 bits)
CKS2
CKS1
CKS0
16-bit PPG cycle setting buffer register upper 8 bits buffer
16-bit PPG duty setting buffer register for upper 8 bits buffer
16-bit PPG duty setting buffer register for lower 8 bits buffer
1
0
Internal data bus
MCLK/1
MCLK/2
MCLK/4
MCLK/8
MCLK/16
MCLK/32
FCH/2^7
FCH/2^8
Prescaler
CLK
LOAD
16-bit down-counter
STOP
START
BORROW
Comparator circuit
MDSE
PGMS
OSEL
POEN
POEN
Lower 8 bits
16-bit PPG down-counter register
S
Q
R
Pin
PPG
Edge detection
Pin
TRG
EGS1
EGS0
STS1
STS0
Start trigger selection
CH0_P
CH0_N
START
STOP
CH1_P
CH1_N
CH2_P
CH2_N
CH3_P
CH3_N
Stop trigger selection
SPS2
SPS1
SPS0
STRG
CNTE
RTRG
Interrupt selection
Interrupt of 16-bit PPG
IRS1
IRS0
IRQF
IREN

● Count clock selector

The clock for the countdown of 16-bit down-counter is selected from eight types of internal count clocks.

● 16 bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" until the value of 16-bit down-counter is corresponding to the value of the 16-bit PPG duty setting buffer register from the value of 16-bit PPG cycle setting buffer register.

Afterwards, after keep "L" the output until the counter value is corresponding to "1", it keeps counting 16-bit down-counter from the value of 16-bit PPG cycle setting buffer register.

- CHn_P/CHn_N (n = 0, 1, 2, 3)

CHn_P: the nth channel of the rising edge signal of the voltage comparator

CHn_N: the nth channel of the falling edge signal of the voltage comparator

- 16-bit PPG down-counter register upper, lower (PDCRH0, PDCRL0)

The value of 16-bit down-counter of 16-bit PPG timer is read.

- 16-bit PPG cycle setting buffer register upper, lower (PCSRH0, PCSRL0)

The compare value for the cycle of 16-bit PPG timer is set.

- 16-bit PPG duty setting buffer register upper, lower (PDUTH0, PDUTL0)

The compare value for "H" width of 16-bit PPG timer is set.

- 16-bit PPG status control register upper, lower (PCNTH0, PCNTL0)

The operation mode and the operation condition of 16-bit PPG timer are set.

- 16-bit PPG trigger source control register (PTGS)

The hardware trigger source of the 16-bit PPG timer is set.

■ Input Clock

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

23.3 Channel of 16-bit PPG Timer

This section describes the channel of the 16-bit PPG timer.

■ Channel of 16-bit PPG Timer

The MB95430H Series has one 16-bit PPG timer.

Table 23.3-1 and Table 23.3-2 show the pins and registers of the 16-bit PPG timer respectively.

Table 23.3-1 Pins of 16-bit PPG Timer

Channel	Pin name	Pin function
0	PPG	PPG output
	TRG	Trigger input

Table 23.3-2 Registers of 16-bit PPG Timer

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	PDCRH0	16-bit PPG down-counter register (upper)
	PDCRL0	16-bit PPG down-counter register (lower)
	PCSRH0	16-bit PPG cycle setting buffer register (upper)
	PCSRL0	16-bit PPG cycle setting buffer register (lower)
	PDUTH0	16-bit PPG duty setting buffer register (upper)
	PDUTL0	16-bit PPG duty setting buffer register (lower)
	PCNTH0	16-bit PPG status control register (upper)
	PCNTL0	16-bit PPG status control register (lower)
	PTGS0	16-bit PPG trigger source control register

23.4 Pins of 16-bit PPG Timer

This section describes the pins of the 16-bit PPG timer.

■ Pins of 16-bit PPG Timer

The pins of the 16-bit PPG timer are namely the PPG pin and TRG pin.

● PPG pin

This pin serves as a general-purpose I/O port as well as a 16-bit PPG timer output.

PPG: A PPG waveform is output to this pin. The PPG waveform can be output by using the 16-bit PPG status control register to enable output (PCNTL0: POEN=1).

● TRG pin

TRG: Used to start the 16-bit PPG timer by the hardware trigger.

■ Block Diagrams of Pins of 16-bit PPG Timer

Figure 23.4-1 Block Diagram of Pin PPG of 16-bit PPG

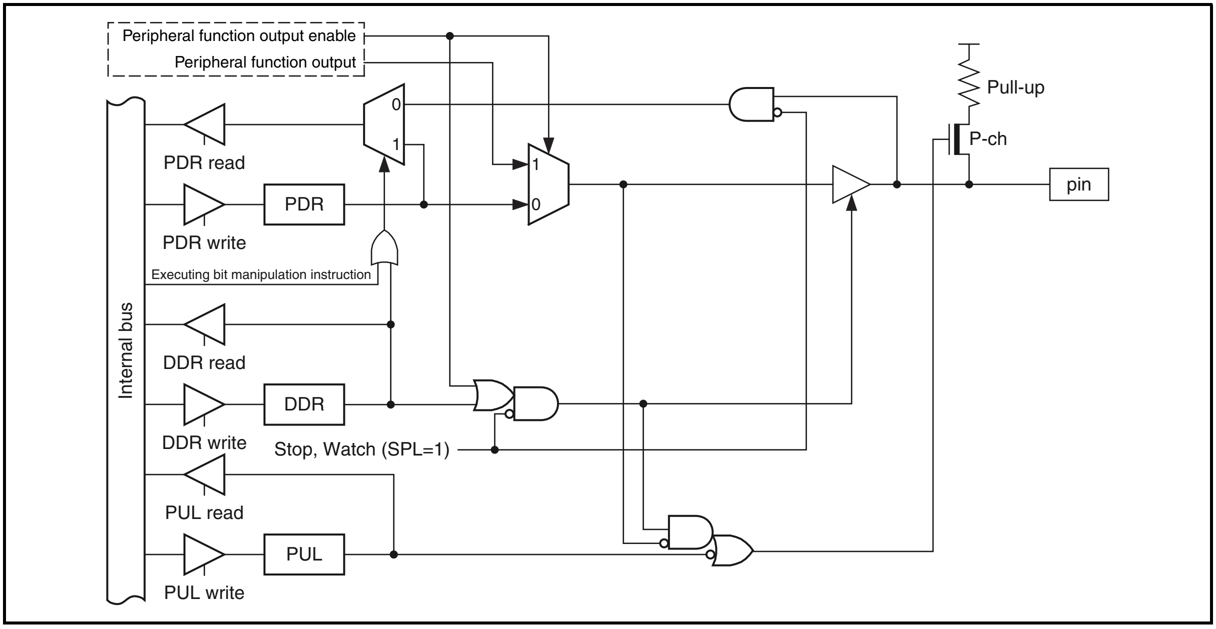
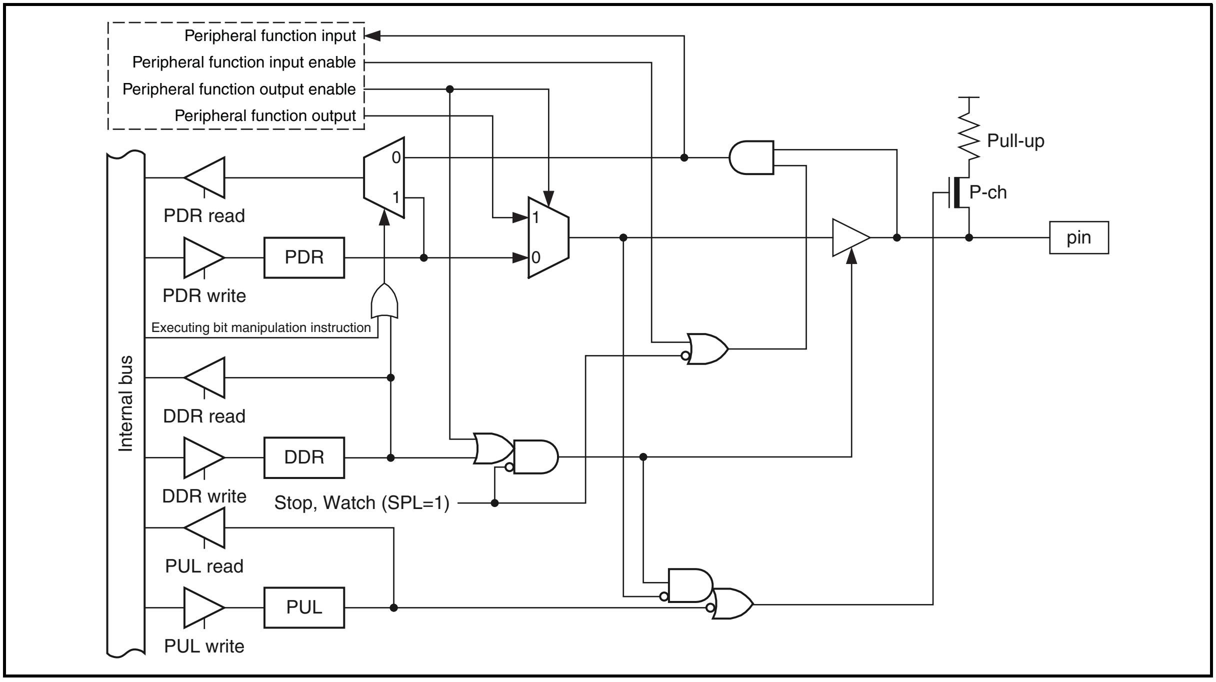


Figure 23.4-2 Block Diagram of Pin TRG of 16-bit PPG


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Pull-up
P-ch
pin
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
PUL read
PUL
PUL write

23.5 Registers of 16-bit PPG Timer

This section describes the registers of the 16-bit PPG timer.

■ Registers of 16-bit PPG Timer

Figure 23.5-1 Registers of 16-bit PPG Timer

16-bit PPG down-counter register (upper) (PDCRH0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAA$_H$	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000$_B$
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down-counter register (lower) (PDCRL0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAB$_H$	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000$_B$
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG cycle setting buffer register (upper) (PCSRH0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAC$_H$	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG cycle setting buffer register (lower) (PCSRL0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAD$_H$	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (upper) (PDUTH0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAE$_H$	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (lower) (PDUTL0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAF$_H$	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG status control register (upper) (PCNTH0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0042$_H$	CNTE	STRG	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	00000000$_B$
	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG status control register (lower) (PCNTL0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0043$_H$	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	00000000$_B$
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

16-bit PPG trigger source control register (PTGS0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0044$_H$	-	-	-	SPS2	SPS1	SPS0	STS1	STS0	00000000$_B$
	R0/WX	R0/WX	R0/WX	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.)
R0,W : Write only (Writable. The read value is "0".)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.

23.5.1 16- bit PPG Down-counter Registers Upper, Lower (PDCRH0, PDCRL0)

The 16-bit PPG down-counter registers upper, lower (PDCRH0, PDCRL0) form a 16-bit register which is used to read the count value from the 16-bit PPG down-counter.

■ 16-bit PPG Down-counter Registers Upper, Lower (PDCRH0, PDCRL0)

Figure 23.5-2 16-bit PPG Down-counter Registers Upper, Lower (PDCRH0, PDCRL0)

16-bit PPG down-counter register (upper) (PDCRH0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAA$_H$	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000$_B$
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down-counter register (lower) (PDCRL0)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAB$_H$	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000$_B$
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)

These registers form a 16-bit register which is used to read the count value from the 16-bit down-counter. The initial values of the register are all "0".

Always use one of the following procedures to read from this register.

- Use the "MOVW" instruction (use a 16-bit access instruction to read the PDCRH0 register address).
- Use the "MOV" instruction and read PDCRH0 first and then PDCRL0 (reading PDCRH0 automatically copies the lower 8 bits of the down-counter to PDCRL0).

These registers are read-only and writing has no effect on the operation.

Note:

If you use the "MOV" instruction and read PDCRL0 before PDCRH0, PDCRL0 will return the value from the previous valid read operation. Therefore, the value of the 16-bit down-counter will not be read correctly.

23.5.2 16-bit PPG Cycle Setting Buffer Registers Upper, Lower (PCSRH0, PCSRL0)

The 16-bit PPG cycle setting buffer registers are used to set the cycle for the output pulses generated by the PPG.

■ 16-bit PPG Cycle Setting Buffer Registers Upper, Lower (PCSRH0, PCSRL0)

Figure 23.5-3 16-bit PPG Cycle Setting Buffer Registers Upper, Lower (PCSRH0, PCSRL0)

16-bit PPG cycle setting buffer register (upper) (PCSRH0)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAC$_H$	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG cycle setting buffer register (lower) (PCSRL0)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAD$_H$	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

These registers form a 16-bit register which sets the period for the output pulses generated by the PPG. The values set in these registers are loaded to the down-counter.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PCSRH0 register address).
- Use the "MOV" instruction and write to PCSRH0 first and then PCSRL0.
  If a down-counter load occurs after writing data to PCSRH0 (but before writing data to PCSRL0), the previous valid PCSRH0/PCSRL0 value will be loaded to the down-counter. If the PCSRH0/PCSRL0 value is modified during counting, the modified value will become effective from the next load of the down-counter.
- Do not set PCSRH0 and PCSRL0 to "00$_H$", or PCSRH0 to "01$_H$" and PCSRL0 to "01$_H$".

Note:

If the down-counter load occurs after the "MOV" instruction is used to write data to PCSRL0 before PCSRH0, the previous valid PCSRH0 value and newly written PCSRL0 value are loaded to the down-counter. It should be noted that as a result, the correct period cannot be set.

23.5.3 16-bit PPG Duty Setting Buffer Registers Upper, Lower (PDUTH0, PDUTL0)

The 16-bit PPG duty setting buffer registers control the duty ratio for the output pulses generated by the PPG.

■ 16-bit PPG Duty Setting Buffer Registers Upper, Lower (PDUTH0, PDUTL0)

Figure 23.5-4 16-bit PPG Duty Setting Buffer Registers Upper, Lower (PDUTH0, PDUTL0)

16-bit PPG duty setting buffer register (upper) (PDUTH0)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAE$_H$	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (lower) (PDUTL0)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAF$_H$	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

These registers form a 16-bit register which controls the duty ratio for the output pulses generated by the PPG. Transfer of the data from the 16-bit PPG duty setting buffer registers to the duty setting registers is performed at the same timing as the down-counter read.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PDUTH0 register address).
- Use the "MOV" instruction and write to PDUTH0 first and then PDUTL0.
  If a down-counter load occurs after writing data to PDUTH0 (but before writing data to PDUTL0), the value of the 16-bit PPG duty setting buffer registers is not transferred to the duty setting registers.

The relation between the value of the 16-bit PPG duty setting registers and output pulse is as follows:

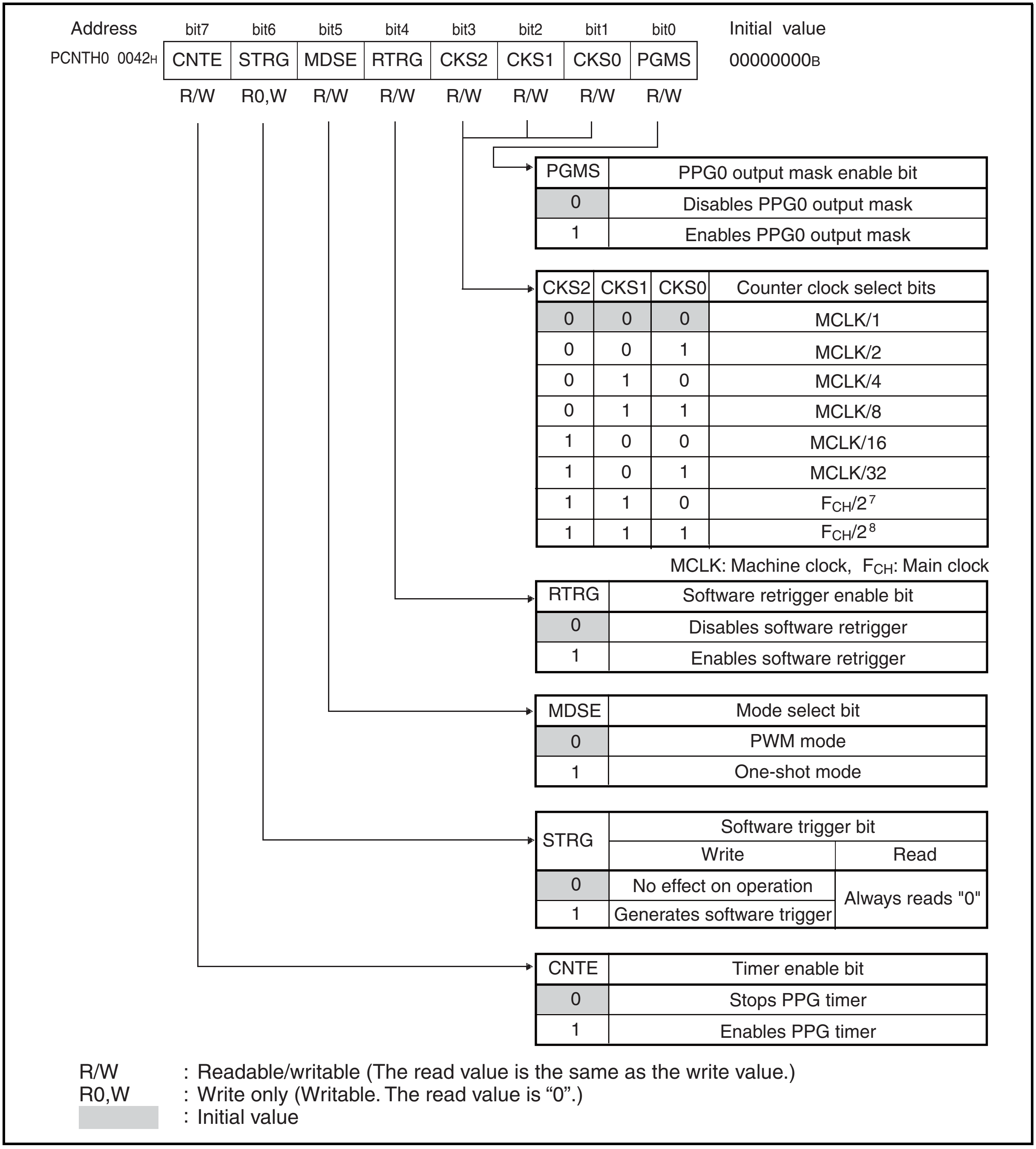
- When the same value is set in both the 16-bit PPG cycle setting buffer registers and duty setting registers, the "H" level will always be output if normal polarity is set, or the "L" level will always be output if inverted polarity is set.
- When the duty setting registers are set to "00$_B$", the "L" level will always be output if normal polarity is set, or the "H" level will always be output if inverted polarity is set.
- When the value set in the duty setting registers is greater than the value in the 16-bit PPG cycle setting buffer registers, the "L" level will always be output if normal polarity is set, and the "H" level will always be output if inverted polarity is set.

23.5.4 16-bit PPG Status Control Register Upper, Lower (PCNTH0, PCNTL0)

The 16-bit PPG status control register is used to enable and disable the 16-bit PPG timer and also to set the operating status for the software trigger, retrigger control interrupt, and output polarity. This register can also check the operation status.

■ 16-bit PPG Status Control Register, Upper (PCNTH0)

Figure 23.5-5 16-bit PPG Status Control Register, Upper (PCNTH0)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCNTH0 0042H	CNTE	STRG	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	00000000B
	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	R/W	

PGMS	PPG0 output mask enable bit
0	Disables PPG0 output mask
1	Enables PPG0 output mask

CKS2	CKS1	CKS0	Counter clock select bits
0	0	0	MCLK/1
0	0	1	MCLK/2
0	1	0	MCLK/4
0	1	1	MCLK/8
1	0	0	MCLK/16
1	0	1	MCLK/32
1	1	0	$F_{CH}/2^7$
1	1	1	$F_{CH}/2^8$

MCLK: Machine clock, F_{CH}: Main clock

RTRG	Software retrigger enable bit
0	Disables software retrigger
1	Enables software retrigger

MDSE	Mode select bit
0	PWM mode
1	One-shot mode

STRG	Software trigger bit	
	Write	Read
0	No effect on operation	Always reads "0"
1	Generates software trigger	

CNTE	Timer enable bit
0	Stops PPG timer
1	Enables PPG timer

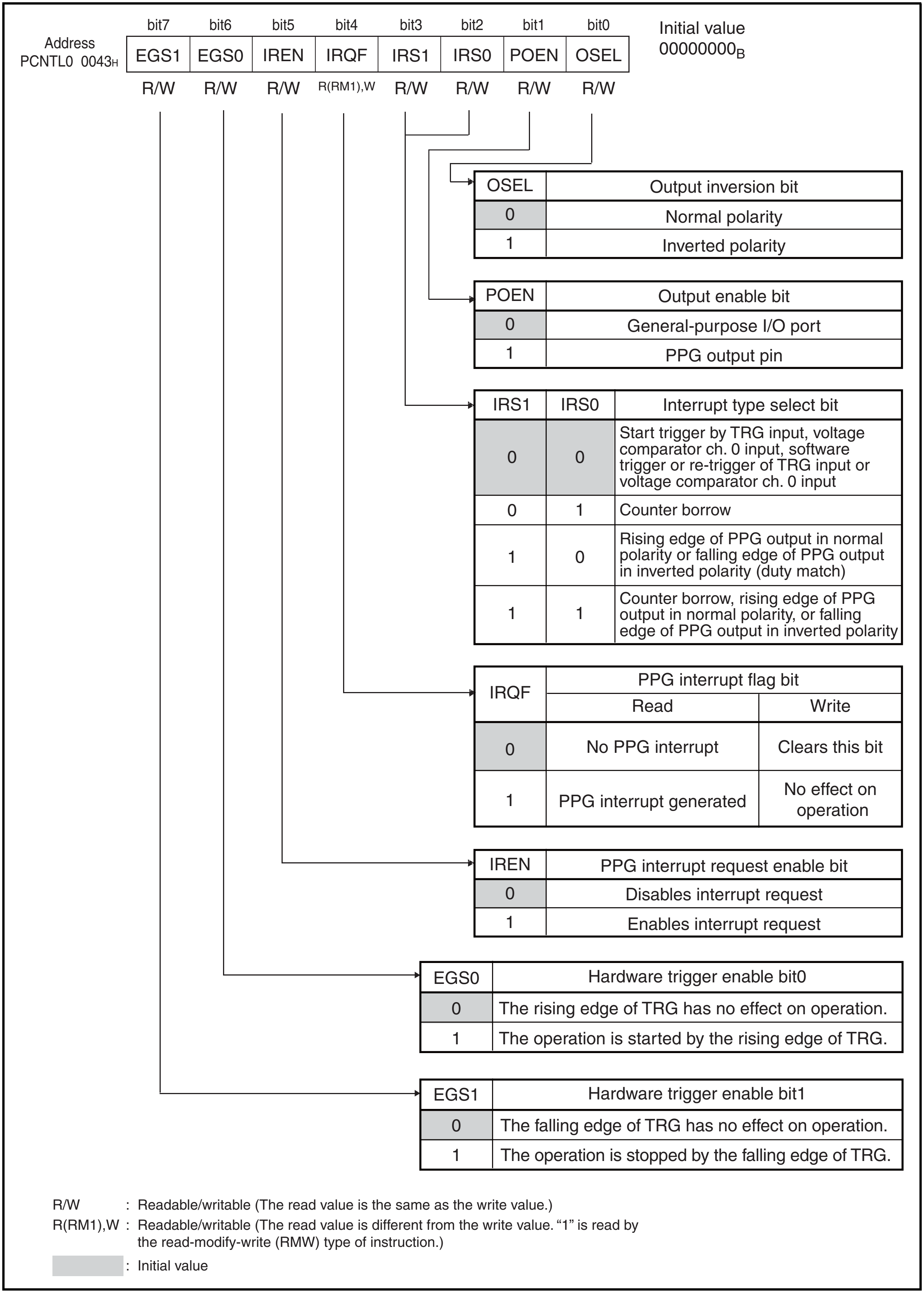
R/W : Readable/writable (The read value is the same as the write value.)
R0,W : Write only (Writable. The read value is “0”.)
▨ : Initial value

Table 23.5-1 16-bit PPG Status Control Register, Upper (PCNTH0)

Bit name		Function
bit7	CNTE: Timer enable bit	This bit is used to enable/stop PPG timer operation. **Writing "0":** The PPG operation halts immediately and the PPG1 output goes to the initial level ("L" output if OSEL is "0"; "H" output if OSEL is "1"). **Writing "1"**: PPG operation is enabled and the PPG goes to standby to wait for a trigger.
bit6	STRG: Software trigger bit	This bit is used to start the PPG timer by software. When the bit is set to "1", setting the CNTE bit to "1" starts the PPG timer. Reading this bit always returns "0".
bit5	MDSE: Mode select bit	This bit is used to set the PPG operation mode. **Writing "0"**: The PPG operates in PWM mode. **Writing "1"**: The PPG operates in one-shot mode. Note: Modifying this bit is prohibited during operation.
bit4	RTRG: Software retrigger enable bit	This bit is used to enable or disable the software retrigger function of the PPG during operation. **Writing "0":** Disables the software retrigger function. **Writing "1":** Enables the software retrigger function.
bit3 to bit1	CKS2 to CKS0: Count clock select bits	These bits select the operating clock for the 16-bit PPG timer. The count clock signal is generated by the prescaler. See "6.12 Operation of Prescaler". Note: As the time-base timer (TBT) is halted in subclock mode, $F_{CH}/2^7$ and $F_{CH}/2^8$ cannot be selected in this case.
bit0	PGMS: PPG output mask enable bit	This bit is used to mask the PPG1 output to a specific level regardless of the mode setting (MDSE: bit5), period setting (PCSRH0, PCSRL0), and duty setting (PDUTH0, PDUTL0). **Writing "0":** The PPG1 output mask function is disabled. **Writing "1":** The PPG1 output mask function is enabled. When the PPG0 output polarity setting is set to "normal" (PCNTL0: OSEL = 0), the output is always masked to "L". When the polarity setting is se to "inverted" (PCNTL0: OSEL = 1), the PPG0 output is always masked to "H".

■ 16-bit PPG Status Control Register, Lower (PCNTL0)

Figure 23.5-6 16-bit PPG Status Control Register, Lower (PCNTL0)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCNTL0 0043H	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	00000000B
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

OSEL	Output inversion bit
0	Normal polarity
1	Inverted polarity

POEN	Output enable bit
0	General-purpose I/O port
1	PPG output pin

IRS1	IRS0	Interrupt type select bit
0	0	Start trigger by TRG input, voltage comparator ch. 0 input, software trigger or re-trigger of TRG input or voltage comparator ch. 0 input
0	1	Counter borrow
1	0	Rising edge of PPG output in normal polarity or falling edge of PPG output in inverted polarity (duty match)
1	1	Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity

IRQF	PPG interrupt flag bit	
	Read	Write
0	No PPG interrupt	Clears this bit
1	PPG interrupt generated	No effect on operation

IREN	PPG interrupt request enable bit
0	Disables interrupt request
1	Enables interrupt request

EGS0	Hardware trigger enable bit0
0	The rising edge of TRG has no effect on operation.
1	The operation is started by the rising edge of TRG.

EGS1	Hardware trigger enable bit1
0	The falling edge of TRG has no effect on operation.
1	The operation is stopped by the falling edge of TRG.

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 23.5-2 16-bit PPG Status Control Register, Lower (PCNTL0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>EGS1:
Hardware trigger enable bit1</td><td>This bit determines whether to allow or disallow the falling edge of TRG input to stop operation.
<b>Writing "0":</b> The falling edge of TRG has no effect on operation.
<b>Writing "1":</b> The operation is stopped by the falling edge of TRG.</td></tr>
<tr><td>bit6</td><td>EGS0:
Hardware trigger enable bit0</td><td>This bit determines whether to allow or disallow the rising edge of TRG input to start operation.
<b>Writing "0":</b> The rising edge of TRG has no effect on operation.
<b>Writing "1":</b> The operation is started by the rising edge of TRG.</td></tr>
<tr><td>bit5</td><td>IREN:
PPG interrupt request enable bit</td><td>This bit enables or disables PPG interrupt request to the interrupt controller.
<b>Writing "0":</b> Enables the interrupt request.
<b>Writing "1":</b> Disables the interrupt request.</td></tr>
<tr><td>bit4</td><td>IRQF:
PPG interrupt flag bit</td><td>This bit is set to "1" when a PPG interrupt occurs.
<b>Writing "0":</b> Clears the bit.
<b>Writing "1":</b> Has no effect on operation.
"1" is always read in read-modify-write (RMW) instruction.</td></tr>
<tr><td>bit3,
bit2</td><td>IRS1, IRS0:
Interrupt type select bits</td><td>These bits select the interrupt type for the PPG timer.
<table>
<tr><th>IRS1</th><th>IRS0</th><th>Type of interrupt</th></tr>
<tr><td>0</td><td>0</td><td>Start trigger by TRG input, voltage comparator ch. 0 input, software trigger or retrigger of TRG input or voltage comparator ch. 0 input</td></tr>
<tr><td>0</td><td>1</td><td>Counter borrow</td></tr>
<tr><td>1</td><td>0</td><td>Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr>
<tr><td>1</td><td>1</td><td>Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr>
</table></td></tr>
<tr><td>bit1</td><td>POEN:
Output enable bit</td><td>This bit enables or disables output from the PPG output pin.
<b>Writing "0"</b>: The pin serves as a general-purpose port.
<b>Writing "1"</b>: The pin serves as the PPG timer output pin.</td></tr>
<tr><td>bit0</td><td>OSEL:
Output inversion bit</td><td>This bit selects the polarity of PPG output pin.
<b>Writing "0"</b>: The PPG output goes to "H" when "L" is output in the internal start and the 16-bit down-counter value matches the duty setting register value, and goes to "L" when a down-counter borrow occurs (Normal polarity).
<b>Writing "1"</b>: The PPG output is inverted (Inverted polarity).</td></tr>
</table>

23.5.5 16-bit PPG Trigger Source Control Register (PTGS0)

The 16-bit PPG trigger source control register controls the trigger source of the 16-bit PPG timer.

■ 16-bit PPG Trigger Source Control Register (PTGS0)

Figure 23.5-7 16-bit PPG Trigger Source Control Register (PTGS0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PTGS0	0044_H	-	-	-	SPS2	SPS1	SPS0	STS1	STS0	00000000_B
		R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	

STS1	STS0	Hardware start trigger select bits
0	0	Enables selecting the rising edge of TRG as a PPG start trigger.
0	1	Selects the rising edge of voltage comparator ch. 0 as a PPG start trigger.
1	0	Selects the falling edge of voltage comparator ch. 0 as a PPG start trigger.
1	1	Selects both edges of voltage comparator ch. 0 as a PPG start trigger.

SPS2	SPS1	SPS0	Hardware stop trigger select bits
0	0	0	Enables selecting the falling edge of TRG as a PPG stop trigger.
0	0	1	Selects the rising edge of voltage comparator ch. 1 as a PPG stop trigger.
0	1	0	Selects the falling edge of voltage comparator ch. 1 as a PPG stop trigger.
0	1	1	Selects both edges of voltage comparator ch. 1 as a PPG stop trigger.
1	0	0	Selects the rising edge of voltage comparator ch. 1/2/3* as a PPG stop trigger.
1	0	1	Selects the falling edge of voltage comparator ch. 1/2/3* as a PPG stop trigger.
1	1	0	Setting prohibited
1	1	1	Setting prohibited

R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

* : The earliest one of voltage comparator ch. 1, ch. 2 and ch. 3

Table 23.5-3 16-bit PPG Trigger Source Control Register (PTGS0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7 to bit5</td><td>Undefined bits</td><td>The read value is always "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit4 to bit2</td><td>SPS2, SPS1, SPS0: Hardware stop trigger select bits</td><td>These bits select a hardware stop trigger for the 16-bit PPG timer.
The stop trigger has higher priority than the start trigger.
<table>
<tr><th>SPS2</th><th>SPS1</th><th>SPS0</th><th>16-bit PPG stop trigger source select bits</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Enables selecting the falling edge of TRG as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Selects the rising edge of voltage comparator ch. 1 as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Selects the falling edge of voltage comparator ch. 1 as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Selects both edges of voltage comparator ch. 1 as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Selects the rising edge of voltage comparator ch. 1/2/3* as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Selects the falling edge of voltage comparator ch. 1/2/3* as a 16-bit PPG timer stop trigger.</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table>
*: The specific edge of the voltage comparator channel that occurs first will be used as the start trigger.</td></tr>
<tr><td>bit1, bit0</td><td>STS1, STS0: Hardware start trigger select bits</td><td>These bits select a hardware start trigger for the 16-bit PPG timer.
The start trigger has lower priority than the stop trigger.
<table>
<tr><th>STS1</th><th>STS0</th><th>16-bit PPG start trigger source select bits</th></tr>
<tr><td>0</td><td>0</td><td>Enables selecting the rising edge of TRG as a 16-bit PPG timer start trigger.</td></tr>
<tr><td>0</td><td>1</td><td>Selects the rising edge of voltage comparator ch. 0 as a 16-bit PPG timer start trigger.</td></tr>
<tr><td>1</td><td>0</td><td>Selects the falling edge of voltage comparator ch. 0 as a 16-bit PPG timer start trigger.</td></tr>
<tr><td>1</td><td>1</td><td>Selects both edges of voltage comparator ch. 0 as a 16-bit PPG timer start trigger.</td></tr>
</table></td></tr>
</table>

Notes:

- Before turning off the voltage comparator, disable the 16-bit PPG timer by selecting a rising/falling edge of a channel of the voltage comparator as a stop trigger, then clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF).
- After turning on the voltage comparator, clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF), then enable the 16-bit PPG timer by selecting a rising/falling edge of a channel of the voltage comparator as a start trigger.

23.6 Interrupts of 16-bit PPG Timer

The 16-bit PPG timer can generate interrupt requests in the following cases:

- **When a trigger or counter borrow occurs**
- **When a rising edge of PPG is generated in normal polarity**
- **When a falling edge of PPG is generated in inverted polarity**

The interrupt operation is controlled by IRS1 (bit3) and IRS0 (bit2) in the PCNTL register.

■ Interrupts of 16-bit PPG Timer

Table 23.6-1 shows interrupt control bits and interrupt sources of the 16-bit PPG timer.

Table 23.6-1 Interrupt Control Bits and Interrupt Sources of 16-bit PPG Timer

Item	Description
Interrupt flag bit	PCNTL0:IRQF
Interrupt request enable bit	PCNTL0:IREN
Interrupt type select bits	PCNTL0:IRS1, IRS0
Interrupt sources	PCNTL0:IRS1, IRS0=00_B Hardware trigger by TRG Pin input of 16-bit down-counter, software trigger and retrigger
	PCNTL0:IRS1, IRS0=01_B Counter borrow of 16-bit down-counter
	PCNTL0:IRS1, IRS0=10_B Rising edge of PPG1 output in normal polarity, or falling edge of PPG1 output in inverted polarity
	PCNTL0:IRS1, IRS0=11_B Counter borrow of 16-bit down-counter, rising edge of PPG1 output in normal polarity, or falling edge of PPG1 output in inverted polarity

When IRQF (bit4) in the 16-bit PPG status control register (PCNTL0) is set to "1" and interrupt requests are enabled (PCNTL0:IREN: bit5 = 1) in the 16-bit PPG timer, an interrupt request is generated and output to the controller.

■ Register and Vector Table Addresses Related to Interrupts of 16-bit PPG Timer

Table 23.6-2 Register and Vector Table Addresses Related to Interrupts of 16-bit PPG Timer

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
16-bit PPG timer ch. 1	IRQ17	ILR4	L17	$FFD8_H$	$FFD9_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

23.7 Operations of 16-bit PPG Timer and Setting Procedure Example

The 16-bit PPG timer can operate in PWM mode or one-shot mode. In addition, a retrigger function can be used in the 16-bit PPG timer.

■ PWM Mode (MDSE of PCNTH Register: bit5 = 0)

In PWM mode, the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are loaded and the 16-bit down-counter starts down-count operation when a software trigger is input or a hardware trigger by TRG pin input is input. When the count value reaches "1", the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are reloaded to repeat the down-count operation.

The initial state of the PPG output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H" synchronizing with count clock. The output changes back to "L" when the "H" was output until the value of duty setting. (The output levels will be reversed if OSEL is set to "1".)

When the retrigger function is disabled (RTRG = 0), software triggers (STRG = 1) are ignored during the operation of the down-counter.

When the down-counter is not running, the maximum time between a valid trigger input occurring and the down-counter starting is as follows.

Software trigger: 1 count clock cycle + 2 machine clock cycles

Hardware trigger by TRG Pin input: 1 count clock cycle + 3 machine clock cycles

The minimum time is as follows.

Software trigger: 2 machine clock cycles

Hardware trigger by TRG Pin input: 3 machine clock cycles

When the down-counter is running, the maximum time between a valid retrigger input occurring and the down-counter restarting is as follows.

Software trigger: 1 count clock cycle + 2 machine clock cycles

Hardware trigger by TRG Pin input: 1 count clock cycle + 3 machine clock cycles

The minimum time is as follows.

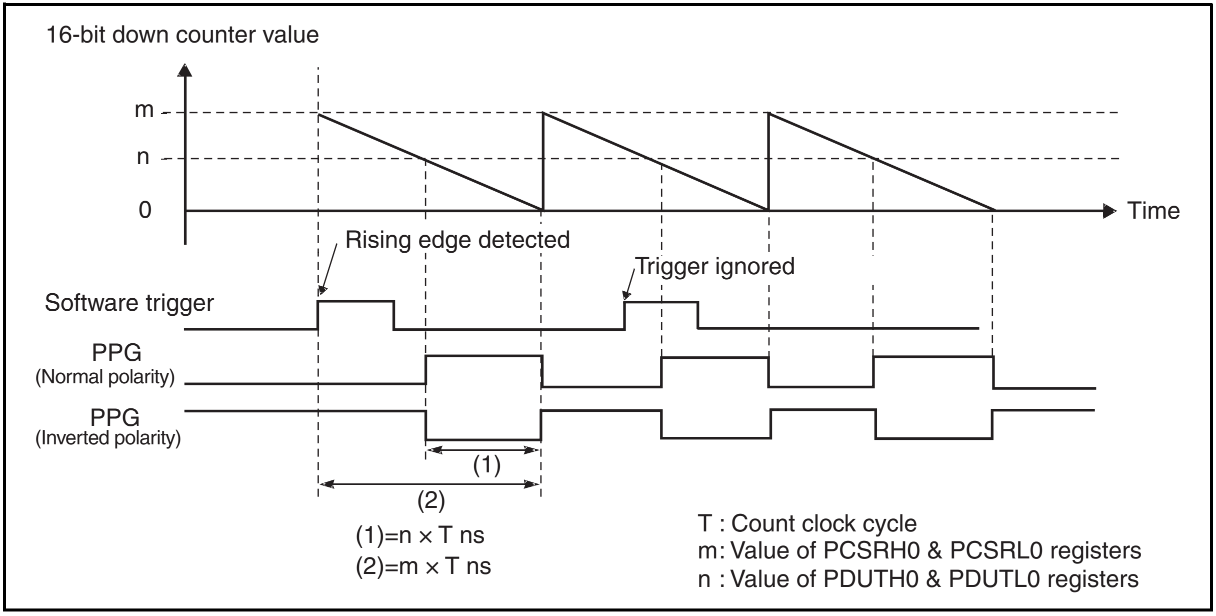
Software trigger: 2 machine clock cycles

Hardware trigger by TRG Pin input: 3 machine clock cycles

Hardware trigger by internal voltage comparator input: 3 machine clock cycles

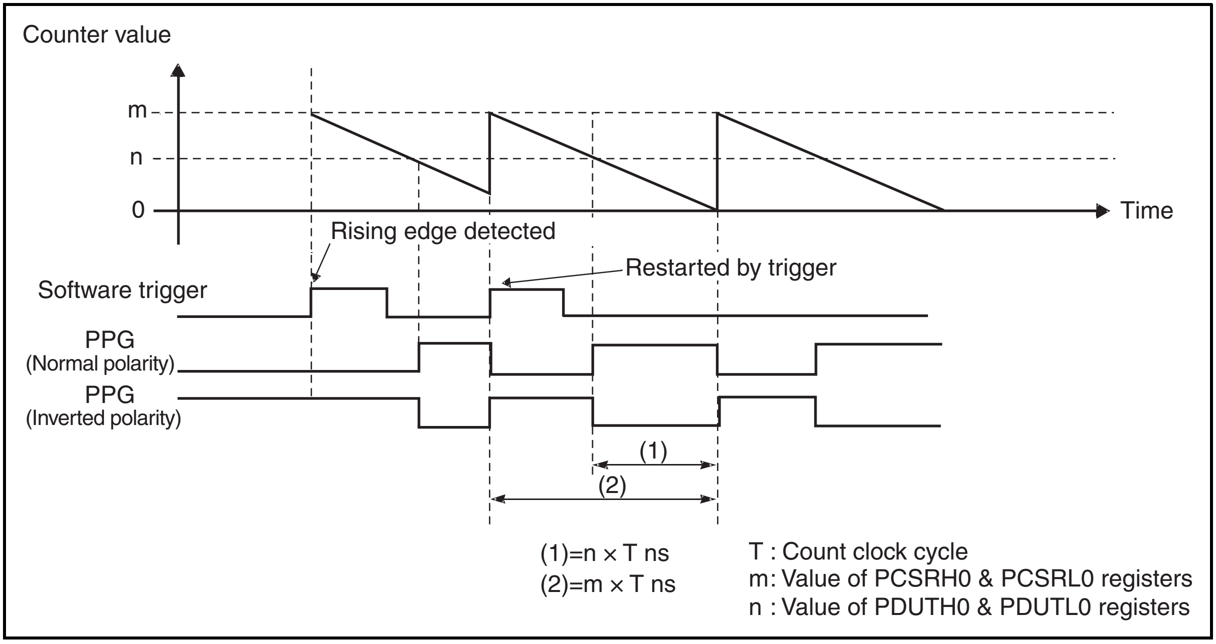
- Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 23.7-1 When Retrigger Is Invalid in PWM Mode



- Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 23.7-2 When Retrigger Is Valid in PWM Mode



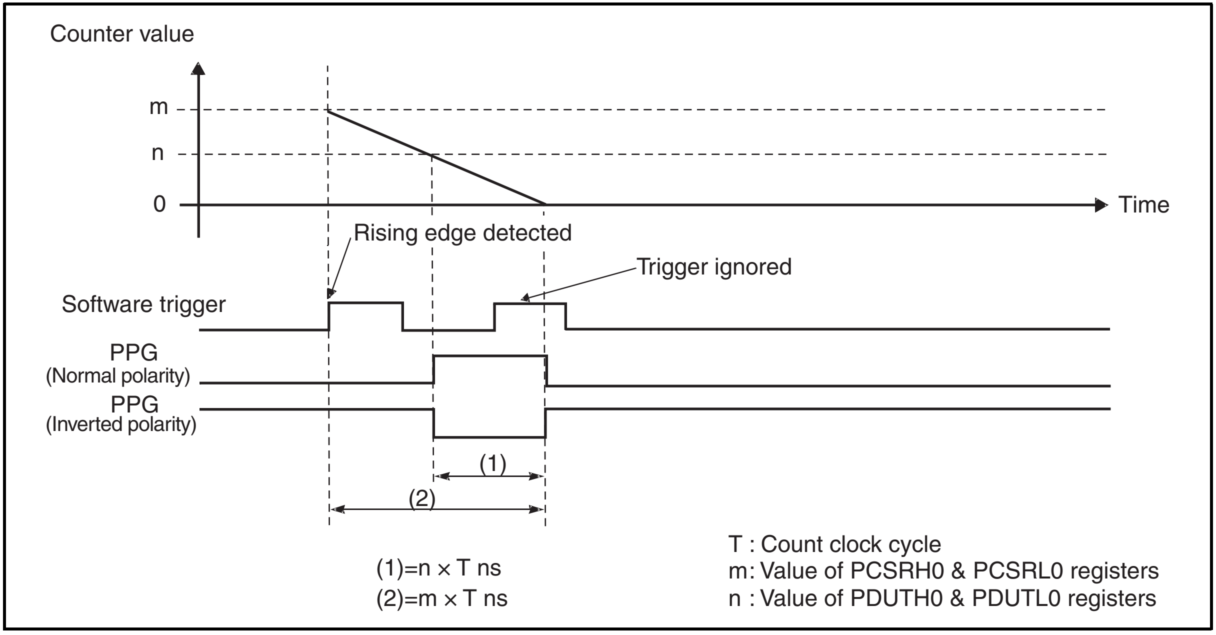
■ One-shot Mode (MDSE of PCNTH0 Register: bit5 = 1)

One-shot operation mode can be used to output a single pulse with a specified width when a valid trigger input occurs. When retriggering is enabled and a valid trigger is detected during the counter operation, the down-counter value is reloaded.

The initial state of the PPG0 output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H". The output changes back to "L" when the counter reaches "1". (The output levels will be reversed if OSEL is set to "1".)

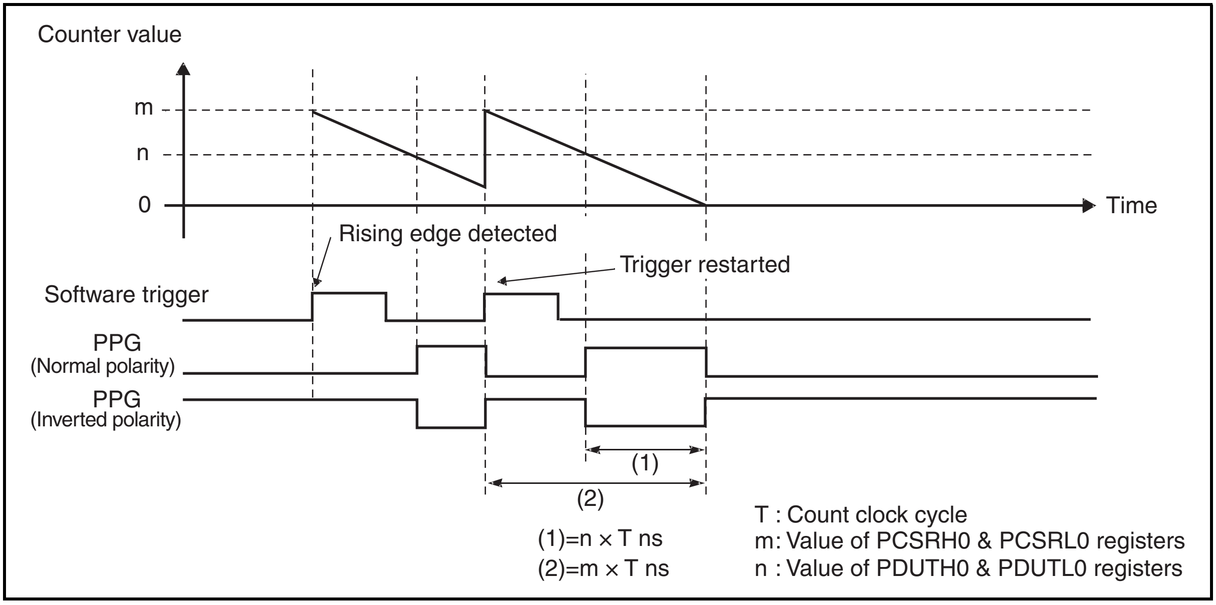
- Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 23.7-3 When Retrigger Is Invalid in One-shot Mode



- Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 23.7-4 When Retrigger Is Valid in One-shot Mode



■ Hardware Trigger

"Hardware trigger" refers to PPG activation by signal input to the TRG input pin or internal voltage comparator (VC) input.

Condition 1: PPG start triggering and stop triggering do not occur simultaneously.

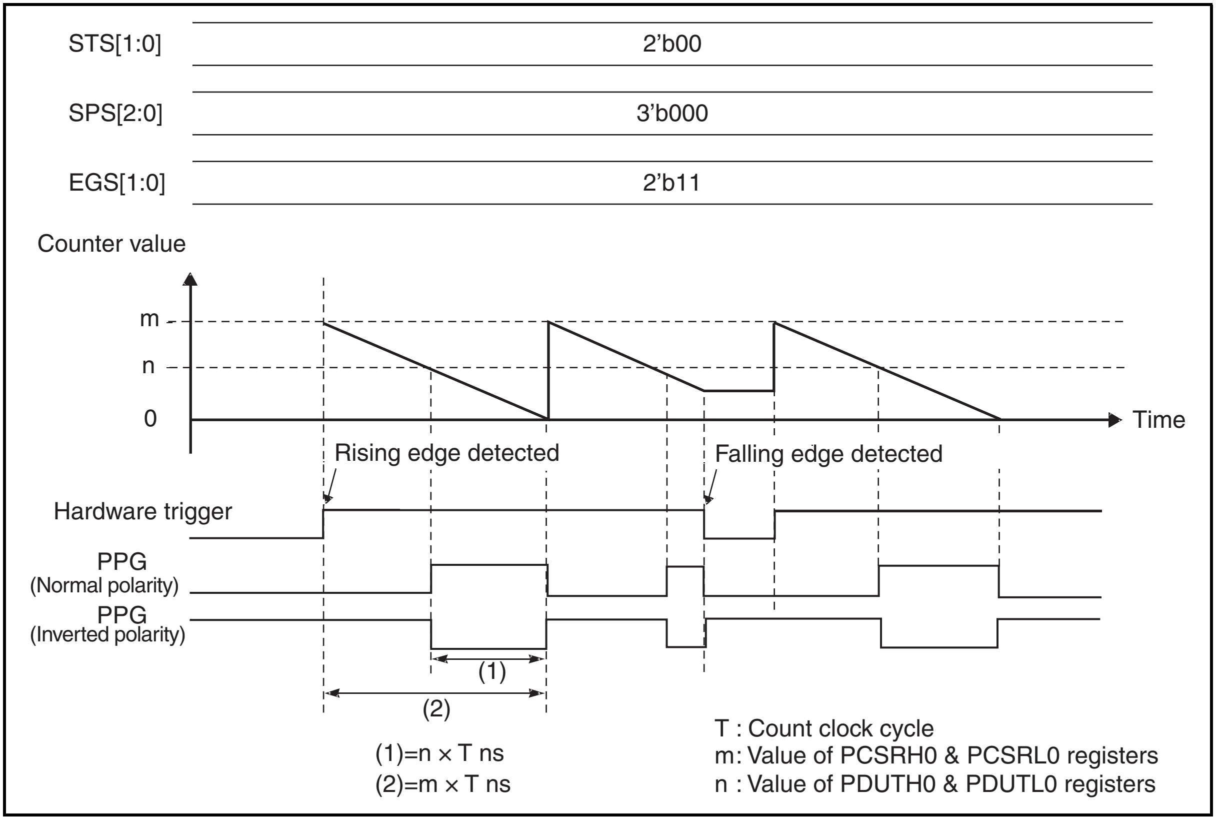
(1) TRG

When STS1 and STS0 are set to "00_B", SPS2, SPS1 and SPS0 to "000_B", and EGS1 and EGS0 to "11_B" and the hardware trigger input from TRG is used, the 16-bit PPG timer starts operating at a rising edge and stops upon the detection of a falling edge.

Moreover, the 16-bit PPG timer also starts operating at the following rising edge from the beginning.

The operation can be re-triggered by a valid TRG input hardware trigger regardless of the retrigger setting of the RTRG bit when the TRG input hardware trigger has been selected.

Figure 23.7-5 Hardware Trigger from TRG in PWM Mode



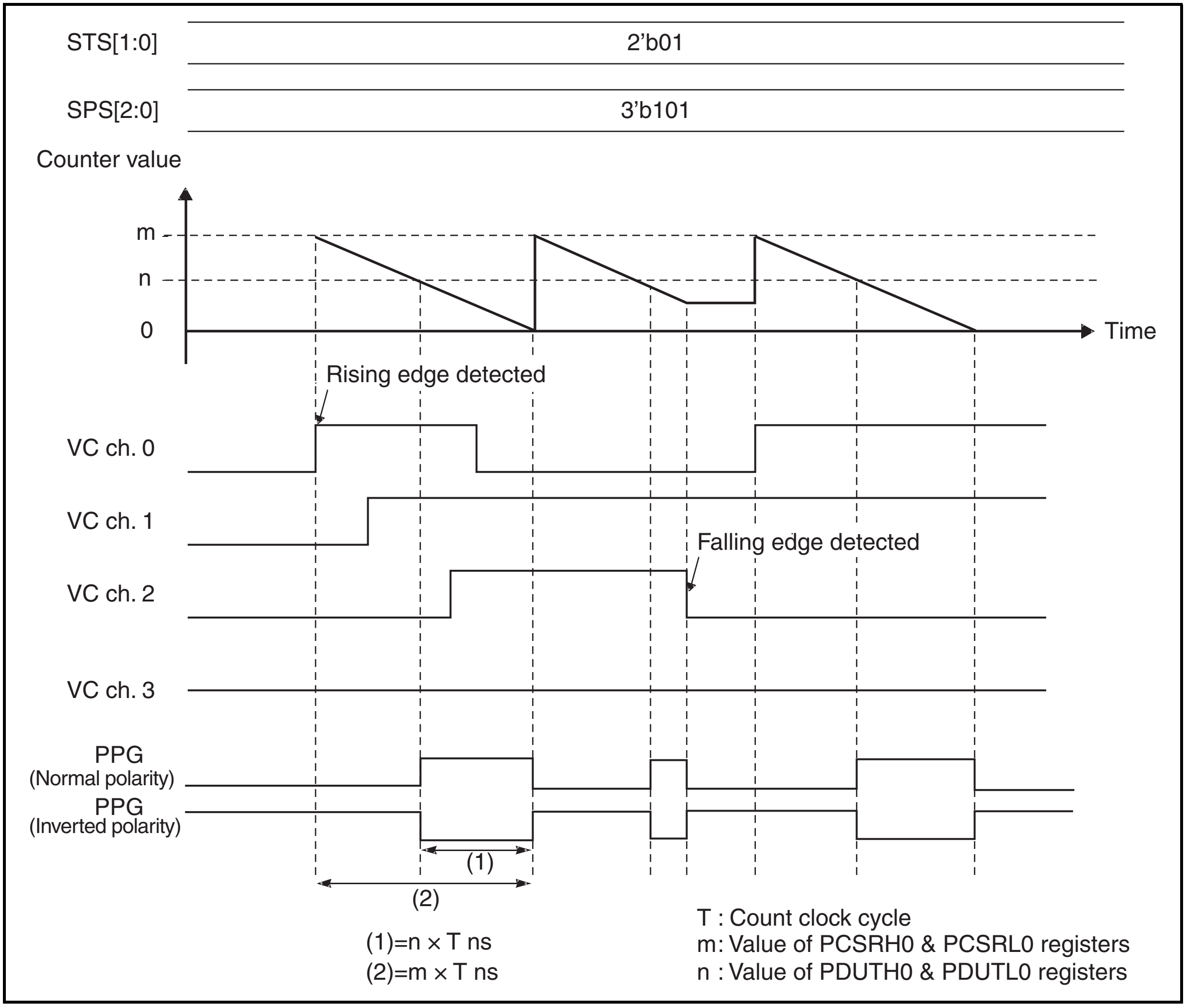
(2) Voltage comparator (VC)

When STS1 and STS0 are set to "01_B", SPS2, SPS1 and SPS0 to "101_B", and the hardware trigger input from the VC ch. 0 is used, the 16-bit PPG timer starts operating at a rising edge of VC ch. 0 and stops upon the detection of a falling edge of VC ch. 1/2/3.

Moreover, the 16-bit PPG timer also starts operating at the following rising edge of VC ch. 0 from the beginning.

The operation can be re-triggered by a valid rising edge of VC ch. 0 input hardware trigger regardless of the retrigger setting of the RTRG bit when the VC ch. 0 input hardware trigger has been selected.

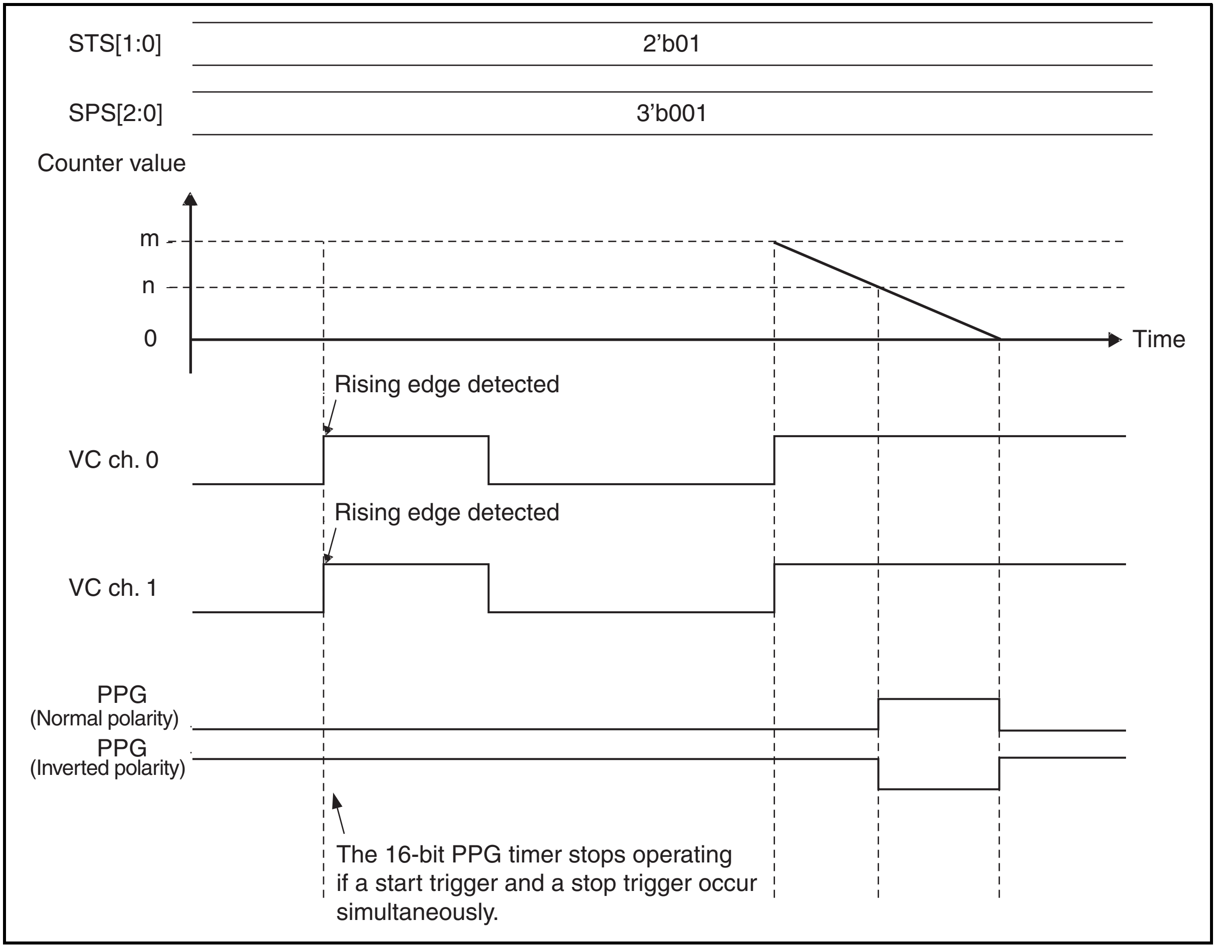
Figure 23.7-6 Hardware Trigger from VC in PWM Mode



Condition 2: PPG start triggering and stop triggering occur simultaneously.

When STS1 and STS0 are set to "01_B", SPS2, SPS1 and SPS0 to "001_B", and if the hardware trigger from a rising edge of VC ch. 0 and another rising edge of VC ch. 1 occur simultaneously, the PPG will halt (the stop trigger has higher priority than the start trigger.).

Figure 23.7-7 Hardware Triggers Start and Stop Occur Simultaneously in PWM Mode



■ Setting Procedure Example

Below is an example of procedure for setting the 16-bit PPG timer.

● Initial setup

1) Set the interrupt level. (ILR3, ILR4)
2) Enable the hardware trigger and interrupts, select the interrupt type, and enable output. (PCNTL0)
3) Select the count clock and the mode, and enable timer operation. (PCNTH0)
4) Set the cycle. (PCSRH0, PCSRL0)
5) Set the duty. (PDUTH0, PDUTL0)
6) Start the PPG by the software trigger. (PCNTH0:STRG = 1)

● Interrupt processing

1) Process any interrupt.
2) Clear the interrupt request flag. (PCNTL0:IRQF)

23.8 Notes on Using 16-bit PPG Timer

This section provides notes on using the 16-bit PPG timer.

■ Notes on Using 16-bit PPG Timer

● Notes on setting the program

Do not use the retrigger if the same values are set for the cycle and duty. If used, the PPG output will go to the "L" level for one count clock cycle after the retrigger, and then go back to the "H" level when normal polarity has been selected.

If the microcontroller enters a standby mode, the TRG pin setting may change and cause the device to malfunction. Therefore, disable the timer enable bit (PCNTH0:CNTE = 0) or disable the hardware trigger enable bit (PCNTL0:EGS1, EGS0 = 00_B).

When the cycle and duty are set to the same value, an interrupt is generated only once by duty match. Moreover, if the duty is set to a value greater than the value of the period, no interrupt will be generated by duty match.

Do not disable the timer enable bit (PCNTH0:CNTE = 0) and software trigger (PCNTH0:STRG = 1) at the same time when retrigger by the software is enabled (PCNTH0:RTRG = 1) and the retrigger is selected as an interrupt type (PCNTL0:IRS1, IRS0 = 00_B) during count operation. If it occurs, interrupt flag bit may set by retrigger although timer stops.

● Notes on using a rising/falling edge of the voltage comparator

- Before turning off the voltage comparator, disable the 16-bit PPG timer by selecting a rising/falling edge of a channel of the voltage comparator as a stop trigger, then clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF).
- After turning on the voltage comparator, clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF), then enable the 16-bit PPG timer by selecting a rising/falling edge of a channel of the voltage comparator as a start trigger.

23.9 Sample Settings for 16-bit PPG Timer

This section provides sample settings for the 16-bit PPG timer.

■ Sample Settings

● How to set the PPG operation mode

The operation mode select bit (PCNTH0:MDSE) is used.

Operation mode	Operation mode select bit (MDSE)
PWM mode	Set the bit to "0".
One-shot mode	Set the bit to "1".

● How to select the operating clock

The operating clock select bits (PCNTH0:CKS2/CKS1/CKS0) are used to select the clock.

● How to enable/disable the PPG output pin

The output enable bit (PCNTL0:POEN) is used.

Operation	Output enable bit (POEN)
To enable PPG output	Set the bit to "1".
To disable PPG output	Set the bit to "0".

● How to enable/disable PPG operation

The timer enable bit (PCNTH0:CNTE) is used.

Operation	Timer enable bit (CNTE)
To disable PPG operation	Set the bit to "0".
To enable PPG operation	Set the bit to "1".

Enable PPG operation before starting the PPG.

● How to start PPG operation by software

The software trigger bit (PCNTH0:STRG) is used.

Operation	Software trigger bit (STRG)
To start PPG operation with software	Set the bit to "1".

- How to enable/disable the retrigger function of the software trigger

The retrigger enable bit (PCNTH0:RTRG) is used.

Operation	Retrigger enable bit (RTRG)
To enable retrigger function	Set the bit to "1".
To disable retrigger function	Set the bit to "0".

- How to start operation on a rising edge of trigger input from TRG

Set PTGS0.STS1 and PTGS0.STS0 to "00_B".

The hardware trigger enable bit (PCNTL0:EGS0) is used.

Operation	Hardware trigger enable bit (EGS0)
The rising edge of TRG is active.	Set the bit to "1".
The rising edge of TRG is not used.	Set the bit to "0".

- How to stop operation on a falling edge of trigger input from TRG

Set PTGS.SPS2, PTGS0.SPS1 and PTGS0.SPS0 to "000_B".

The hardware trigger enable bit (PCNTL0:EGS1) is used.

Operation	Hardware trigger enable bit (EGS1)
The falling edge of TRG is active.	Set the bit to "1".
The falling edge of TRG is not used.	Set the bit to "0".

- How to select a hardware trigger source

The hardware trigger source select bits (PTGS0.SPS2, PTGS0.SPS1, PTGS0.SPS0, PTGS0.STS1, PTGS0.STS0) are used.

Interrupt source	Hardware trigger select bits (PTGS0.SPS[2:0], PTGS0.STS[1:0])
To enable selecting TRG* only	Set the bits to "00000_B".
To select an internal VC trigger	Set the bits to "00101_B".
To select an internal VC trigger and also enable selecting TRG	Set the bits to "00100_B".

*: If using TRG as a start trigger, PCNTH0.EGS0 must be set to "1".
If using TRG as a stop trigger, PCNTH0.EGS1 must be set to "1".

● How to invert PPG output

The output inversion bit (PCNTL0:OSEL) is used.

Operation	Output inversion bit (OSEL)
To invert PPG output	Set the bit to "1".

● How to set the PPG output to the "H" or "L" level

The PPG output mask enable bit (PCNTH0:PGMS) and the output inversion bit (PCNTL0:OSEL) are used.

Operation	PPG output mask enable bit (PGMS)	Output inversion bit (OSEL)
To set output to "H" level	Set the bit to "1"	Set the bit to "1".
To set output to "L" level	Set the bit to "1"	Set the bit to "0".

● How to select an interrupt source

The interrupt select bits (PCNTL0:IRS1/IRS0) are used to select the interrupt source.

Interrupt source	Interrupt select bits (IRS1/IRS0)
Trigger by input, software trigger, or retrigger	Set the bits to "00_B".
Counter borrow	Set the bits to "01_B".
Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	Set the bits to "10_B".
Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	Set the bits to "11_B".

● Interrupt-related registers

The interrupt level is set by the level setting registers shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
ch. 1	Interrupt level register (ILR4) Address: $0007D_H$	#17 Address: $0FFD8_H$

● How to enable/disable/clear interrupts

The interrupt request enable bit (PCNTL0:IREN) is used to enable interrupts.

Operation	Interrupt request enable bit (IREN)
To disable interrupt requests	Set the bit to "0".

Operation	Interrupt request enable bit (IREN)
To enable interrupt requests	Set the bit to "1".

The interrupt request flag (PCNTL0:IRQF) is used to clear an interrupt request.

Operation	Interrupt request flag (IRQF)
To clear an interrupt request	Set this bit to "0".

CHAPTER 24
BUZZER OUTPUT

This chapter describes the functions and operations of the buzzer output.

24.1 Overview of Buzzer Output

The buzzer output has seven output frequency (square wave) options. It can be used for applications such as sounding a buzzer to confirm key input.

■ Overview of Buzzer Output

- The buzzer output outputs a signal (square wave) suitable for applications such as sounding a buzzer to confirm an operation.
- The buzzer output frequency can be selected from seven output frequencies. The buzzer output can be disabled.
- There are seven buzzer output signal options: four from the time-base timer and three from the watch prescaler.

Notes:

- Since frequencies output from the time-base timer and the watch prescaler are input to the buzzer as buzzer input signals, the buzzer output will be cleared when a selected signal source (time-base timer or watch prescaler) is cleared.
- Since the time-base timer stops when the main clock or the main CR clock stops (in subclock mode or sub-CR clock mode), do not select a frequency output from the time-base timer as a buzzer output frequency when using the subclock mode or the sub-CR clock mode.

Table 24.1-1 Output Frequencies

Clock supplier	**Main clock mode**		**Main CR clock mode**	
	Buzzer output cycle	**Square wave output (at 4.2 MHz)**	**Buzzer output cycle**	**Square wave output (at 8 MHz)**
Time-base timer	$2^{12}/F_{CH}$	$F_{CH}/2^{12}$ (1.025 kHz)	$2^{11}/F_{CRH}$	$F_{CRH}/2^{11}$ (3.906 kHz)
	$2^{11}/F_{CH}$	$F_{CH}/2^{11}$ (2.051 kHz)	$2^{10}/F_{CRH}$	$F_{CRH}/2^{10}$ (7.812 kHz)
	$2^{10}/F_{CH}$	$F_{CH}/2^{10}$ (4.102 kHz)	$2^{9}/F_{CRH}$	$F_{CRH}/2^{9}$ (15.625 kHz)
	$2^{9}/F_{CH}$	$F_{CH}/2^{9}$ (8.203 kHz)	$2^{8}/F_{CRH}$	$F_{CRH}/2^{8}$ (31.25 kHz)
Watch prescaler	**Subclock mode**		**Sub-CR clock mode**	
	Buzzer output cycle	**Square wave output (at 32.768 kHz)**	**Buzzer output cycle**	**Square wave output (at 100 kHz)**
	$2^{5}/F_{CL}$	$F_{CL}/2^{5}$ (1.024 kHz)	$2^{5}/F_{CRL}$	$F_{CRL}/2^{5}$ (3.125 kHz)
	$2^{4}/F_{CL}$	$F_{CL}/2^{4}$ (2.048 kHz)	$2^{4}/F_{CRL}$	$F_{CRL}/2^{4}$ (6.25 kHz)
	$2^{3}/F_{CL}$	$F_{CL}/2^{3}$ (4.096 kHz)	$2^{3}/F_{CRL}$	$F_{CRL}/2^{3}$ (12.5 kHz)

F_{CH}: Main clock oscillation frequency F_{CRH}: Main CR clock oscillation frequency

F_{CL}: Subclock oscillation frequency F_{CRL}: Sub-CR clock oscillation frequency

- Example of frequency calculation

  If the main clock oscillation frequency is 4.2 MHz and the buzzer output selected is $F_{CH}/2^{10}$ (BZCR.BUZ[2:0] = 011_B), the output frequency of the BZ pin becomes:

  $$\begin{aligned}\text{Output frequency} &= F_{CH}/2^{10}\\ &= 4.2\ \text{MHz}/1024\\ &= \underline{\underline{4.102\ \text{kHz}}}\end{aligned}$$

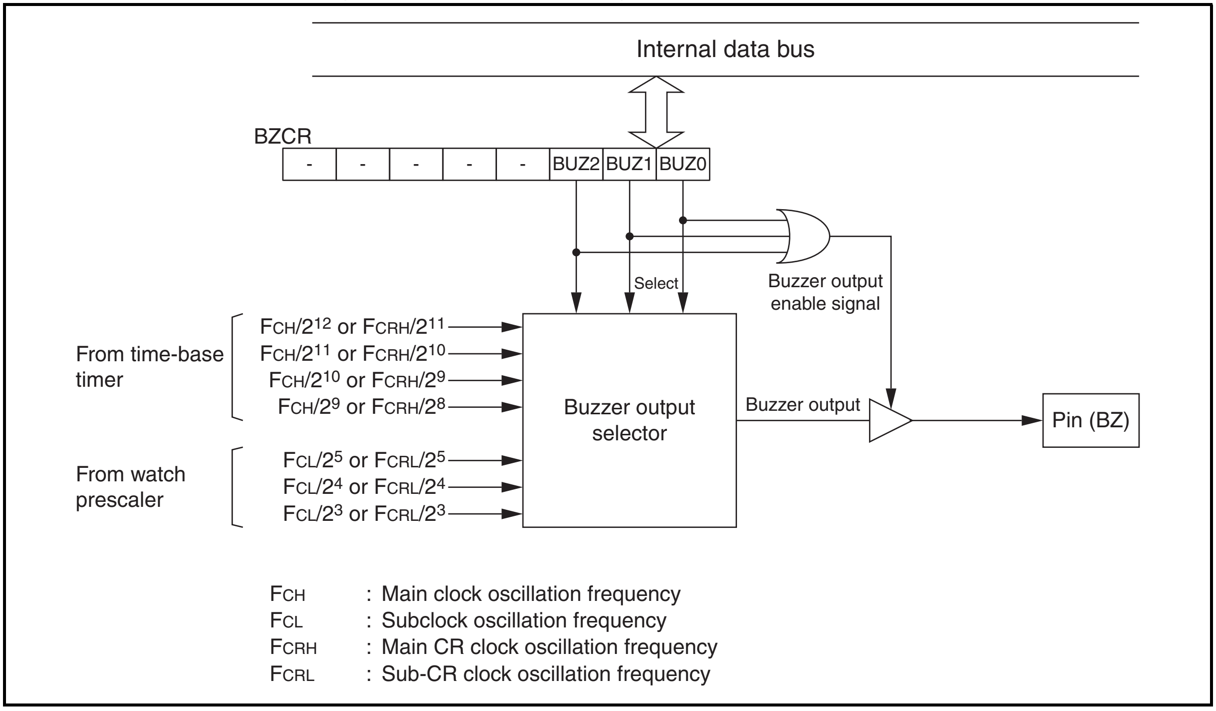
24.2 Configuration of Buzzer Output

The buzzer output consists of the following two blocks:

- **Buzzer output selector**
- **Buzzer register (BZCR)**

■ Block Diagram of Buzzer Output

Figure 24.2-1 Block Diagram of Buzzer Output


Internal data bus
BZCR
-
-
-
-
-
BUZ2
BUZ1
BUZ0
Select
Buzzer output enable signal
From time-base timer
$F_{CH}/2^{12}$ or $F_{CRH}/2^{11}$
$F_{CH}/2^{11}$ or $F_{CRH}/2^{10}$
$F_{CH}/2^{10}$ or $F_{CRH}/2^{9}$
$F_{CH}/2^{9}$ or $F_{CRH}/2^{8}$
From watch prescaler
$F_{CL}/2^{5}$ or $F_{CRL}/2^{5}$
$F_{CL}/2^{4}$ or $F_{CRL}/2^{4}$
$F_{CL}/2^{3}$ or $F_{CRL}/2^{3}$
Buzzer output selector
Buzzer output
Pin (BZ)
F_{CH} : Main clock oscillation frequency
F_{CL} : Subclock oscillation frequency
F_{CRH} : Main CR clock oscillation frequency
F_{CRL} : Sub-CR clock oscillation frequency

● Buzzer output selector

The buzzer output selector is a circuit for selecting an output frequency (square wave) from seven options, four from the time-base timer and three from the watch prescaler. The buzzer register (BZCR) is used to select the output frequency.

● Buzzer selector (BZCR)

The buzzer register (BZCR) is a register for selecting the buzzer output frequency and enabling the buzzer output. When the BZCR register is set to a value (except "000_B") to select an output frequency, the buzzer output is automatically enabled and the PG1/TRG/ADTG/X0A/BZ/OUT0 pin also automatically becomes the buzzer output pin (BZ). In addition, the P01/AN01/INT01/BZ pin can also be used as the buzzer output pin. For details of using the P01/AN01/INT01/BZ pin as the buzzer output pin, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

24.3 Pins of Buzzer Output

This section describes the pins of the buzzer output.

■ Pins of Buzzer Output

● P01/AN01/INT01/BZ pin

The P01/AN01/INT01/BZ pin can work as a general-purpose I/O pin (P01), an external interrupt pin (INT01), an A/D converter analog input pin (AN01), or an output pin (BZ) of the buzzer output. For details of setting this pin, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

● PG1/TRG/ADTG/X0A/BZ/OUT0 pin

The PG1/TRG/ADTG/X0A/BZ/OUT0 pin can work as a general-purpose I/O pin (PG1), a 16-bit PPG trigger input pin (TRG), an A/D converter trigger input pin (ADTG), a subclock I/O oscillation pin (X0A), an output pin (BZ) of the buzzer output or an output compare ch. 0 output pin (OUT0).

● BZ pin

The BZ pin outputs to the buzzer a square wave of a frequency specified for the BZ pin. When a buzzer output frequency is specified in BZCR:BUZ[2:0] (except BZCR:BUZ[2:0] = 000_B), the PG1/TRG/ADTG/X0A/BZ/OUT0 pin automatically changes to work as the BZ pin regardless of the value of the output latch.

■ Block Diagrams of Pins of Buzzer Output

Figure 24.3-1 Block Diagram of Pin BZ (P01/AN01/INT01/BZ) of Buzzer Output

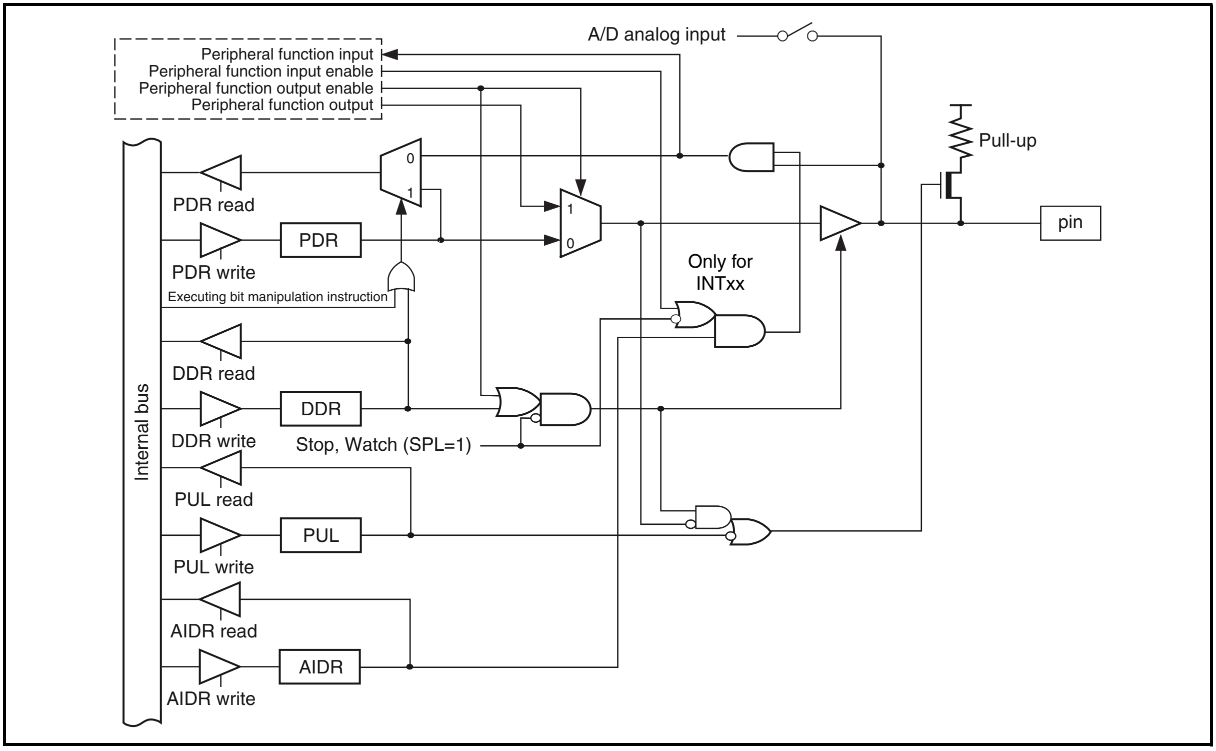
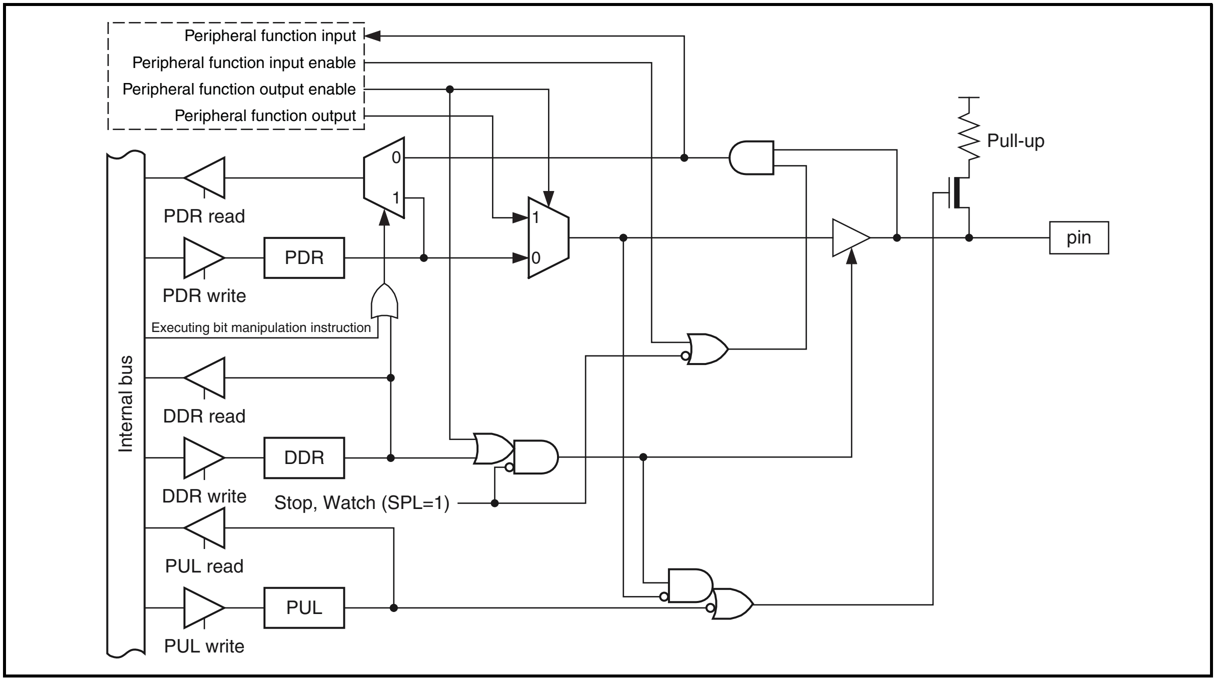


Figure 24.3-2 Block Diagram of Pin BZ (PG1/TRG/ADTG/X0A/BZ/OUT0) of Buzzer Output


Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Pull-up
pin
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
Internal bus
PUL read
PUL
PUL write

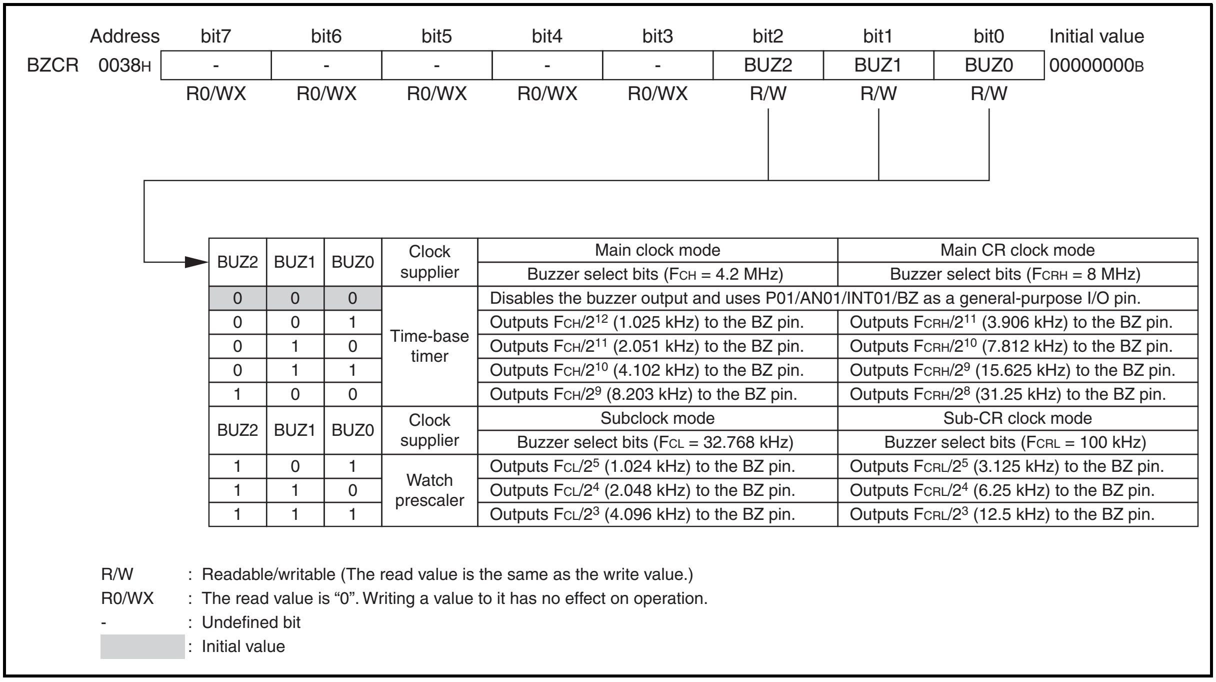
MB95430H Series

24.4 Buzzer Register (BZCR)

The buzzer register (BZCR) is used to select an output frequency of the buzzer output and to enable the buzzer output.

■ Buzzer Register (BZCR)

Figure 24.4-1 Buzzer Register (BZCR)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
BZCR	0038H	-	-	-	-	-	BUZ2	BUZ1	BUZ0	00000000B
		R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	

BUZ2	BUZ1	BUZ0	Clock supplier	Main clock mode	Main CR clock mode
				Buzzer select bits (F_{CH} = 4.2 MHz)	Buzzer select bits (F_{CRH} = 8 MHz)
0	0	0	Time-base timer	Disables the buzzer output and uses P01/AN01/INT01/BZ as a general-purpose I/O pin.	
0	0	1		Outputs $F_{CH}/2^{12}$ (1.025 kHz) to the BZ pin.	Outputs $F_{CRH}/2^{11}$ (3.906 kHz) to the BZ pin.
0	1	0		Outputs $F_{CH}/2^{11}$ (2.051 kHz) to the BZ pin.	Outputs $F_{CRH}/2^{10}$ (7.812 kHz) to the BZ pin.
0	1	1		Outputs $F_{CH}/2^{10}$ (4.102 kHz) to the BZ pin.	Outputs $F_{CRH}/2^{9}$ (15.625 kHz) to the BZ pin.
1	0	0		Outputs $F_{CH}/2^{9}$ (8.203 kHz) to the BZ pin.	Outputs $F_{CRH}/2^{8}$ (31.25 kHz) to the BZ pin.
BUZ2	BUZ1	BUZ0	Clock supplier	Subclock mode	Sub-CR clock mode
				Buzzer select bits (F_{CL} = 32.768 kHz)	Buzzer select bits (F_{CRL} = 100 kHz)
1	0	1	Watch prescaler	Outputs $F_{CL}/2^{5}$ (1.024 kHz) to the BZ pin.	Outputs $F_{CRL}/2^{5}$ (3.125 kHz) to the BZ pin.
1	1	0		Outputs $F_{CL}/2^{4}$ (2.048 kHz) to the BZ pin.	Outputs $F_{CRL}/2^{4}$ (6.25 kHz) to the BZ pin.
1	1	1		Outputs $F_{CL}/2^{3}$ (4.096 kHz) to the BZ pin.	Outputs $F_{CRL}/2^{3}$ (12.5 kHz) to the BZ pin.

R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 24.4-1 Functions of Bits in Buzzer Register (BZCR)

Bit name		Function
bit7 to bit3	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit2 to bit0	BUZ2, BUZ1, BUZ0: Buzzer select bits	These bits select a buzzer output frequency and enables the buzzer output. • If these bits are set to "000_B", the buzzer output is disabled. • If these bits are set to any value other than "000_B", the P01/AN01/INT01/BZ pin becomes the buzzer output pin and will output square waves. For details of using the P01/AN01/INT01/BZ pin as the buzzer output pin, see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER". • There are seven buzzer output signal options: four from the time-base timer and three from the watch prescaler. Notes: • Do not select a time-base timer output in subclock mode or sub-CR clock mode. • The subclock or the sub-CR clock can operate in main stop mode (when SYCC2:SOSCE or SYCC2:SCRE is set to "1"). Therefore, if the pin state setting bit in the STBC register (STBC:SPL) is set to "0", the buzzer output can be used even in main stop mode provided that a frequency output from the watch prescaler (BUZ[2:0] = 101_B, 110_B or 111_B) has been selected.

24.5 Sample Program for Buzzer Output

This section shows a sample program for the buzzer output.

■ Sample Program for Buzzer Output

● Processing specification

- To output a buzzer output frequency of approximately 1.025 kHz from the BZ pin and then disable the buzzer output.
- With the main clock oscillation frequency being 4.2 MHz, set BZCR:BUZ[2:0] to "001_B" to select $F_{CH}/2^{12}$. The buzzer output frequency becomes:

$$\begin{aligned}\text{Buzzer output frequency} &= 4.2\ \text{MHz}/2^{12}\\ &= 4.2\ \text{MHz}/4096\\ &= \underline{\underline{1.025\ \text{kHz}}}\end{aligned}$$

● Coding example

```
BZCR      EQU     0038H            ;Buzzer register
;-----------------------------------------------------------------------------------------------------------
CSEG                                  ;[CODE SEGMENT]
:
BUZON     MOV     BZCR,#00000001B     ;Buzzer output "ON".
:
:
:
BUZOFF    MOV     BZCR,#00000001B     ;Buzzer output "OFF".
:
ENDS
;-----------------------------------------------------------------------------------------------------------
END
```

CHAPTER 25 OPERATIONAL AMPLIFIER

This chapter describes the functions and operations of the operational amplifier.

25.1 Overview of Operational Amplifier

The operational amplifier can be used to sense the ground current, and support front-end analog signal conditioning prior to A/D conversion. It can operate in either closed loop mode or standalone open loop mode.

■ Closed Loop Mode

The operational amplifier can be configured as a non-inverting closed loop operational amplifier.

It has six software-selectable closed loop gain options for ground current sensing according to different sense voltage values.

Table 25.1-1 Software-selectable Closed Loop Gains of Operational Amplifier

No.	Gain
1	10 V/V
2	20 V/V
3	30 V/V
4	40 V/V
5	50 V/V
6	60 V/V

■ Standalone Open Loop Mode

In this mode, the operational amplifier input pins are connected to external signals without any output feedback.

The standalone open loop mode is designed for users that can choose more flexible gain using external resistors.

Note:

- In closed loop mode, connecting the P61/OPAMP_N pin to the ground is recommended.
- In any of the standby modes of this device, the operational amplifier module can be made to continue operating.

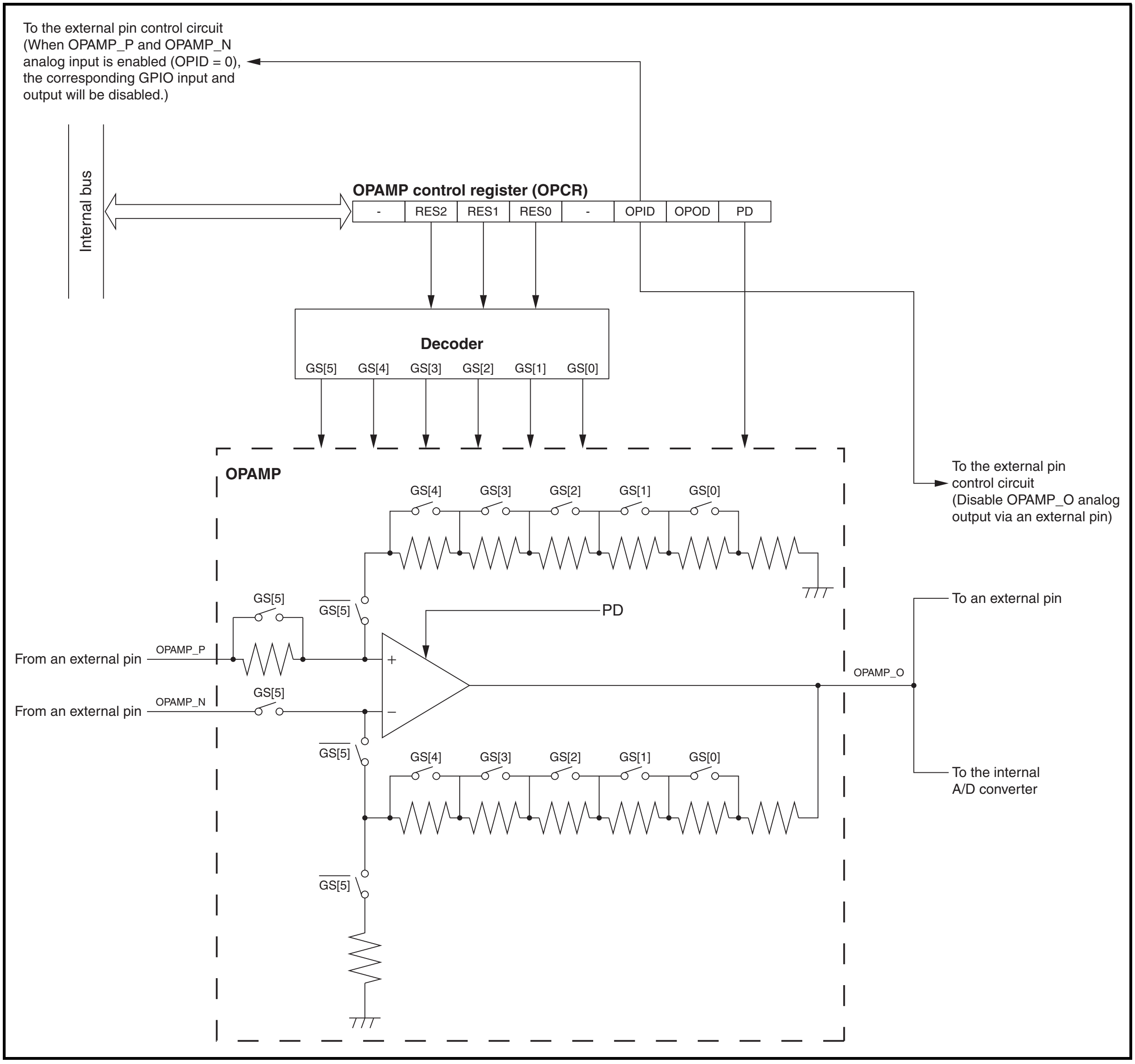
25.2 Configuration of Operational Amplifier

The operational amplifier consists of the following two blocks:

- **OPAMP**
- **OPAMP control register (OPCR)**

■ Block Diagram of Operational Amplifier

Figure 25.2-1 Block Diagram of Operational Amplifier



For details of A/D conversion of the OPAMP_O signal, see "CHAPTER 20 8/10-BIT A/D CONVERTER".

● OPAMP

The OPAMP uses the OPAMP_P pin and the OPAMP_N pin as the analog input pins of the operational amplifier, and uses the OPAMP_O pin as the analog output pin of the operational amplifier.

When GS[5] is set to "1B" and GS[4:0] are set to "00000B", the OPAMP will work as a standalone open loop operational amplifier.

When GS[5] is set to "0B", the OPAMP will work as a non-inverting closed loop operational amplifier. It provides six different closed loop gain settings through the software.

● OPAMP control register (OPCR)

The OPAMP control register is used to turn on or off the OPAMP (using OPCR:PD), to enable and disable OPAMP analog output (using OPCR:OPOD), and to enable and disable OPAMP analog input (OPCR:OPID).

The register can also be used to set the OPAMP to operate as a standalone open loop operational amplifier, or a non-inverting closed loop operational amplifier with six different closed loop gain settings that can be selected by the software (using OPCR:RES[2:0]).

25.3 Pins of Operational Amplifier

This section describes the pins of the operational amplifier.

■ Pins of Operational Amplifier

Table 25.3-1 Pins of Operational Amplifier

Pin Name	Pin Function	I/O Type	Pull-up Option	Standby Control	Settings Required for Using The Pin	Default Status

P60/OPAMP_P	GPIO/
	OPAMP positive analog input
		CMOS input/
		CMOS output/
		Analog input
			Unavailable	Available	OPCR:OPID = 0
					(Enables analog input)
						GPIO input disabled;
						GPIO output disabled;
						analog input enabled

P61/OPAMP_N	GPIO/
	OPAMP negative analog input
		CMOS input/
		CMOS output/
		Analog input
					OPCR:OPID = 0
					(Enables analog input)
						GPIO input disabled;
						GPIO output disabled;
						analog input enabled

P62/OPAMP_O	GPIO/
	OPAMP analog output
		CMOS input/
		CMOS output/
		Analog output
					OPCR:OPOD = 0
					(Enables analog output)
						GPIO input enabled;
						GPIO output disabled;
						analog output disabled

Note:

- In closed loop mode, connecting the P61/OPAMP_N pin to the ground is recommended.

■ Block Diagrams of Pins of Operational Amplifier

Figure 25.3-1 Block Diagram of Pins OPAMP_P and OPAMP_N (P60/OPAMP_P and P61/OPAMP_N) of Operational Amplifier

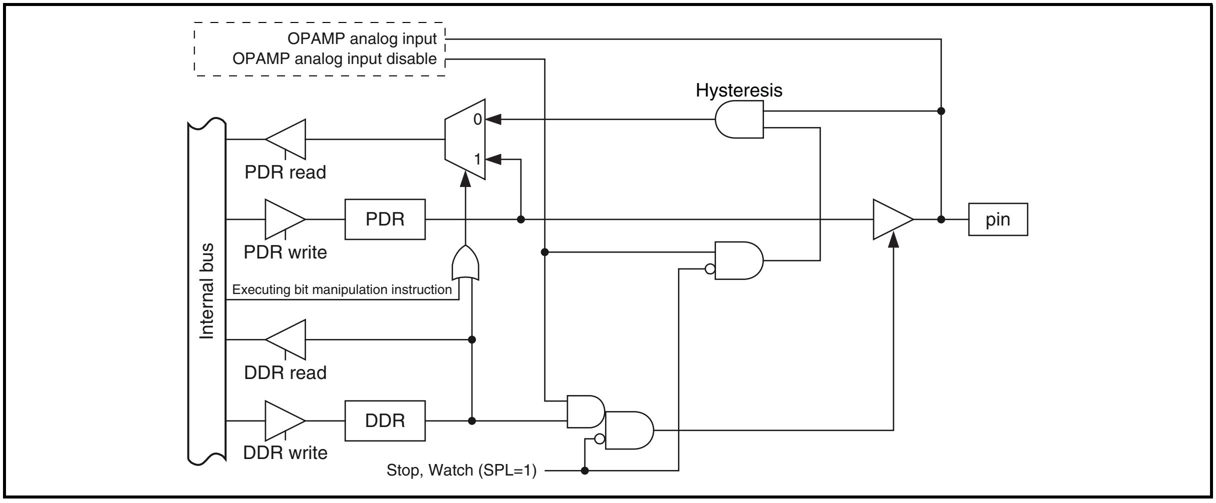
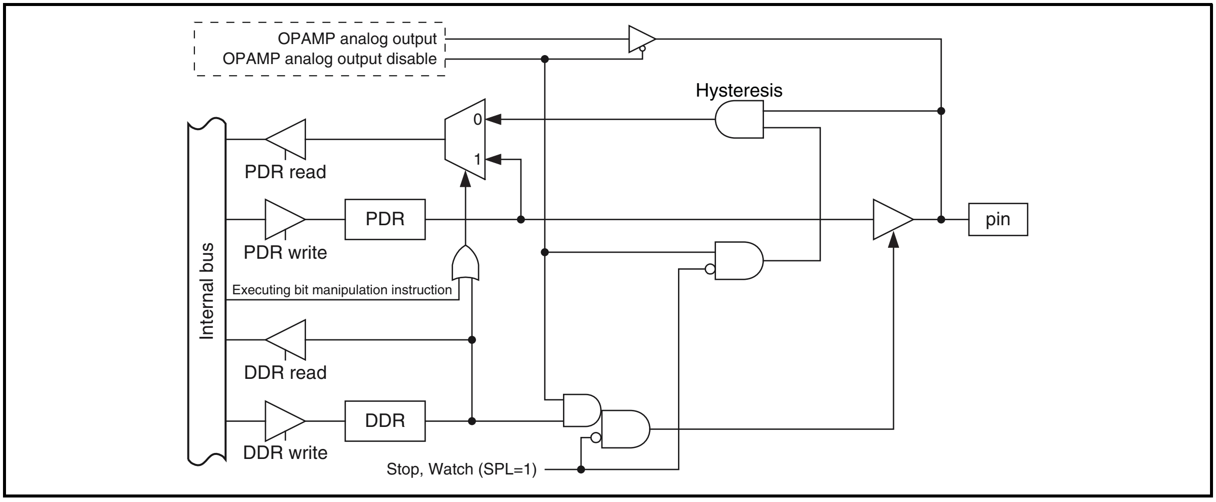


Figure 25.3-2 Block Diagram of Pin OPAMP_O (P62/OPAMP_O) of Operational Amplifier



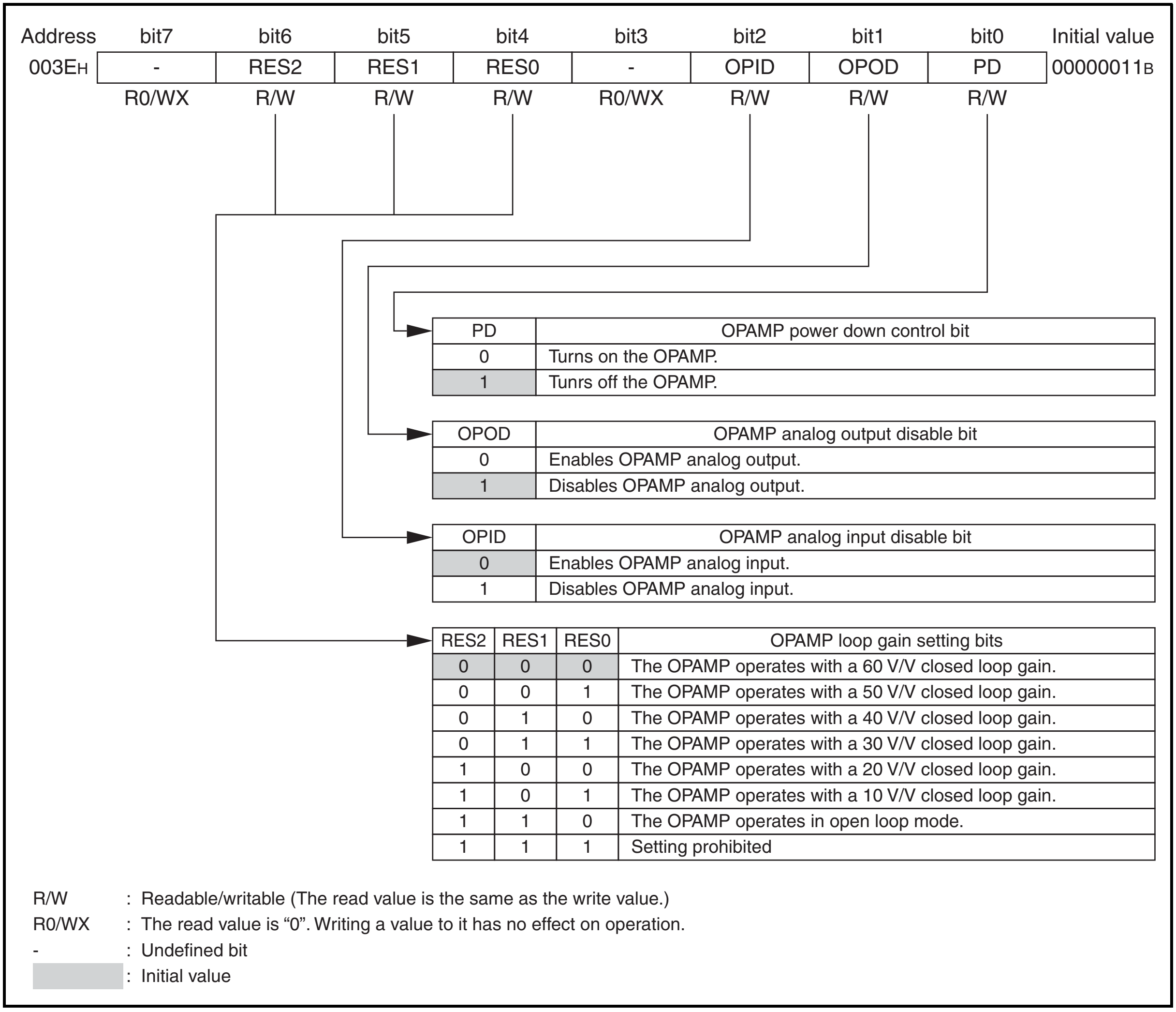
25.4 OPAMP Control Register (OPCR)

The OPAMP control register (OPCR) is used to turn on and off the OPAMP, to enable and disable OPAMP analog output, and to enable and disable OPAMP analog input.

The register can also be used to set the OPAMP to operate as a standalone open loop operational amplifier, or a non-inverting closed loop operational amplifier with six different closed loop gain settings that can be selected by the software.

■ OPAMP Control Register (OPCR)

Figure 25.4-1 OPAMP Control Register (OPCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
003EH	-	RES2	RES1	RES0	-	OPID	OPOD	PD	00000011B
	R0/WX	R/W	R/W	R/W	R0/WX	R/W	R/W	R/W	

PD	OPAMP power down control bit
0	Turns on the OPAMP.
1	Tunrs off the OPAMP.

OPOD	OPAMP analog output disable bit
0	Enables OPAMP analog output.
1	Disables OPAMP analog output.

OPID	OPAMP analog input disable bit
0	Enables OPAMP analog input.
1	Disables OPAMP analog input.

RES2	RES1	RES0	OPAMP loop gain setting bits
0	0	0	The OPAMP operates with a 60 V/V closed loop gain.
0	0	1	The OPAMP operates with a 50 V/V closed loop gain.
0	1	0	The OPAMP operates with a 40 V/V closed loop gain.
0	1	1	The OPAMP operates with a 30 V/V closed loop gain.
1	0	0	The OPAMP operates with a 20 V/V closed loop gain.
1	0	1	The OPAMP operates with a 10 V/V closed loop gain.
1	1	0	The OPAMP operates in open loop mode.
1	1	1	Setting prohibited

R/W	:	Readable/writable (The read value is the same as the write value.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	Initial value

Table 25.4-1 Functions of Bits in OPAMP Control Register (OPCR)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit6 to bit4	RES2, RES1, RES0: OPAMP loop gain setting bits	These bits select an OPAMP loop gain in closed loop mode from six options and can set the OPAMP to operate in open loop mode.
bit3	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit2	OPID: OPAMP analog input disable bit	This bit is used to enable or disable OPAMP analog input. **Writing "0":** Enables OPAMP analog input. **Writing "1":** Disables OPAMP analog input.
bit1	OPOD: OPAMP analog output disable bit	This bit is used to enable or disable OPAMP analog output. **Writing "0":** Enables OPAMP analog output. **Writing "1":** Disables OPAMP analog output.
bit0	PD: OPAMP power down control bit	This bit is used to turn on or off the OPAMP. **Writing "0":** Turns on the OPAMP. **Writing "1":** Turns off the OPAMP.

Note:

The OPAMP can be set to operate in open loop mode or closed loop mode according to the settings of the RES[2:0] bits. In closed loop mode, six different closed loop gains can be selected.

Table 25.4-2 OPAMP Operating Mode Settings

RES2	RES1	RES0	OPAMP Loop Gain Settings
0	0	0	The OPAMP operates with a 60 V/V closed loop gain.
0	0	1	The OPAMP operates with a 50 V/V closed loop gain.
0	1	0	The OPAMP operates with a 40 V/V closed loop gain.
0	1	1	The OPAMP operates with a 30 V/V closed loop gain.
1	0	0	The OPAMP operates with a 20 V/V closed loop gain.
1	0	1	The OPAMP operates with a 10 V/V closed loop gain.
1	1	0	The OPAMP operates in open loop mode.
1	1	1	Setting prohibited

Notes:

- While the OPAMP is operating, modifying the settings of RES2, RES1 and RES0 is allowed, however, do not use the output signal of the OPAMP or execute A/D conversion until OPAMP output becomes stable.
- It is recommended to turn off the operational amplifier before modifying the settings of RES2, RES1 and RES0.

25.5 Operations of Operational Amplifier

The operational amplifier can be activated by setting the PD bit in the OPCR register using the software. It can operate in closed loop mode or open loop mode, depending on the settings of the RES2, RES1 and RES0 bits in the OPCR register.

■ Activating Operational Amplifier by Software

The settings shown in Figure 25.5-1 are required for activating the operational amplifier using the software.

Figure 25.5-1 Settings for Activating Operational Amplifier

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OPCR	-	RES2	RES1	RES0	-	OPID	OPOD	PD
	×	❍	❍	❍	×	0	0	0

❍ : Bit to be used
× : Unused bit
0 : Set to "0"

After the operational amplifier is activated as shown above, it has to stabilize before starting to operate.

■ Operations of OPAMP in Closed Loop Mode

Before being activated, the operational amplifier can be set to operate in closed loop mode in advance by setting the RES[2:0] bits in the OPCR register to "000_B", "001_B", "010_B", "011_B", "100_B" or "101_B".

Six different closed loop gains are available to be used in closed loop mode. Select a desired closed loop gain by setting RES[2:0] in OPCR to the value corresponding to that gain.

Notes:

- In closed loop mode, connecting the P61/OPAMP_N pin to the ground is recommended.
- While the OPAMP is operating, modifying the settings of RES2, RES1 and RES0 is allowed; however, do not use the output signal of the OPAMP or execute A/D conversion until OPAMP output becomes stable.
- It is recommended to turn off the operational amplifier before modifying the settings of RES2, RES1 and RES0.

■ Operations of OPAMP in Open Loop Mode

Before being activated, the operational amplifier can be set to operate in open loop mode in advance by setting the RES[2:0] bits in the OPCR register to "110_B".

Note:

- While the OPAMP is operating, switching it from closed loop mode to open loop mode, and vice versa, is allowed, however, do not use the output signal of the OPAMP or execute A/D conversion until OPAMP output becomes stable.

■ Setting Procedure Example

Below is an example of procedure for setting the operational amplifier:

● Initial settings

1) Set both OPCR:OPID and OPCR:OPOD to "0" to enable both OPAMP analog input and OPAMP analog output.
2) Set the feedback resistor and RES[2:0] in OPCR.
3) Set OPCR:PD to "0" to turn on the operational amplifier.
4) Wait until the operation amplifier stabilizes.
5) Start A/D conversion if necessary.

For details of A/D conversion of the OPAMP_O signal, see "CHAPTER 20 8/10-BIT A/D CONVERTER".

CHAPTER 26

VOLTAGE COMPARATOR

This chapter describes the functions and operations of the voltage comparator.

26.1 Overview of Voltage Comparator

The voltage comparator is used to monitor the voltages of two external analog inputs, and can automatically generate an interrupt upon detection of a change in the edge of voltage comparator output.

■ Function of Voltage Comparator

The function of the voltage comparator is to monitor the voltages of two external analog inputs and compare them. Using the voltage of the positive analog input used as a reference voltage, the voltage comparator will output "H" if the voltage of the negative analog input is lower than the reference voltage; otherwise, it will output "L". In addition, upon detection of a rising edge or falling edge of the voltage comparator output, the voltage comparator outputs a corresponding interrupt.

MB95430H Series

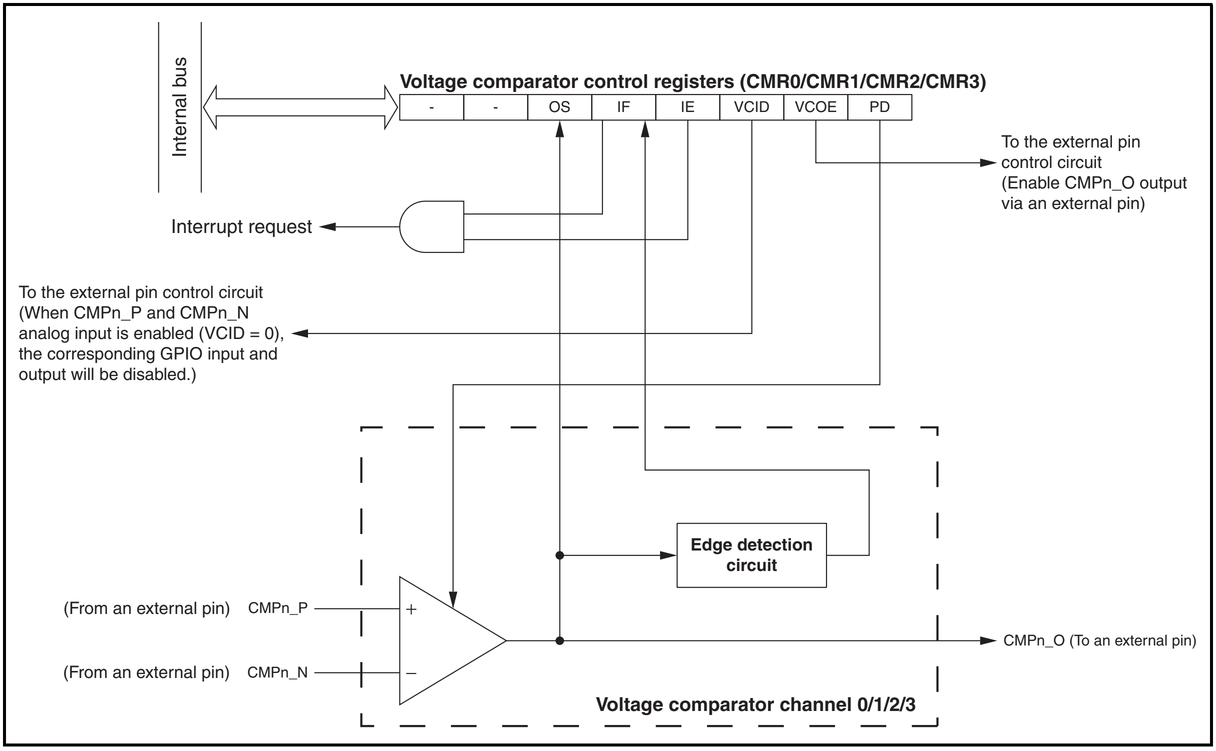
26.2 Configuration of Voltage Comparator

The entire voltage comparator module consists of the following blocks:

- **Voltage comparator × 4 channels**
- **Edge detection circuit × 4 channels**
- **Voltage comparator control registers × 4 channels (CMR0/CMR1/CMR2/ CMR3)**

■ Block Diagram of Voltage Comparator

Figure 26.2-1 Block Diagram of Voltage Comparator


Internal bus
Voltage comparator control registers (CMR0/CMR1/CMR2/CMR3)
-
-
OS
IF
IE
VCID
VCOE
PD
To the external pin control circuit (Enable CMPn_O output via an external pin)
Interrupt request
To the external pin control circuit (When CMPn_P and CMPn_N analog input is enabled (VCID = 0), the corresponding GPIO input and output will be disabled.)
Edge detection circuit
(From an external pin) CMPn_P
(From an external pin) CMPn_N
+
–
CMPn_O (To an external pin)
Voltage comparator channel 0/1/2/3

● Voltage comparator

The voltage comparator monitors the voltages of two external analog inputs and compares them. Using the voltage of the positive analog input used as a reference voltage, the voltage comparator will output "H" if the voltage of the negative analog input is lower than the reference voltage; otherwise, it will output "L".

● Edge detection circuit

Except in stop mode, watch mode or time-base timer mode, upon detection of a rising edge or falling edge of voltage comparator output, the edge detection circuit automatically raises an interrupt flag an interrupt flag (CMR0/CMR1/CMR2/CMR3:IF).

● Voltage comparator control registers (CMR0/CMR1/CMR2/CMR3)

These registers are used to turn on and off the voltage comparator (CMR0/CMR1/CMR2/CMR3:PD), to enable and disable voltage comparator output (CMR0/CMR1/CMR2/CMR3:VCOE), and to enable and disable voltage comparator analog inputs (CMR0/CMR1/CMR2/CMR3:VCID).

Except in stop mode, watch mode or time-base timer mode, if CMR0/CMR1/CMR2/CMR3:IE has been set to "1", upon detection of a rising edge or falling edge of voltage comparator output, the voltage comparator generates an interrupt request and the interrupt flag bit (CMR0/CMR1/CMR2/CMR3:IF) is automatically set to "1" at the same time.

The output status of a voltage comparator channel (ch. 0/1/2/3) can be read through the OS bit in its corresponding voltage comparator control register (CMR0/CMR1/CMR2/CMR3).

26.3 Pins of Voltage Comparator

This section describes the pins of the voltage comparator.

■ Pins of Voltage Comparator

Table 26.3-1 Pins of Voltage Comparator (1 / 2)

Pin Name	Pin Function	I/O Type	Pull-up Option	Standby Control	Settings Required for Using The Pin	Default Status

P71/CMP0_P/AN08	GPIO/
	Voltage comparator ch. 0 positive analog input/
	A/D converter ch. 8 analog input
		CMOS input/
		CMOS output/
		Analog input
			Unavailable	Available	CMR0:VCID = 0
					(Enables voltage comparator analog input)
						GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 8 analog input disabled

P72/CMP0_N/AN09	GPIO/
	Voltage comparator ch. 0 positive analog input/
	A/D converter ch. 9 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR0:VCID = 0
					(Enables voltage comparator analog input)
						GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 9 analog input disabled

P70/CMP0_O/OUT0/TRG	GPIO/
	Voltage comparator ch. 0 output/
	Output compare unit ch. 0 output/
	16-bit PPG external trigger input
		CMOS input/
		CMOS output
					CMR0:VCOE = 1
					(Enables voltage comparator output)
						GPIO input enabled;
						GPIO output disabled;
						voltage comparator output disabled;
						output compare ch. 0 output disabled;
						16-bit PPG external trigger input enabled

P74/CMP1_P/AN10	GPIO/
	Voltage comparator ch. 1 positive analog input/
	A/D converter ch. 10 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR1:VCID = 0
					(Enables voltage comparator analog input)
						GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 10 analog input disabled

P75/CMP1_N/AN11	GPIO/
	Voltage comparator ch. 1 negative analog input/
	A/D converter ch. 11 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR1:VCID = 0
					(Enables voltage comparator analog input)
						GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 11 analog input disabled

P73/CMP1_O/OUT1/PPG	GPIO/
	Voltage comparator ch. 1 output/
	Output compare unit ch. 1 output/
	16-bit PPG output
		CMOS input/
		CMOS output
					CMR1:VCOE = 1
					(Enables voltage comparator output)
						GPIO input enabled;
						GPIO output disabled;
						voltage comparator output disabled;
						output compare ch. 0 output disabled;
						16-bit PPG output disabled

P63/CMP2_P/AN12	GPIO/
	Voltage comparator ch. 2 positive analog input/
	A/D converter ch. 12 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR2:VCID = 0
					(Enables voltage comparator analog input)
						GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 12 analog input disabled

Table 26.3-1 Pins of Voltage Comparator (2 / 2)

Pin Name	Pin Function	I/O Type	Pull-up Option	Standby Control	Settings Required for Using The Pin	Default Status

P64/CMP2_N/ AN13	GPIO/
	Voltage comparator ch. 2 negative analog input/
	A/D converter ch. 13 analog input
		CMOS input/
		CMOS output/
		Analog input
			Unavailable	Available	CMR2:VCID = 0 (Enables voltage comparator analog input)	GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 13 analog input disabled

P76/CMP2_O/ UCK	GPIO/
	Voltage comparator ch. 2 output/
	UART/SIO clock I/O
		CMOS input/
		CMOS output
					CMR2:VCOE = 1 (Enables voltage comparator output)	GPIO input enabled;
						GPIO output disabled;
						voltage comparator output disabled;
						UART/SIO clock input enabled;
						UART/SIO clock output disabled

P66/CMP3_P/ AN14	GPIO/
	Voltage comparator ch. 3 positive analog input/
	A/D converter ch. 14 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR3:VCID = 0 (Enables voltage comparator analog input)	GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 14 analog input disabled

P67/CMP3_N/ AN15	GPIO/
	Voltage comparator ch. 3 negative analog input/
	A/D converter ch. 15 analog input
		CMOS input/
		CMOS output/
		Analog input
					CMR3:VCID = 0 (Enables voltage comparator analog input)	GPIO input disabled;
						GPIO output disabled;
						voltage comparator analog input enabled;
						A/D converter ch. 15 analog input disabled

P65/CMP3_O/ UO/SDA	GPIO/
	Voltage comparator ch. 3 output/
	UART/SIO data output/
	I^{2}C SDA I/O
		CMOS input/
		CMOS output
					CMR3:VCOE = 1 (Enables voltage comparator output)	GPIO input enabled;
						GPIO output disabled;
						voltage comparator output disabled;
						UART/SIO data output enabled;
						I^{2}C SDA input enabled;
						I^{2}C SDA output disabled

Note:

- When the voltage comparator analog input function of a pin is enabled, the GPIO function, and the input and output functions of other resources of the same pin will be disabled.

■ Block Diagrams of Pins of Voltage Comparator

Figure 26.3-1 Block Diagram of Pins CMPn_P and CMPn_N (n = 0, 1, 2, 3) of Voltage Comparator

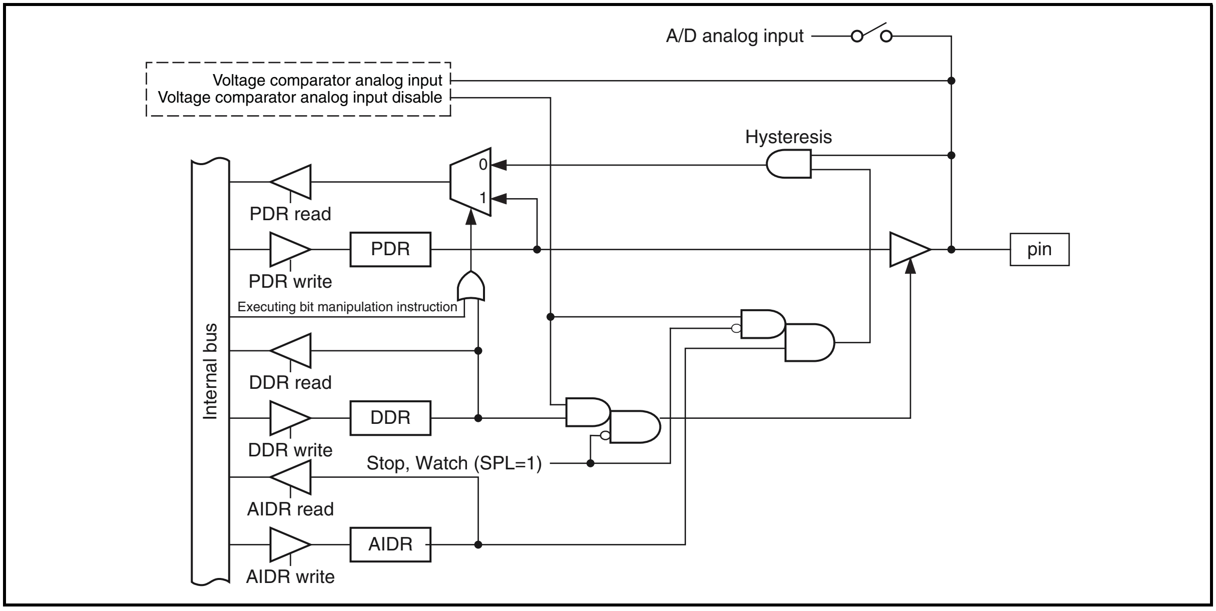


Figure 26.3-2 Block Diagram of Pin CMPn_O (n = 0, 2) of Voltage Comparator

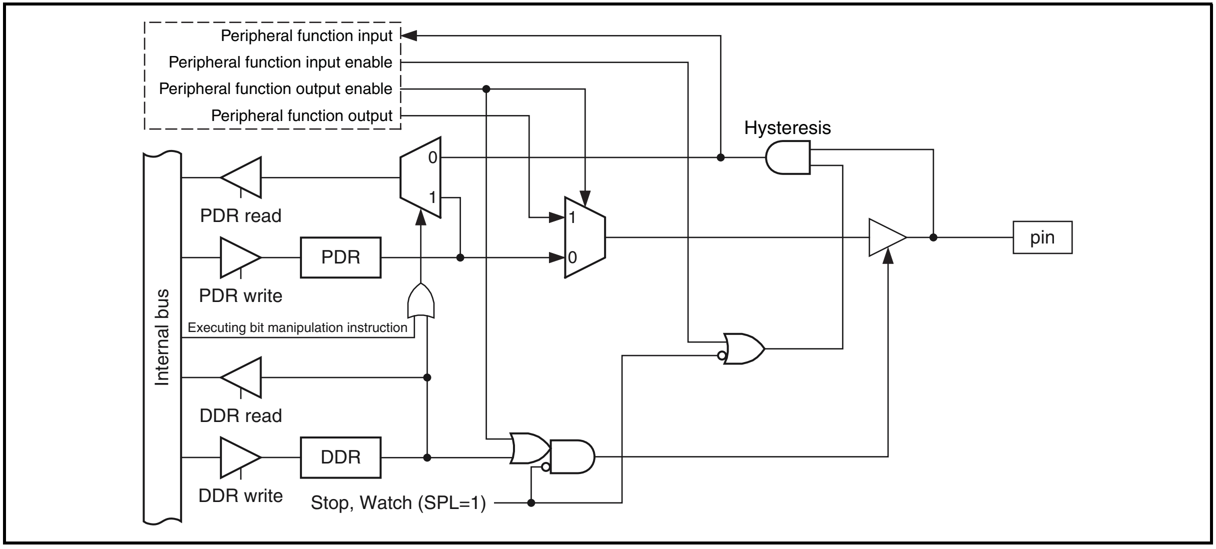

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
0
1
1
0
PDR read
PDR
PDR write
pin
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)

Figure 26.3-3 Block Diagram of Pin CMP3_O of Voltage Comparator

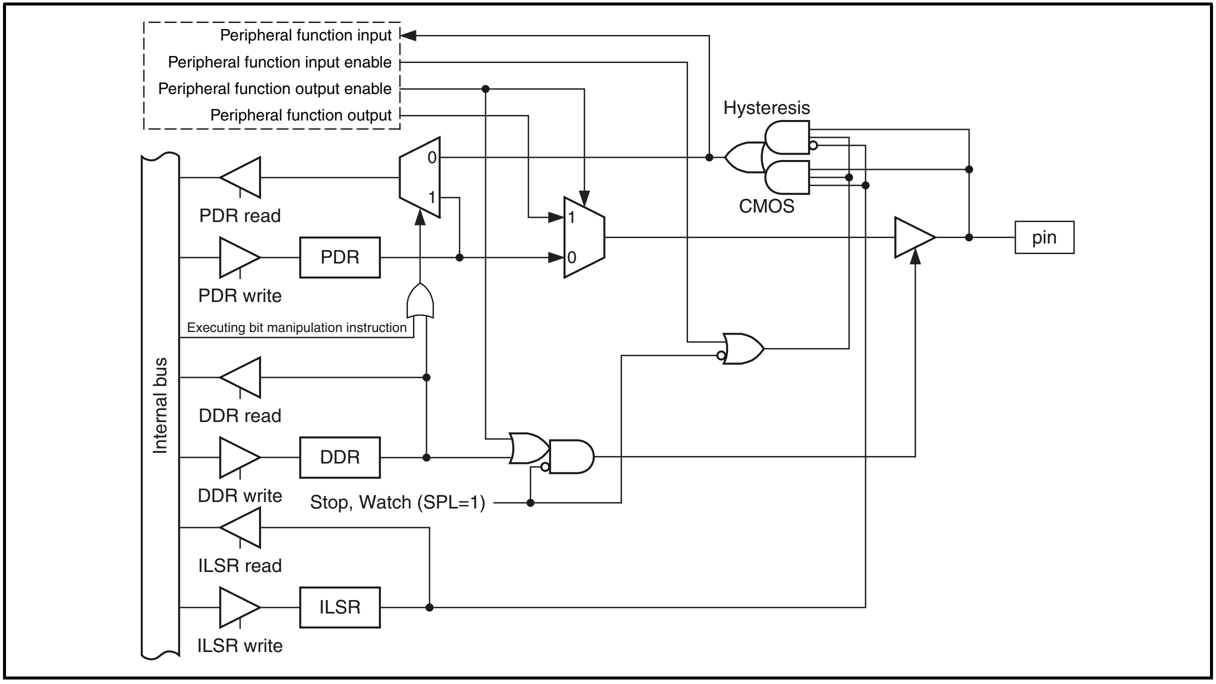
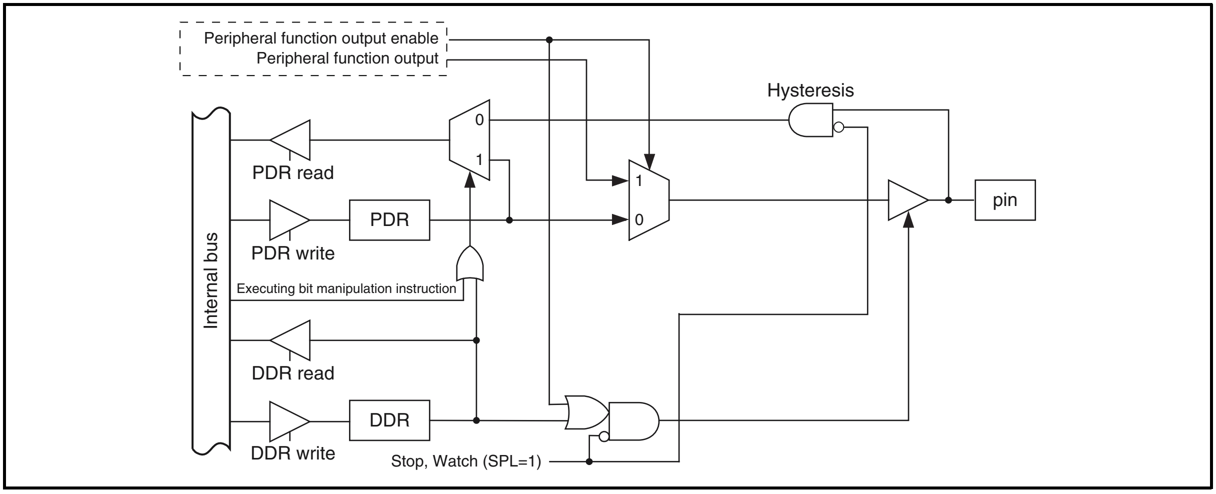

Peripheral function input
Peripheral function input enable
Peripheral function output enable
Peripheral function output
Hysteresis
CMOS
pin
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)
ILSR read
ILSR
ILSR write

Figure 26.3-4 Block Diagram of Pin CMP1_O of Voltage Comparator


Peripheral function output enable
Peripheral function output
Hysteresis
pin
0
1
1
0
PDR read
PDR
PDR write
Executing bit manipulation instruction
Internal bus
DDR read
DDR
DDR write
Stop, Watch (SPL=1)

MB95430H Series

26.4 Registers of Voltage Comparator

This section describes the registers of the voltage comparator.

■ Registers of Voltage Comparator

Figure 26.4-1 Registers of Voltage Comparator

Voltage comparator control register (CMR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CMR0	003A$_H$	-	-	OS	IF	IE	VCID	VCOE	PD	000X0001$_B$
		R0/WX	R0/WX	R/WX	R(RM1),W	R/W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CMR1	003B$_H$	-	-	OS	IF	IE	VCID	VCOE	PD	000X0001$_B$
		R0/WX	R0/WX	R/WX	R(RM1),W	R/W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CMR2	003C$_H$	-	-	OS	IF	IE	VCID	VCOE	PD	000X0001$_B$
		R0/WX	R0/WX	R/WX	R(RM1),W	R/W	R/W	R/W	R/W	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CMR3	003D$_H$	-	-	OS	IF	IE	VCID	VCOE	PD	000X0001$_B$
		R0/WX	R0/WX	R/WX	R(RM1),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.)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
X : Indeterminate
- : Undefined bit

26.4.1 Voltage Comparator Control Register 0/1/2/3 (CMR0/CMR1/CMR2/CMR3)

The voltage comparator control registers are used to turn on and off the voltage comparator (CMR0/CMR1/CMR2/CMR3:PD), to enable and disable voltage comparator output (CMR0/CMR1/CMR2/CMR3:VCOE), and to enable and disable voltage comparator analog inputs (CMR0/CMR1/CMR2/CMR3:VCID).
Except in stop mode, watch mode or time-base timer mode, if CMR0/CMR1/CMR2/CMR3:IE has been set to "1", upon detection of a rising edge or falling edge of voltage comparator output, the voltage comparator generates an interrupt request and the interrupt flag bit (CMR0/CMR1/CMR2/CMR3:IF) is automatically set to "1" at the same time.
The output status of a voltage comparator channel (ch. 0/1/2/3) can be read through the OS bit in its corresponding voltage comparator control register (CMR0/CMR1/CMR2/CMR3).

■ Voltage Comparator Control Register 0/1/2/3 (CMR0/CMR1/CMR2/CMR3)

Figure 26.4-2 Voltage Comparator Control Register 0/1/2/3 (CMR0/CMR1/CMR2/CMR3)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CMR0	003AH	-	-	OS	IF	IE	VCID	VCOE	PD	000X0001B
CMR1	003BH	R0/WX	R0/WX	R/WX	R(RM1),W	R/W	R/W	R/W	R/W	
CMR2	003CH									
CMR3	003DH									

PD	Voltage comparator power down control bit
0	Turns on the voltage comparator.
1	Turns off the voltage comparator.

VCOE	Voltage comparator output enable bit
0	Disables voltage comparator output.
1	Enables voltage comparator output.

VCID	Voltage comparator analog input disable bit
0	Enables voltage comparator analog input.
1	Disables voltage comparator analog input.

IE	Interrupt request enable bit
0	Disables the interrupt request.
1	Enables the interrupt request.

IF	Output edge detection interrupt flag bit	
	Read	Write
0	No output rising/falling edge has occurred.	Clears this bit.
1	An output rising/falling edge has occurred.	No effect on operation.

OS	Output status bit
0	The voltage comparator outputs “L”.
1	The voltage comparator outputs “H”.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “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.)
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.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
X	:	Indeterminate
▨	:	Initial value

Table 26.4-1 Functions of Bits in Voltage Comparator Control Register 0/1/2/3 (CMR0/CMR1/CMR2/CMR3)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5	OS: Output status bit	This bit indicates the output status of the voltage comparator. **Reading "0":** Indicates the voltage comparator outputs "H". **Reading "1":** Indicates the voltage comparator outputs "L". Note: This bit will not be updated in stop mode, watch mode or time-base timer mode. When the PD bit is set to "1" (to turn off the voltage comparator), the OS bit will become "0".
bit4	IF: Output edge detection interrupt flag bit	This bit detects the output rising edge and the output falling edge of the voltage comparator. • With the voltage comparator in operation, this bit will be set to "1" if an output rising edge or an output falling edge occurs. • When read by a read-modify-write (RMW) instruction, this bit always returns "1". **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. Note: This bit will not be updated in stop mode, watch mode or time-base timer mode.
bit3	IE: Interrupt request enable bit	This bit enables or disables the interrupt request of the voltage comparator. **Writing "0":** Disables the interrupt request of the voltage comparator. **Writing "1":** Enables the interrupt request of the voltage comparator. With the interrupt request enabled, the voltage comparator will generate an interrupt request when detecting an output rising edge or an output falling edge.
bit2	VCID: Voltage comparator analog input disable bit	This bit is used to enable or disable voltage comparator analog input. **Writing "0":** Enables voltage comparator analog input. **Writing "1":** Disables voltage comparator analog input.
bit1	VCOE: Voltage comparator output enable bit	This bit is used to enable or disable voltage comparator output. **Writing "0":** Disables voltage comparator output. The output pins of the voltage comparator will be used as general-purpose I/O ports. **Writing "1":** Enables voltage comparator output.
bit0	PD: Voltage comparator power down control bit	This bit is used to turn on or off the voltage comparator. **Writing "0":** Turns on the voltage comparator. **Writing "1":** Turns off the voltage comparator.

26.5 Interrupts of Voltage Comparator

The voltage generator generates an interrupt called output edge detection interrupt. An interrupt request number and an interrupt vector are assigned to the interrupt.

■ Output Edge Detection Interrupt

Table 26.5-1 shows details of the output edge detection interrupt.

Table 26.5-1 Details of Output Edge Detection Interrupt

Item	Details
Interrupt generating condition	An output rising edge or output falling edge occurs.
Interrupt flag	CMR0/CMR1/CMR2/CMR3:IF
Interrupt enable bit	CMR0/CMR1/CMR2/CMR3:IE

Note:

In stop mode, watch mode or time-base timer mode, the edge detection circuit stops operating, and the interrupt flag (IF) bits in the respective voltage comparator control registers (CMR0, CMR1, CMR2 and CMR3) will not be updated even if the voltage comparator has been turned on.

■ Registers and Vector Table Addresses Related to Interrupts of Voltage Comparator

Table 26.5-2 Registers and Vector Table Addresses Related to Interrupts of Voltage Comparator

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Voltage comparator ch. 0	IRQ10	ILR2	L10	$FFE6_H$	$FFE7_H$
Voltage comparator ch. 1	IRQ11	ILR2	L11	$FFE4_H$	$FFE5_H$
Voltage comparator ch. 2	IRQ12	ILR3	L12	$FFE2_H$	$FFE3_H$
Voltage comparator ch. 3	IRQ13	ILR3	L13	$FFE0_H$	$FFE1_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

26.6 Operations of Voltage Comparator

The voltage comparator can be activated by the software according to the settings of the PD bits in the respective CMR0, CMR1, CMR2 and CMR3 registers.

■ Software Activation of Voltage Comparator

The settings shown in Figure 26.6-1 are required for activating the voltage comparator using the software.

Figure 26.6-1 Settings for Activating Voltage Comparator

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CMR0/CMR1/ CMR2/CMR3	-	-	OS	IF	IE	VCID	VCOE	PD
	×	×	❍	❍	0	0	0	0

❍ : Bit to be used
× : Unused bit
0 : Set to "0"

After the voltage comparator is activated as shown above, it has to stabilize before starting to operate.

Note:

Before activating the voltage comparator, set the IE bits in the respective CMR0, CMR1, CMR2 and CMR3 registers to "0" in advance in order to avoid any unexpected interrupt generated due to the voltage comparator being unstable at its startup.

■ Setting Procedure Example

Below is an example of procedure for setting the voltage comparator:

● Initial settings

1) Disable the voltage comparator interrupt request. (CMR0/CMR1/CMR2/CMR3:IE = 0)
2) Activate the voltage comparator according to the settings shown in Figure 26.6-1.
3) Wait until the voltage comparator stabilizes.
4) Clear the interrupt flag bit. (CMR0/CMR1/CMR2/CMR3:IF = 0)
5) Enable the voltage comparator interrupt request. (CMR0/CMR1/CMR2/CMR3:IE = 1), and enable the voltage comparator output (CMR0/CMR1/CMR2/CMR3:VCOE = 1) if necessary.

CHAPTER 27

16-BIT FREE-RUNNING TIMER & 16-BIT OUTPUT COMPARE UNIT

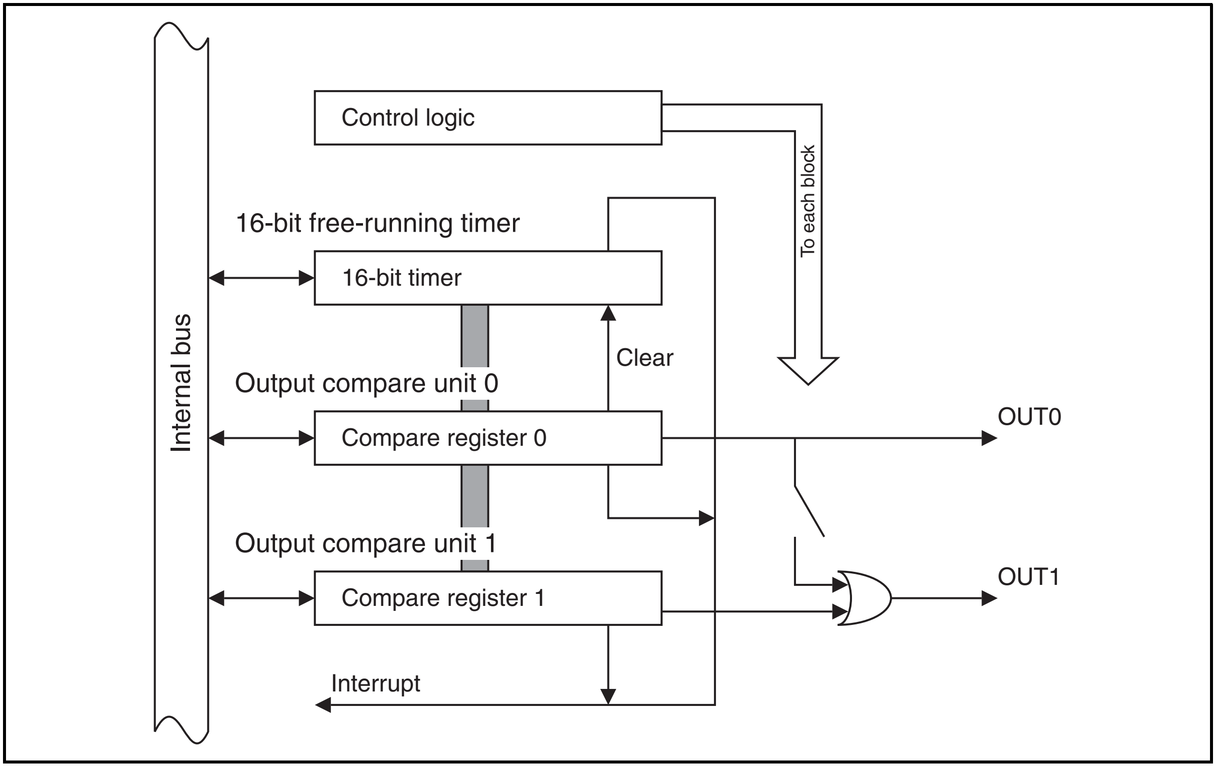
This chapter describes the functions and operations of the 16-bit free-running timer and output compare unit.

27.1 Overview of 16-bit Free-running Timer and 16-bit Output Compare Unit

The 16-bit free-running timer (FRT) and 16-bit output compare unit (OCU) consists of a 16-bit free-running timer and a 16-bit output compare unit. The module is used for generating pulse sequences.

■ Block Diagram of FRT and OCU

Figure 27.1-1 Block Diagram of FRT and OCU



■ 16-bit Free-running Timer (FRT)

The FRT consists of an up/down counter, two control status registers, two extended control status registers, a 16-bit compare clear register, a 16-bit compare clear buffer register and a prescaler. A value output from the FRT is used as a timer base for the 16-bit output compare unit.

It has eight counter clocks.

It can generate an interrupt upon one of the following events:

- An up/down counter overflow occurs.
- The output value of the FRT matches that of the output compare register 0.
- The up/down counter value reaches "0000_H". An interrupt caused by the occurrence of this event is called "zero-detect interrupt".
- The up/down counter value matches the compare clear register value. An interrupt caused by the occurrence of this event is called "compare clear match interrupt".

The up/down counter value is initialized to "0000_H" when a reset or a compare clear match interrupt occurs, or when the counter is cleared by the software. The compare clear match interrupt is generated only when the up/down counter value matches the compare clear register value in up count mode.

It is possible to mask a zero-detect interrupt, or a compare clear match interrupt, or both interrupts so that only one interrupt occurs after the event corresponding to an interrupt has occurred for several times.

In up/down count mode, the up/down counter starts counting down after its value reaches a designated compare clear match value.

■ 16-bit Output Compare Unit (two channels)

The 16-bit output compare unit consists of two 16-bit output compare registers, two 16-bit output compare buffer registers, two compare output latches, an extended output compare control status register, an output compare mode control register and an output compare output control register. Each output compare register (OCCP0/1) is paired with an output compare buffer register (OCCPB0/1). When an FRT value matches the corresponding output compare register value, the output level will be toggled and an interrupt will be generated.

The two 16-bit output compare registers can be used separately for the two channels of the 16-bit output compare unit.

The output pins can be controlled and toggled by the two output compare registers.

The OUT0 pin is controlled according to the setting of any of the output compare registers and the OUT1 pin the setting of both output compare registers.

The initial values of output pins can be set by the output compare control status register (upper).

An interrupt can be generated upon a compare match.

27.2 16-bit Free-running Timer (FRT)

The 16-bit free-running timer (FRT) consists of a 16-bit up/down counter, two compare clear registers, two compare clear buffer registers, two control status registers, two extended control status registers and a prescaler. A counter value output from this timer is used as a timer base for the 16-bit output compare unit. In addition, it has eight different counter clock frequency options.

The FRT can work as an up counter or an up/down counter according to the setting of the count mode bit.

It can generate an interrupt upon one of the following events:

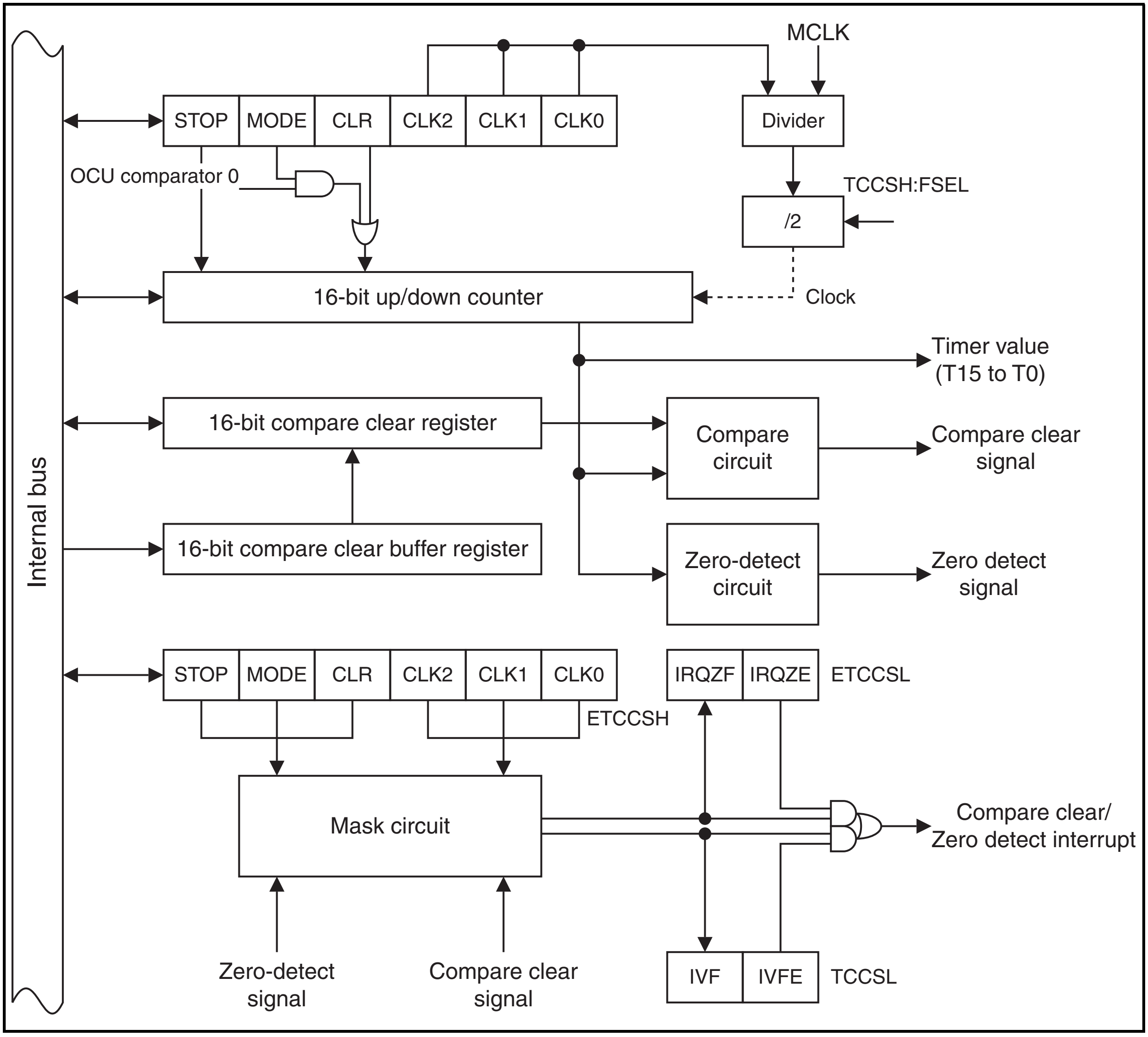
- A counter overflow occurs.
- The output value of the FRT matches that of the output compare register 0.
- The counter value matches the compare clear register value.
- The counter value reaches zero.

The timer has an interrupt mask function to mask specific interrupts.

The counter value can be initialized when a reset occurs, or a compare clear match interrupt occurs, or the counter is cleared by the software. The compare clear match interrupt is generated only in up count mode of the up/down counter.

■ Block Diagram of FRT

Figure 27.2-1 Block Diagram of FRT



27.2.1 Registers of FRT

This section describes the registers of the FRT.

■ Registers of FRT

Figure 27.2-2 Registers of FRT

16-bit free-running timer data register (upper) (TCDTH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005C_H$									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer data register (lower) (TCDTL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005D_H$									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer compare clear buffer register (upper) (CPCLRBH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005E_H$									11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer compare clear buffer register (lower) (CPCLRBL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005F_H$									11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer compare clear register (upper) (CPCLRH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005E_H$									11111111_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit free-running timer compare clear register (lower) (CPCLRL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005F_H$									11111111_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.)
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.)
R0,W	: Write only (Writable. The read value is "0".)
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.
-	: Undefined bit

(Continued)

(Continued)

16-bit free-running timer control status register (upper) (TCCSH)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0060_H	-	FSEL	-	-	-	-	-	-	01000000_B
	R0/WX	R/W	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	

16-bit free-running timer control status register (lower) (TCCSL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0061_H	IVF	IVFE	STOP	MODE	CLR	CLK2	CLK1	CLK0	00000000_B
	R(RM1),W	R/W	R/W	R/W	R0,W	R/W	R/W	R/W	

16-bit free-running timer extended control status register (upper) (ETCCSH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0062_H	-	CIMS2	CIMS1	CIMS0	-	ZIMS2	ZIMS1	ZIMS0	00000000_B
	R0/WX	R/W	R/W	R/W	R0/WX	R/W	R/W	R/W	

16-bit free-running timer extended control status register (lower) (ETCCSL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0063_H	-	-	CNTDIR	-	BFE	IRQZF	IRQZE	CNTMD	00000000_B
	R0/WX	R0/WX	R/WX	R0/WX	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.)
R0,W	: Write only (Writable. The read value is "0".)
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.
-	: Undefined bit

27.2.1.1 16-bit Free-running Timer Data Register Upper/Lower (TCDTH/TCDTL)

The 16-bit free-running timer data register upper (TCDTH) and the 16-bit free-running timer data register lower (TCDTL) form a 16-bit data register from which the value of the 16-bit free-running can be read.
The counter value is cleared to "0000_H" upon a reset.

Writing a value to TCDTH and TCDTL sets the value of the FRT. Before writing a value to TCDTH and TCDTL, ensure that the FRT has stopped operating.

■ 16-bit Free-running Timer Data Register Upper/Lower (TCDTH/TCDTL)

Figure 27.2-3 16-bit Free-running Timer Data Register Upper/Lower (TCDTH/TCDTL)

16-bit free-running timer data register (upper) (TCDTH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005C_H$									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer data register (lower) (TCDTL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005D_H$									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.)

The FRT is initialized upon one of the following events:

- Reset
- The clear timer bit (CLR) in the 16-bit free-running timer control status register is set to "1".
- In up count mode, the counter value matches the value of TCDTH and TCDTL.
- In up count mode, the value of the output compare registers of the 16-bit output compare unit matches the counter value.

Only use one of the following procedures to read and write to TCDTH and TCDTL.

- Use a 16-bit access instruction, e.g. MOVW instruction, to read or write the address of TCDTH.
- Use a byte access instruction, e.g. MOV instruction, to read or write to TCDTH first and then TCDTL.

27.2.1.2 16-bit Free-running Timer Compare Clear Buffer Register Upper/Lower (CPCLRBH/CPCLRBL)

The 16-bit free-running timer compare clear buffer register upper (CPCLRBH) and the 16-bit free-running timer compare clear buffer register lower (CPCLRBL) form a 16-bit buffer register for the 16-bit free-running timer compare clear register upper (CPCLRH) and the 16-bit free-running timer compare clear register lower (CPCLRL) respectively. CPCLRBL and CPCLRH share the same address and so do CPCLRBL and CPCLRL.

■ 16-bit Free-running Timer Compare Clear Buffer Register Upper/Lower (CPCLRBH/CPCLRBL)

Figure 27.2-4 16-bit Free-running Timer Compare Clear Buffer Register Upper/Lower (CPCLRBH/CPCLRBL)

16-bit free-running timer compare clear buffer register (upper) (CPCLRBH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
005E$_H$									11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit free-running timer compare clear buffer register (lower) (CPCLRBL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
005F$_H$									11111111$_B$
	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.)

CPCLRBH is located at the address of CPCLRH and CPCLRBL at the address of CPCLRL.

Immediately after the buffer function is disabled (ETCCSL:BFE = 0) or the FRT is stopped, the values of CPCLRBH and CPCLRBL are transferred to CPCLRH and CPCLRL respectively.

With the buffer function enabled (ETCCSL:BFE = 1), if the FRT detects a counter value, the value will be transferred to CPCLRH and CPCLRL.

Only use one of the following procedures to read and write to CPCLRBH and CPCLRBL.

- Use a 16-bit access instruction, e.g. MOVW instruction, to read or write the address of CPCLRBH.
- Use a byte access instruction, e.g. MOV instruction, to write to CPCLRBH first and then CPCLRBL.

27.2.1.3 16-bit Free-running Timer Compare Clear Register Upper/Lower (CPCLRH/CPCLRL)

The 16-bit free-running timer compare clear register upper (CPCLRH) and the 16-bit free-running timer compare clear register lower (CPCLRL) combine to form a 16-bit register and the value of this 16-bit register is to be compared with the up/down counter value of the FRT.

■ 16-bit Free-running Timer Compare Clear Register Upper/Lower (CPCLRH/ CPCLRL)

Figure 27.2-5 16-bit Free-running Timer Compare Clear Register Upper/Lower (CPCLRH/ CPCLRL)

16-bit free-running timer compare clear register (upper) (CPCLRH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005E_H$									11111111_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit free-running timer compare clear register (lower) (CPCLRL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005F_H$									11111111_B
	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.)

CPCLRH and CPCLRL combine to form a 16-bit register and the value of this 16-bit register is to be compared with the up/down counter value of the FRT. In up count mode, if the up/down counter value matches the value of the combined 16-bit register, the 16-bit free-running timer will be reset to "0" at the next count clock edge.

In up count mode, setting the combined 16-bit register to "$FFFF_B$" (initial value) is equivalent to a counter overflow.

In up/down count mode, when the value of the combined 16-bit register matches the counter value of the FRT, the FRT will switch its operation from up counting to down counting.

Only use one of the following procedures to read CPCLRH and CPCLRL.

- Use a 16-bit access instruction, e.g. MOVW instruction, to read or write the address of CPCLRH.
- Use a byte access instruction, e.g. MOV instruction, to read CPCLRH first and then CPCLRL.

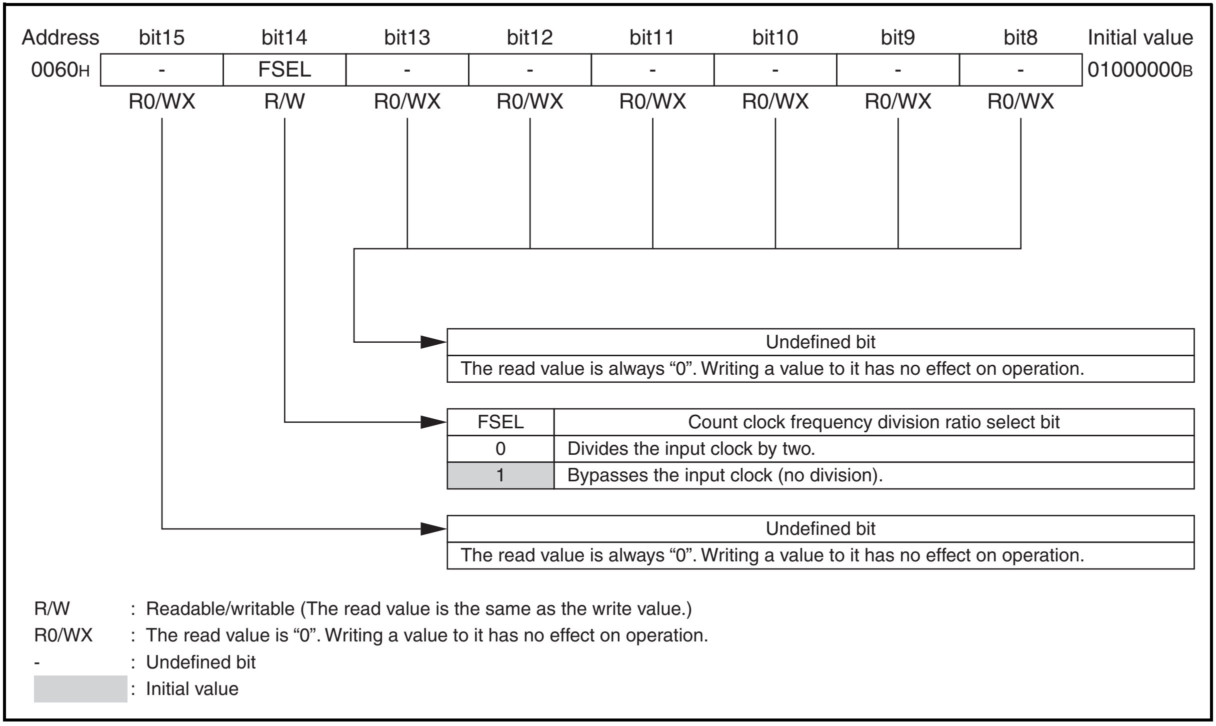
27.2.1.4 16-bit Free-running Timer Control Status Register Upper/Lower (TCCSH/TCCSL)

The 16-bit free-running timer control status register upper (TCCSH) is used to control clock source division. The 16-bit free-running timer control status register lower (TCCSL) is used to set the operating mode of the FRT, start and stop the FRT, and control interrupt generation.

■ 16-bit Free-running Timer Control Status Register Upper (TCCSH)

TCCSH is used to select a clock division ratio and selects a clock source.

Figure 27.2-6 16-bit Free-running Timer Control Status Register Upper (TCCSH)



Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0060$_{H}$	-	FSEL	-	-	-	-	-	-	01000000$_{B}$
	R0/WX	R/W	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

FSEL	Count clock frequency division ratio select bit
0	Divides the input clock by two.
▨ 1	Bypasses the input clock (no division).

Undefined bit
The read value is always “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.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	Initial value

Table 27.2-1 Functions of Bits in 16-bit Free-running Timer Control Status Register Upper (TCCSH)

Bit name		Function
bit15	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bi14	FSEL: Count clock frequency division ratio select bit	This bit is used to select a count clock frequency division ratio. If this bit is set to "0", the number of count clock cycles specified in the count clock frequency select bits (TCCSL:CLK[2:0]) will be divided by two. Before modifying the setting of this bit, ensure that the 16-bit output compare unit has stopped.
bit13 to bit8	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.

■ 16-bit Free-running Timer Control Status Register Lower (TCCSL)

Figure 27.2-7 16-bit Free-running Timer Control Status Register Lower (TCCSL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0061H	IVF	IVFE	STOP	MODE	CLR	CLK2	CLK1	CLK0	00000000B
	R(RM1),W	R/W	R/W	R/W	R0,W	R/W	R/W	R/W	

CLK2	CLK1	CLK0	Count clock frequency select bits
0	0	0	MCLK
0	0	1	MCLK/2
0	1	0	MCLK/4
0	1	1	MCLK/8
1	0	0	MCLK/16
1	0	1	MCLK/32
1	1	0	MCLK/64
1	1	1	MCLK/128

CLR	Clear timer bit	
	Read	Write
0	Always reads “0”.	No effect on operation.
1		Clears the timer to “0000”.

MODE	Timer reset condition bit
0	Initialization by a reset or the CLR bit
1	Initialization by a reset, the CLR bit or the OCCP0 register

STOP	Timer stop bit
0	Enables the up/down counter.
1	Disables the up/down counter (stops the 16-bit free-running timer).

IVFE	Compare clear/overflow interrupt enable bit
0	Disables the generation of the compare clear/overflow interrupt.
1	Enables the generation of the compare clear/overflow interrupt.

IVF	Compare clear interrupt request flag bit	
	Read	Write
0	No interrupt request has been made.	Clears this bit.
1	An interrupt request has been made.	No effect on operation.

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 instruction.)
R0,W : Write only (Writable. The read value is “0”.)
▨ : Initial value

Table 27.2-2 Functions of Bits in 16-bit Free-running Timer Control Status Register Lower (TCCSL)

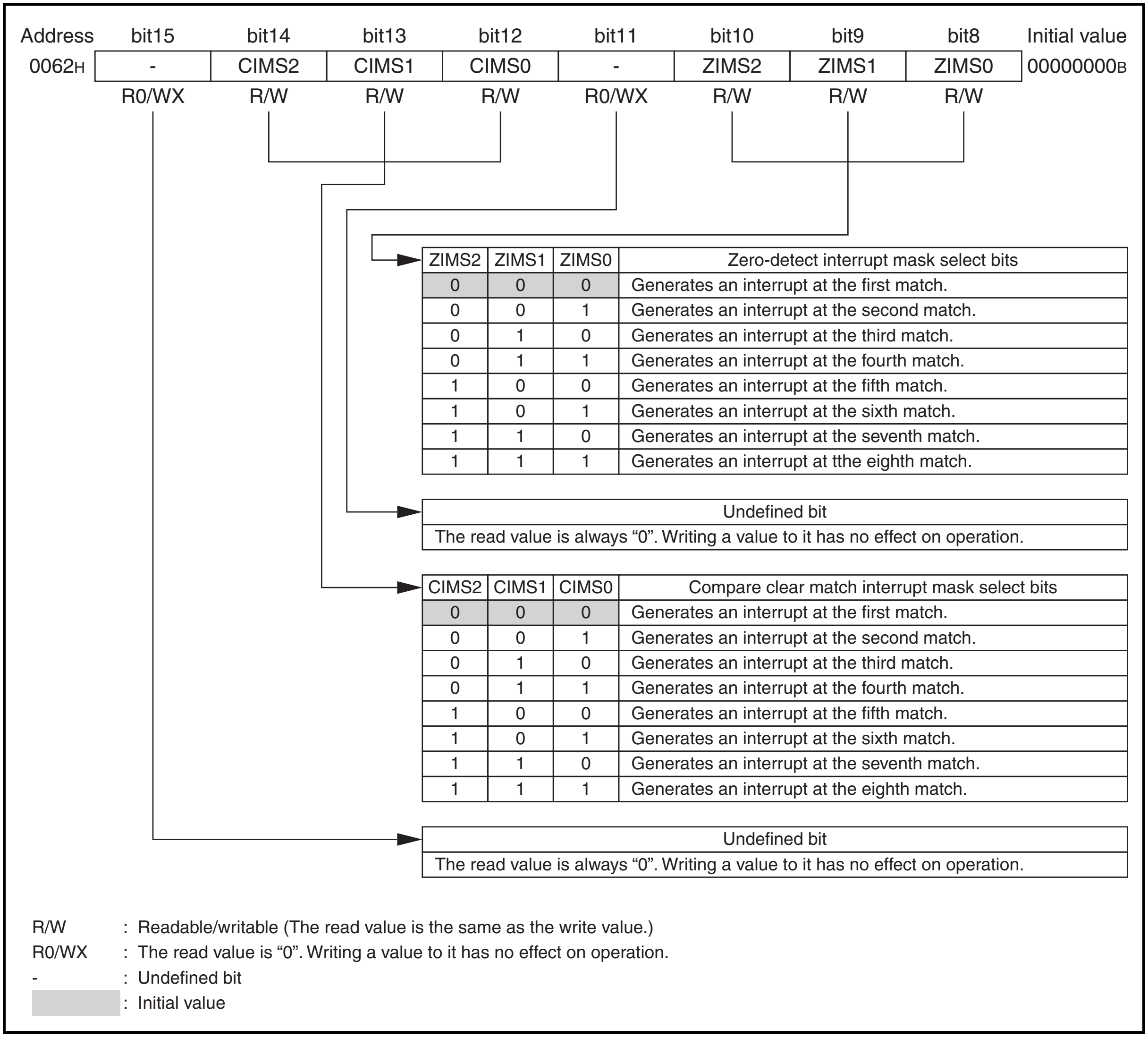
<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>IVF:
Compare clear interrupt request flag bit</td><td>This bit is the compare clear interrupt request flag bit. It can also be used to clear an interrupt request made when a compare clear match interrupt occurs or when the FRT value matches the value of the first compare register of the 16-bit output compare unit connected to the FRT.
<b>Writing "0":</b> Clears this flag bit.
<b>Writing "1":</b> Has no effect on operation.
When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".
With TCCSL:MODE set to "1" and the compare clear interrupt mask counter equal to "000_H", this bit will be set to "1" if the FRT value matches the compare clear register value, or if the timer value matches the value of the first compare register of the 16-bit output compare unit connected to the FRT.
Note: If CPCLRH and CPCLRL are set to "$FFFF_B$", IVF works as a counter overflow flag.</td></tr>
<tr><td>bit6</td><td>IVFE:
Compare clear/ overflow interrupt enable bit</td><td>This bit is used to enable or disable the generation of a compare clear match interrupt request, or the generation of an interrupt request due to the matching of the timer value and the first compare register of the 16-bit output compare unit connected to the FRT.
<b>Writing "0":</b> Disables the generation of the compare clear/overflow interrupt.
<b>Writing "1":</b> Enables the generation of the compare clear/overflow interrupt.</td></tr>
<tr><td>bit5</td><td>STOP:
Timer stop bit</td><td>This bit can be used to stop the FRT.
<b>Writing "0":</b> Enables the up/down counter.
<b>Writing "1":</b> Disables the up/down counter (stops the FRT).</td></tr>
<tr><td>bit4</td><td>MODE:
Timer reset condition bit</td><td>This bit is used to set the conditions for resetting the FRT.
<b>Writing "0":</b> Causes the up/down counter to be initialized by a reset, or by the CLR bit, or by the matching of the up/down counter value and the compare clear registers (CPCLRH and CPCLRL) in up count mode.
<b>Writing "1":</b> Causes the up/down counter to be initialized by a reset, or by the CLR bit, or by the matching of the up/down counter value and the compare clear registers (CPCLRH and CPCLRL) in up count mode, or when the FRT value matches the value of the OCCP0 register of the 16-bit output compare unit connected to the FRT.</td></tr>
<tr><td>bit3</td><td>CLR:
Clear timer bit</td><td>This bit is used to clear the FRT to "0000_H" while it is operating.
<b>Writing "0":</b> Has no effect on operation.
<b>Writing "1":</b> Clears the up/down counter.
Note: To initialize the up/down counter while the FRT stops operating (TCCSL:STOP = 1), write "0000_H" to the 16-bit free-running timer data registers (TCDTH/ TCDTL).</td></tr>
<tr><td>bit2
to
bit0</td><td>CLK2, CLK1, CLK0:
Count clock frequency select bits</td><td>These bits are used to select a count clock for the FRT.
A new count clock selected becomes effective immediately after the value corresponding to that new count clock is written to these bits. Therefore, ensure that the 16-bit output compare unit has stopped operating before writing a value to these bits.
This bit is used to initialize the FRT to "0000_H" while it is operating.
<table>
<tr><th rowspan="2">CLK2</th><th rowspan="2">CLK1</th><th rowspan="2">CLK0</th><th colspan="5">Count clock frequency select bits</th></tr>
<tr><th>Count clock</th><th>MCLK 16 MHz</th><th>MCLK 8 MHz</th><th>MCLK 4 MHz</th><th>MCLK 1 MHz</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>MCLK</td><td>62.5 ns</td><td>125 ns</td><td>0.25 µs</td><td>1 µs</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>MCLK/2</td><td>125 ns</td><td>0.25 µs</td><td>0.5 µs</td><td>2 µs</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>MCLK/4</td><td>0.25 µs</td><td>0.5 µs</td><td>1 µs</td><td>4 µs</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>MCLK/8</td><td>0.5 µs</td><td>1 µs</td><td>2 µs</td><td>8 µs</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>MCLK/16</td><td>1 µs</td><td>2 µs</td><td>4 µs</td><td>16 µs</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>MCLK/32</td><td>2 µs</td><td>4 µs</td><td>8 µs</td><td>32 µs</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>MCLK/64</td><td>4 µs</td><td>8 µs</td><td>16 µs</td><td>64 µs</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>MCLK/128</td><td>8 µs</td><td>16 µs</td><td>32 µs</td><td>128 µs</td></tr>
</table></td></tr>
</table>

27.2.1.5 16-bit Free-running Timer Extended Control Status Register Upper (ETCCSH)

The 16-bit free-running timer extended control status register upper (ETCCSH) is used to control the masking function. The masking function can control the timing of generating a compare clear match interrupt or a zero-detect interrupt by masking events causing a compare clear match interrupt or a zero-detect interrupt to be generated. ETCCSH is used to set how many times such event should occur before a compare clear match interrupt or a zero-detect interrupt is generated.

■ 16-bit Free-running Timer Extended Control Status Register Upper (ETCCSH)

Figure 27.2-8 16-bit Free-running Timer Extended Control Status Register Upper (ETCCSH)



Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0062H	-	CIMS2	CIMS1	CIMS0	-	ZIMS2	ZIMS1	ZIMS0	00000000B
	R0/WX	R/W	R/W	R/W	R0/WX	R/W	R/W	R/W	

ZIMS2	ZIMS1	ZIMS0	Zero-detect interrupt mask select bits
0	0	0	Generates an interrupt at the first match.
0	0	1	Generates an interrupt at the second match.
0	1	0	Generates an interrupt at the third match.
0	1	1	Generates an interrupt at the fourth match.
1	0	0	Generates an interrupt at the fifth match.
1	0	1	Generates an interrupt at the sixth match.
1	1	0	Generates an interrupt at the seventh match.
1	1	1	Generates an interrupt at tthe eighth match.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

CIMS2	CIMS1	CIMS0	Compare clear match interrupt mask select bits
0	0	0	Generates an interrupt at the first match.
0	0	1	Generates an interrupt at the second match.
0	1	0	Generates an interrupt at the third match.
0	1	1	Generates an interrupt at the fourth match.
1	0	0	Generates an interrupt at the fifth match.
1	0	1	Generates an interrupt at the sixth match.
1	1	0	Generates an interrupt at the seventh match.
1	1	1	Generates an interrupt at the eighth match.

Undefined bit
The read value is always “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.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 27.2-3 Functions of Bits in 16-bit Free-running Timer Extended Control Status Register Upper (ETCCSH)

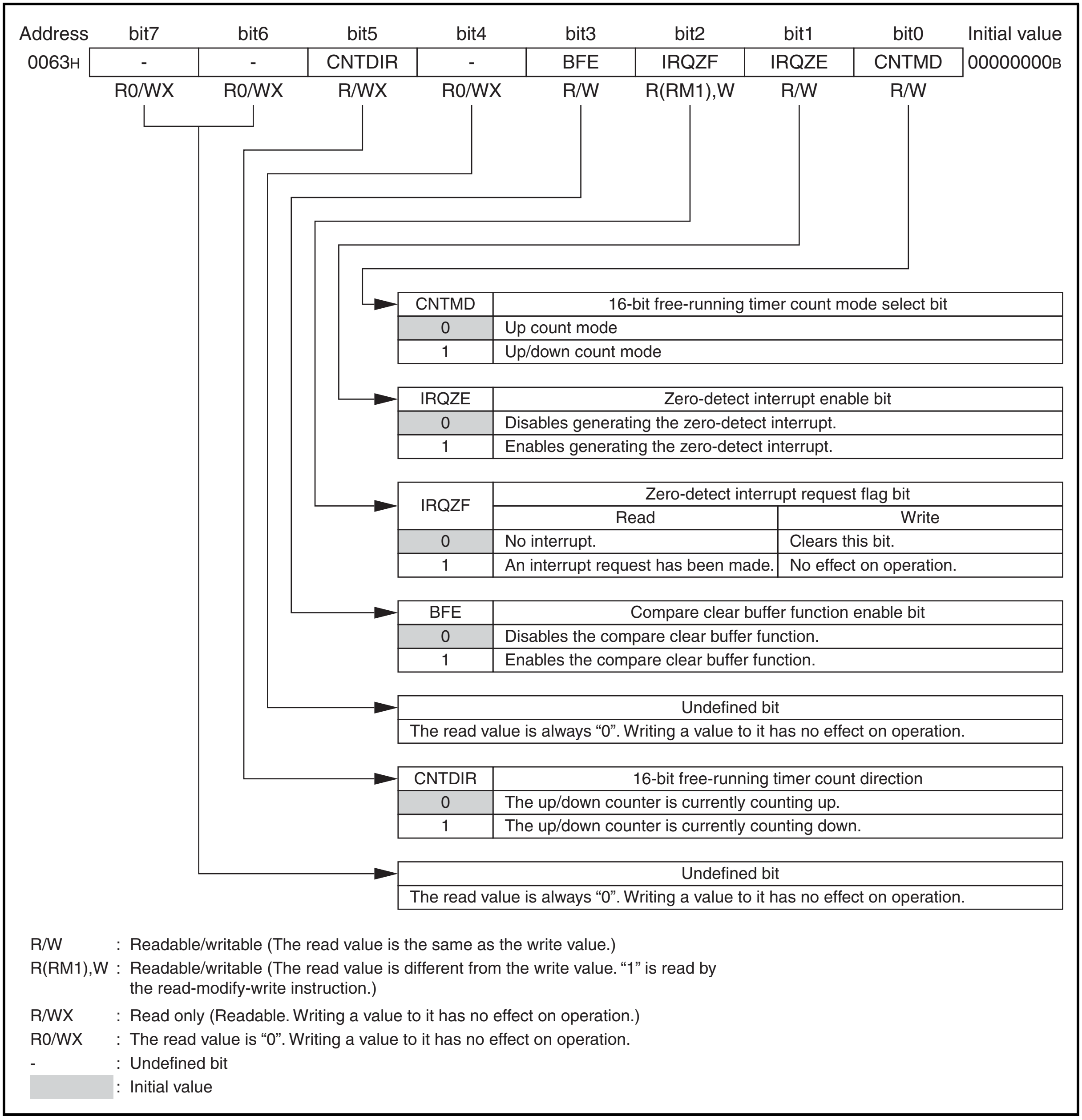
Bit name		Function
bit15	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit14 to bit12	CIMS2, CIMS1, CIMS0: Compare clear match interrupt mask select bits	These bits are used to set the period of masking the compare clear match interrupt. If these bits are set to "000_B", the interrupt source of the compare clear match interrupt will not be masked. These bits return the mask counter value when read. Note: When read by a read-modify-write (RMW) type of instruction, these bits return the masking period specified. In the case of writing a new value to these bits while the FRT is operating (TCCSL:STOP = 0), the new value will be loaded to the mask counter after the mask counter reaches "000_H". In the case of writing a new value to these bits while the FRT stops operating (TCCSL:STOP = 1), the new value will be loaded to the mask counter immediately after being written to these bits.
bit11	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit10 to bit8	ZIMS2, ZIMS1, ZIMS0: Zero-detect interrupt mask select bits	These bits are used to set the period of masking the zero-detect interrupt. If these bits are set to "000_B", the interrupt source of the zero-detect interrupt will not be masked. These bits return the mask counter value when read. Note: When read by a read-modify-write (RMW) type of instruction, these bits return the masking period specified. In the case of writing a new value to these bits while the FRT is operating (TCCSL:STOP = 0), the new value will be loaded to the mask counter after the mask counter reaches "000_H". In the case of writing a new value to these bits while the FRT stops operating (TCCSL:STOP = 1), the new value will be loaded to the mask counter immediately after being written to these bits.

27.2.1.6 16-bit Free-running Timer Extended Control Status Register Lower (ETCCSL)

The 16-bit free-running timer extended control status register (ETCCSL) is used to set the buffer operating mode of the compare clear register upper (CPCLRH) and compare clear register lower (CPCLRL), the count mode of the 16-bit free-running timer, and the generation of zero-detect interrupts.

■ 16-bit Free-running Timer Extended Control Status Register Lower (ETCCSL)

Figure 27.2-9 16-bit Free-running Timer Extended Control Status Register Lower (ETCCSL)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0063H	-	-	CNTDIR	-	BFE	IRQZF	IRQZE	CNTMD	00000000B
	R0/WX	R0/WX	R/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	

CNTMD	16-bit free-running timer count mode select bit
0	Up count mode
1	Up/down count mode

IRQZE	Zero-detect interrupt enable bit
0	Disables generating the zero-detect interrupt.
1	Enables generating the zero-detect interrupt.

IRQZF	Zero-detect interrupt request flag bit	
	Read	Write
0	No interrupt.	Clears this bit.
1	An interrupt request has been made.	No effect on operation.

BFE	Compare clear buffer function enable bit
0	Disables the compare clear buffer function.
1	Enables the compare clear buffer function.

Undefined bit
The read value is always "0". Writing a value to it has no effect on operation.

CNTDIR	16-bit free-running timer count direction
0	The up/down counter is currently counting up.
1	The up/down counter is currently counting down.

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

Table 27.2-4 Functions of Bits in 16-bit Free-running Timer Extended Control Status Register Lower (ETCCSL)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit5	CNTDIR: 16-bit free-running timer count direction bit	This bit is a read-only bit and indicates the count direction of the FRT. **Reading "0":** Indicates that the up/down counter is currently counting up. **Reading "1":** Indicates that the up/down counter is currently counting down. Writing a value to it has no effect on operation.
bit4	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit3	BFE: Compare clear buffer function enable bit	This bit is used to enable or disable the compare clear buffer function. **Writing "0":** Disables the compare clear buffer function. **Writing "1":** Enables the compare clear buffer function.
bit2	IRQZF: Zero-detect interrupt flag bit	This bit is a flag bit indicating that a zero-detect interrupt request has been made and is also used to clear the interrupt request. **Writing "0":** Clears this flag bit. **Writing "1":** Has no effect on operation. When read by the read-modify-write (RMW) type of instruction, this bit always returns "1". This bit is set to "1" when the FRT value reaches "0000_H" and the zero-detect interrupt mask counter reaches "000_H". Note: If the FRT reaches "0000_H" as a result of timer clearing, i.e. writing "1" to TCCSL:CLR to clear the timer, this bit will not be set to "1".
bit1	IRQZE: Zero-detect interrupt enable bit	This bit is used to enable or disable the generation of a zero-detect interrupt request. **Writing "0":** Disables generating the zero-detect interrupt. **Writing "1":** Enables generating the zero-detect interrupt.
bit0	CNTMD: 16-bit free-running timer count mode select bit	This bit is used to select a count mode of the FRT. **Writing "0":** Selects the up count mode. **Writing "1":** Selects the up/down count mode. This bit can be modified regardless of whether the FRT is operating or has stopped. If this bit is modified while the FRT is operating, the new value written to this bit will be stored in a buffer. The new count mode selected according to the new bit value will start when the FRT value reaches "0000_H".

27.2.2 Operations of FRT

The FRT starts counting from "0000$_H$" (counter value) after a reset is released. The counter value is used as the time base for the 16-bit output compare unit.

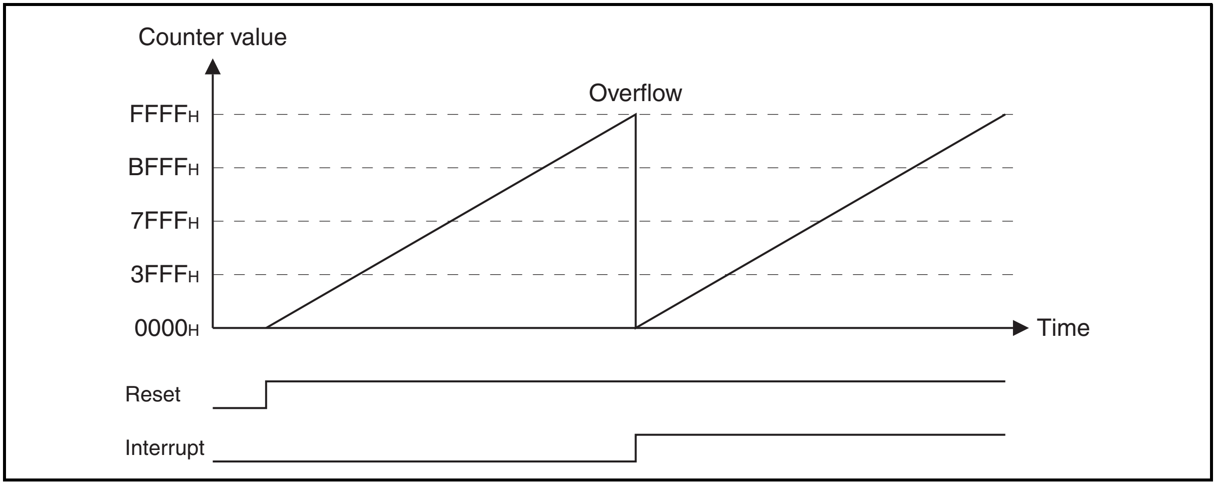
■ Operations of FRT

The counter is cleared upon one of the following events:

- Reset
- A counter overflow occurs.
- In up count mode, the value of the 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL) matches the counter value.
- The value of the output compare registers of the 16-bit output compare unit matches the counter value.
- The clear timer bit (CLR) in the 16-bit free-running timer control status register is set to "1".
- While the FRT stops, "0000$_H$" is written to TCDTH and TCDTL.

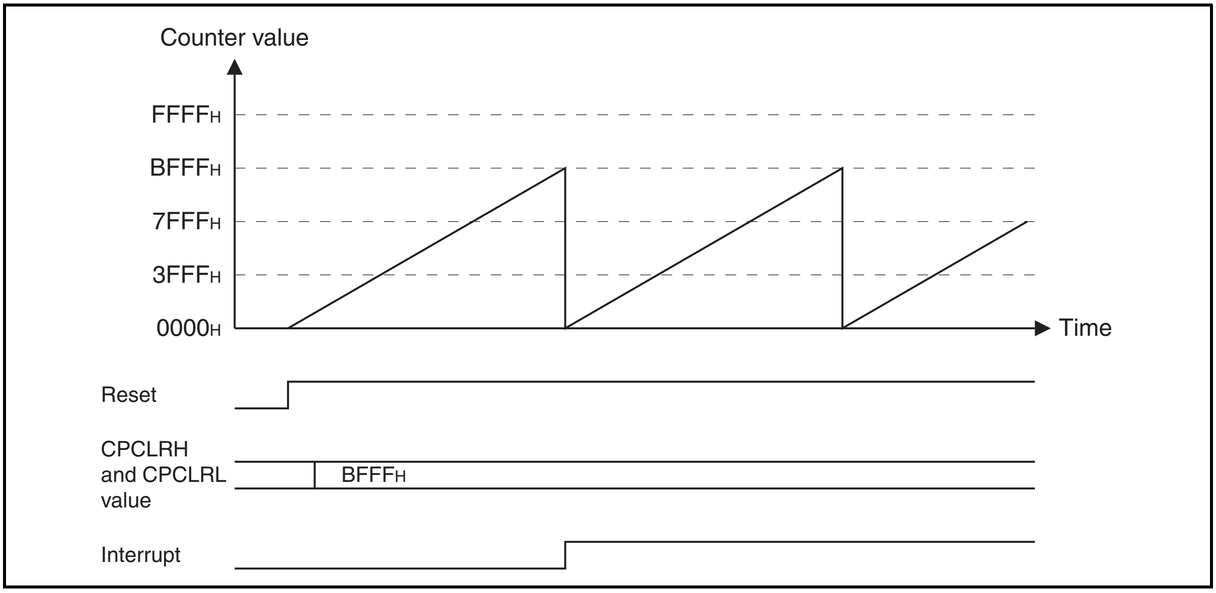
■ Counter Cleared due to Overflow

Figure 27.2-10 Counter Cleared due to Overflow in Up Count Mode



■ Counter Cleared as Value of CPCLRH and CPCLRL Matches Counter Value in Up Count Mode

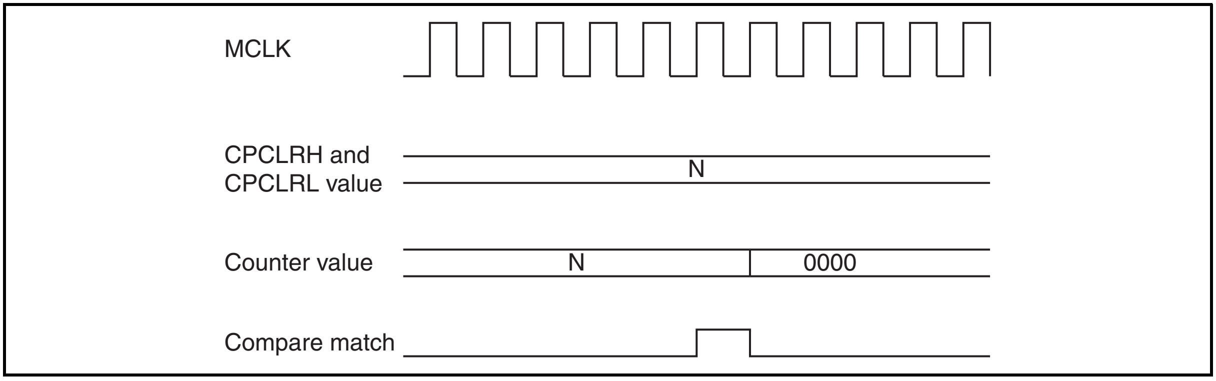
Figure 27.2-11 Counter Cleared as Value of CPCLRH and CPCLRL Matches Counter Value in Up Count Mode



■ Timing of Clearing FRT

The counter can be cleared when a reset occurs, or a compare clear match occurs, or the value of the output compare registers of the 16-bit output compare unit matches the counter value, or the counter is cleared by the software. Upon a reset, the counter is cleared immediately. In the case of a compare clear match, the value of the output compare registers equal to the counter value or clearing the counter by the software (TCCS:CLR = 1), the counter will be cleared in sync with the count time.

Figure 27.2-12 Timing of Clearing FRT



■ Timer Mode

The FRT can operate in one of the following modes:

1. Up count mode (ETCCSL:CNTMD = 0)
2. Up/down count mode (ETCCSL:CNTMD = 1)

In up count mode, the counter of the FRT starts counting from a preset value in the 16-bit free-running timer data registers (TCDTH and TCDTL) and continues counting up until the counter value matches the value in the 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL) or a counter overflow occurs. After the counter value matches the value in the 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL) or a counter overflow occurs, the counter is cleared to "0000_H" and counts up again.

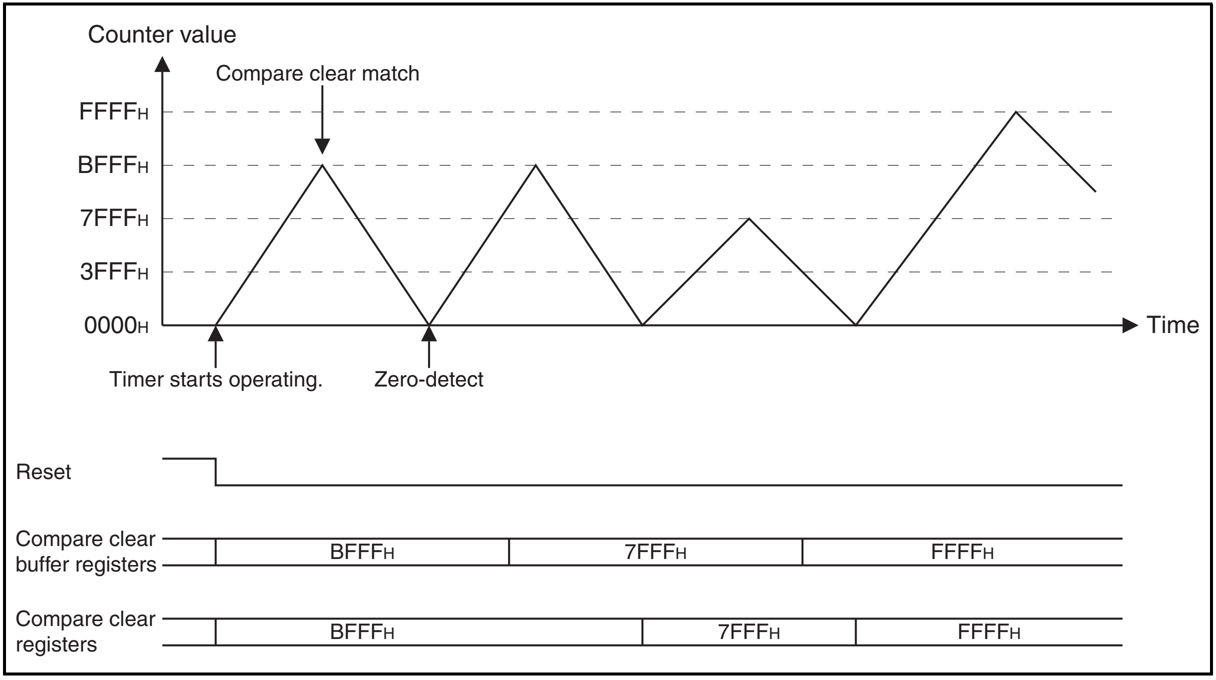
In up/down count mode, the counter starts counting from a preset value in the 16-bit free-running timer data registers (TCDTH and TCDTL) and continues counting up until the counter value matches the value in the 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL). After the counter value matches the value in the 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL), the counter switches from counting up to counting down and then counts down until it reaches "0000_H". After reaching "0000_H", the counter counts up again.

The 16-bit free-running timer count mode select bit (ETCCSL:CNTMD) can be modified regardless of whether the timer is operating or has stopped. If the bit is modified while the 16-bit free-running timer is operating, the new value written to the bit will be stored in a buffer. The new count mode selected according to the new bit value will start when the 16-bit free-running timer value reaches "0000_H".

■ Compare Clear Buffer

The 16-bit free-running timer compare clear registers (CPCLRH and CPCLRL) have a buffer function. With the buffer function enabled (ETCCSL:BFE = 1), data written to the 16-bit free-running timer compare clear buffer registers (CPCLRBH and CPCLRBL) will be transferred to CPCLRH and CPCLRL when the counter reaches "0000_H". When the buffer function is disabled, data will be written directly to CPCLRH and CPCLRL.

Figure 27.2-13 Operation in Up/down Count Mode with Buffer Function Enabled



■ Timer Interrupt

The FRT can generate two types of interrupt:

1. Compare clear/overflow interrupt
2. Zero-detect interrupt

A zero-detect interrupt will be generated when the counter value reaches "0000_H".

A compare clear interrupt will be generated when the counter value matches the value of CPCLRH and CPCLRL.

An overflow interrupt will be generated when the counter value changes from "$FFFF_H$" to 0000_H" in up count mode.

The compare clear interrupt and the overflow interrupt share the same interrupt enable bit and the same interrupt flag.

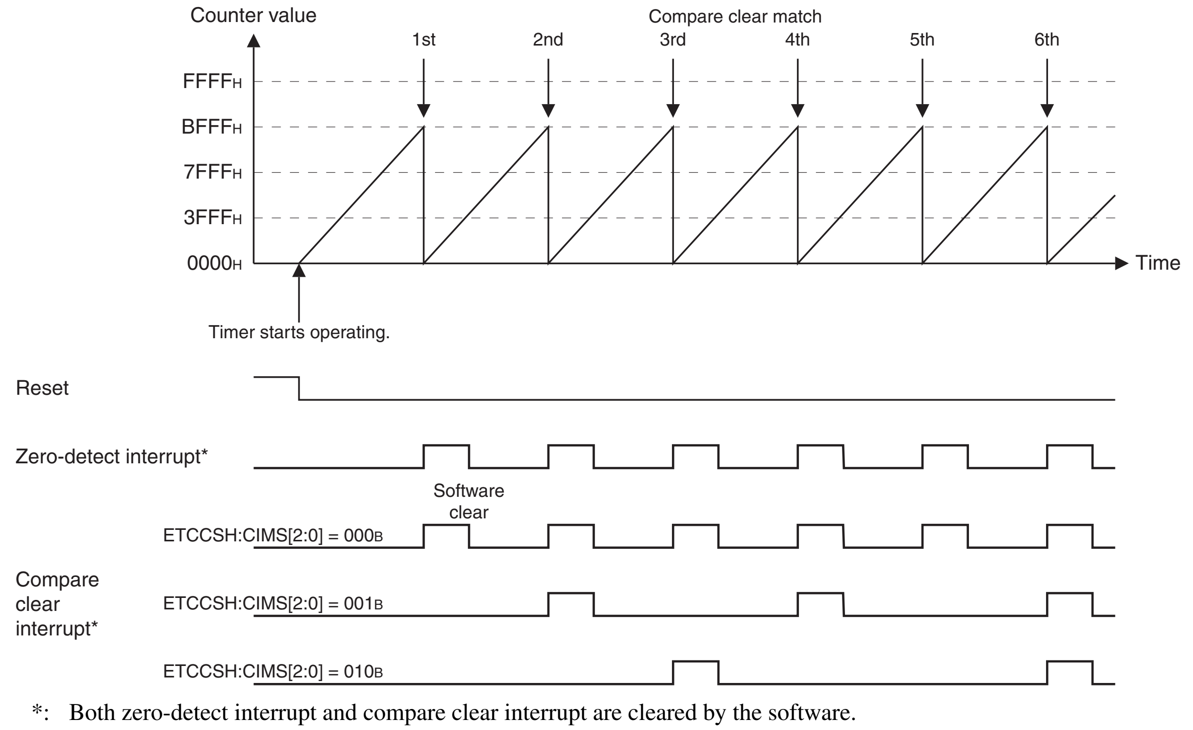
■ Interrupt Mask Function

Both zero-detect interrupt and compare clear interrupt can be masked.

The period of masking the zero-detect interrupt can be selected by ETCCSH:ZIMS2 to ZIMS0. When ZIMS[2:0] are set to "000_B" and generating the zero-detect interrupt is enabled, no zero-detect interrupt source will be masked.

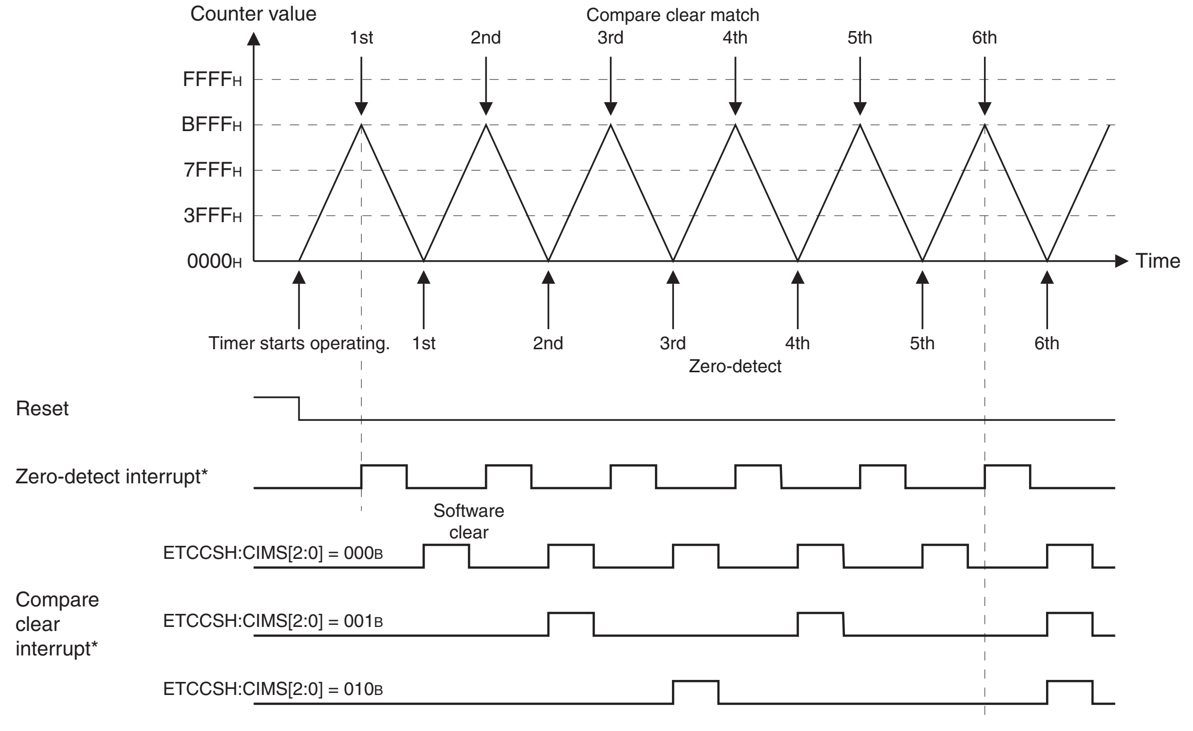
The period of masking the compare clear interrupt can be selected by ETCCSH:CIMS2 to CIMS0. When CIMS[2:0] are set to "000_B" and generating the compare clear interrupt is enabled, no compare clear interrupt source will be masked.

Figure 27.2-14 Compare Clear Interrupt Mask Function in Up Count Mode



*: Both zero-detect interrupt and compare clear interrupt are cleared by the software.

Figure 27.2-15 Zero-detect Interrupt Mask Function in Up/down Count Mode



*: Both zero-detect interrupt and compare clear interrupt are cleared by the software.

■ Register and Vector Table Addresses Related to Interrupts of FRT

Table 27.2-5 Registers and Vector Table Addresses Related to Interrupts of FRT

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
16-bit free-running timer (compare match/ zero-detect/overflow)	IRQ14	ILR3	L14	$FFDE_H$	$FFDF_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

27.3 16-bit Output Compare Unit (OCU)

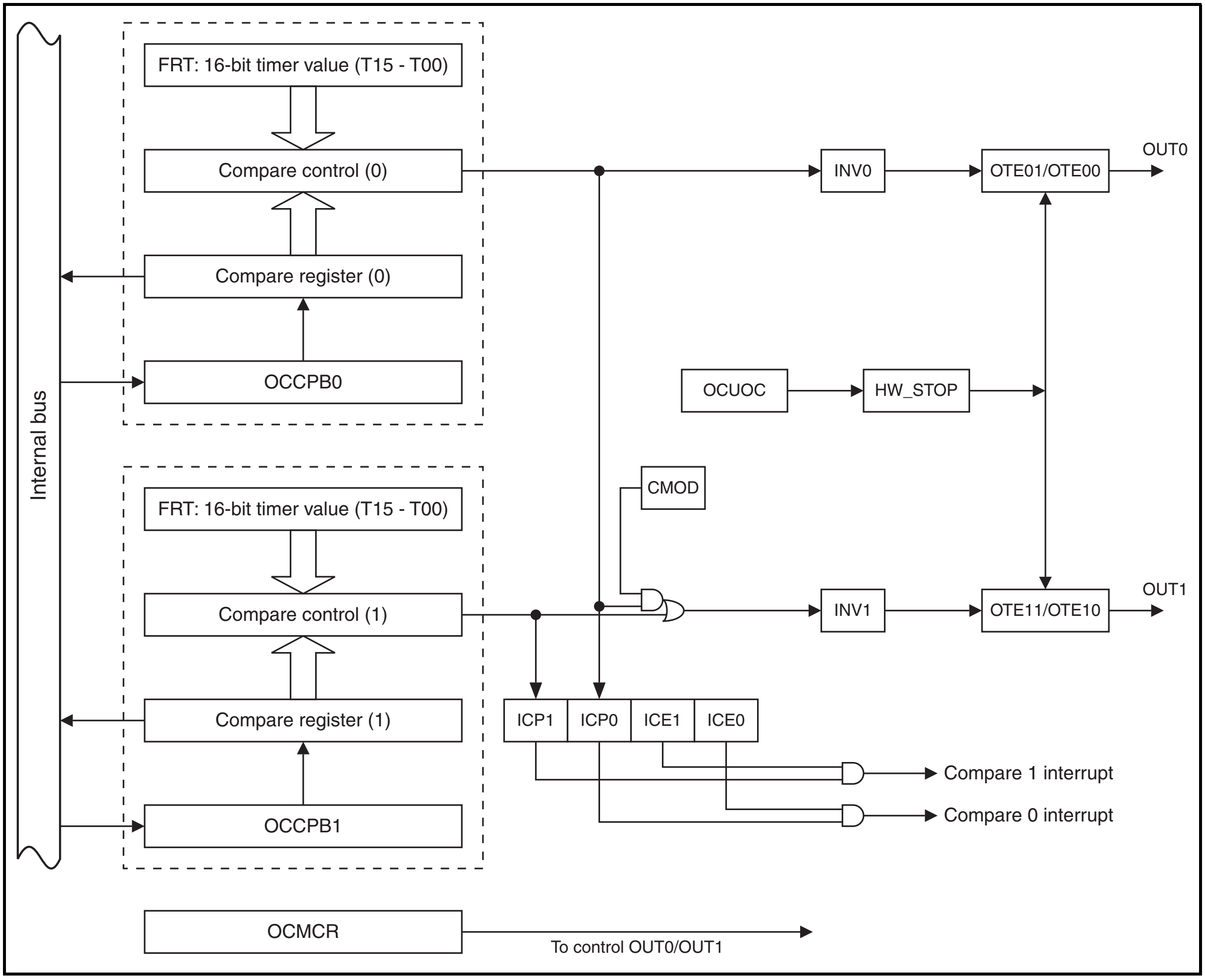
The 16-bit output compare unit (OCU) consists of two 16-bit output compare registers, two 16-bit output compare buffer registers, two compare output latches, an extended output compare control status register, an output compare mode control register and an output compare output control register. When the FRT counter value matches an output compare register value, depending on settings of registers related to output level inversion and interrupt generation, the output level can be toggled and an interrupt can be generated.

■ Features

- The OCU has two channels. Each channel has one output compare register. Both can be used to control pin output depending on mode settings.
- The initial value of each pin output can be specified separately.
- An interrupt can be generated upon a compare match in a comparison.
- Each channel can generate PWM signal using the output compare register.
- The output compare register can be used to generate cyclic waveform signal.
- When the OCU compare register value is "0000_H" or "$FFFF_H$" or is greater than the FRT compare clear register value, the output of the OCU will no be toggled.
- The compare buffer register on an OCU channel is a dedicated buffer register for the output compare register on the same OCU channel.
- The output compare mode control register (OCMCR) is used to control the output level and the inversion of the output level.

■ Block Diagram of OCU

Figure 27.3-1 Block Diagram of OCU



Depending on SYSC2 settings, output compare result OUT0/OUT1 can be output to P70/P73 or PG1/PG2. For details see "CHAPTER 31 SYSTEM CONFIGURATION CONTROLLER".

27.3.1 Registers of OCU

This section describes the registers of the OCU.

■ Registers of OCU

Figure 27.3-2 Registers of OCU

16-bit output compare stop trigger control register (OCUOC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0046_H	—	—	—	—	—	OCSTPSEL			00000000_B
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	

16-bit output compare channel 0 register (upper) (OCCP0H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare buffer channel 0 register (upper) (OCCPB0H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064_H									00000000_B
	W	W	W	W	W	W	W	W	

16-bit output compare channel 0 register (lower) (OCCP0L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0065_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare buffer channel 0 register (lower) (OCCPB0L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0065_H									00000000_B
	W	W	W	W	W	W	W	W	

16-bit output compare channel 1 register (upper) (OCCP1H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0066_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare buffer channel 1 register (upper) (OCCPB1H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0066_H									00000000_B
	W	W	W	W	W	W	W	W	

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

(Continued)

(Continued)

16-bit output compare channel 1 register (lower) (OCCP1L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0067$_H$									00000000$_B$
	R	R	R	R	R	R	R	R	

16-bit output compare buffer channel 1 register (lower) (OCCPB1L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0067$_H$									00000000$_B$
	W	W	W	W	W	W	W	W	

16-bit output compare control status register (upper) (OCSH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0068$_H$	—	CMOD0	OTE11	OTE10	OTE01	OTE00	OTD1	OTD0	00000000$_B$
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit output compare control status register (lower) (OCSL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0069$_H$	ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0	00000000$_B$
	R(RM1),W	R(RM1),W	R/W	R/W	R0/WX	R0/WX	R/W	R/W	

16-bit output compare mode control register (OCMCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006A$_H$	—	FDEN1	INV1	CMPMD1	—	FDEN0	INV0	CMPMD0	00000000$_B$
	R0/WX	R/W	R/W	R/W	R0/WX	R/W	R/W	R/W	

16-bit output compare extended control status register (EOCS)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006B$_H$	HW_STOP	—	BTS1	BUF1	—	—	BTS0	BUF0	00000000$_B$
	R(RM1),W	R0/WX	R/W	R/W	R0/WX	R0/WX	R/W	R/W	

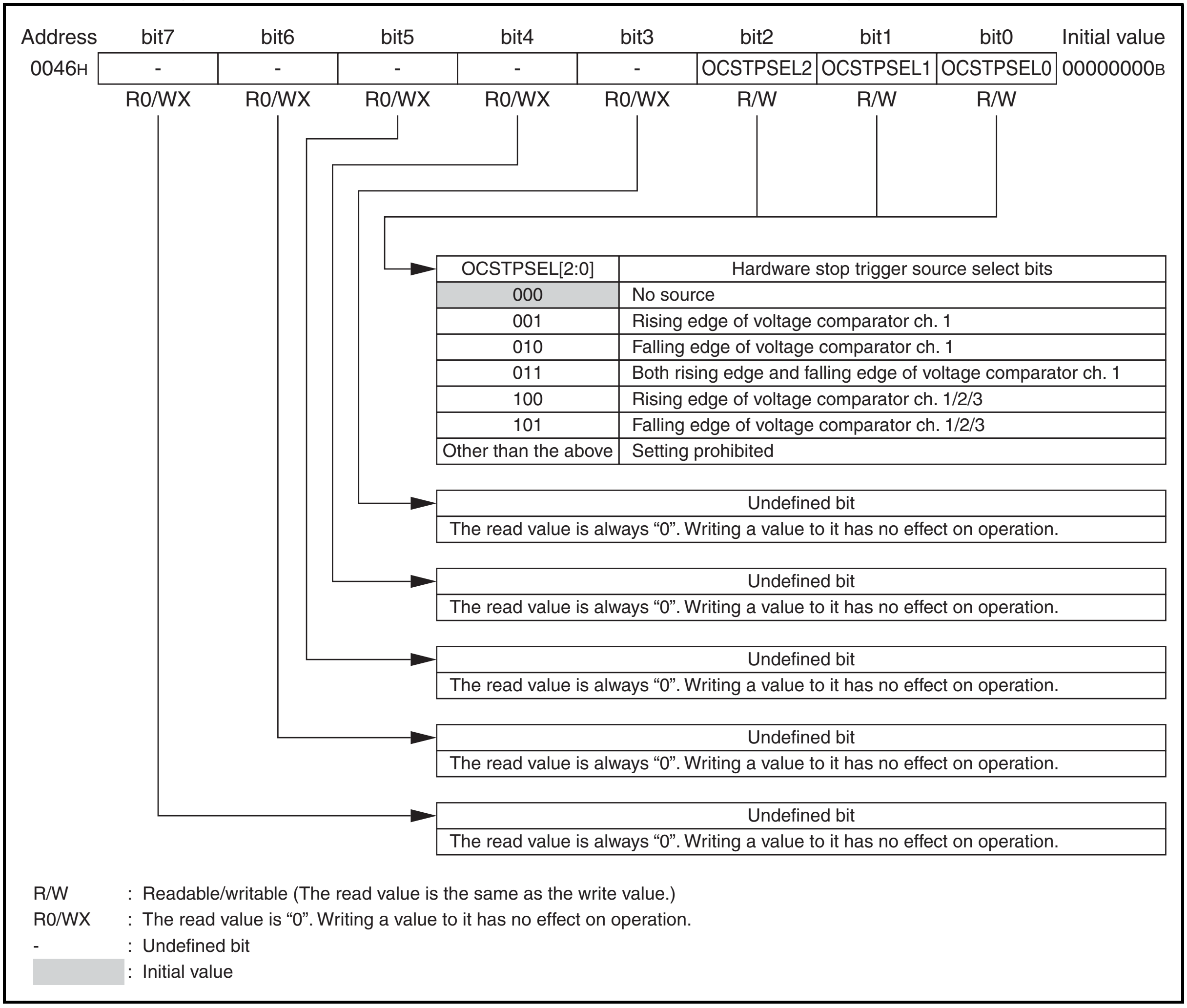
R : Read only
R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R0/WX : The read value is "0". Writing a value to it has no effect on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

27.3.1.1 16-bit Output Compare Output Control Register (OCUOC)

This section describes details of the 16-bit output compare output control register.

■ 16-bit Output Compare Output Control Register (OCUOC)

Figure 27.3-3 16-bit Output Compare Output Control Register (OCUOC)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0046H	-	-	-	-	-	OCSTPSEL2	OCSTPSEL1	OCSTPSEL0	00000000B
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	

OCSTPSEL[2:0]	Hardware stop trigger source select bits
000	No source
001	Rising edge of voltage comparator ch. 1
010	Falling edge of voltage comparator ch. 1
011	Both rising edge and falling edge of voltage comparator ch. 1
100	Rising edge of voltage comparator ch. 1/2/3
101	Falling edge of voltage comparator ch. 1/2/3
Other than the above	Setting prohibited

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “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.)
R0/WX	: The read value is “0”. Writing a value to it has no effect on operation.
-	: Undefined bit
▨	: Initial value

Table 27.3-1 Functions of Bits in 16-bit Output Compare Output Control Register (OCUOC)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7
to
bit3</td><td>Undefined bits</td><td>The read value is always "0". Writing a value to it has no effect on operation.
The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.</td></tr>
<tr><td>bit2
to
bit0</td><td>OCSTPSEL[2:0]: Hardware stop trigger source select bits</td><td>These bits are used to select a hardware stop trigger source.
<table>
<tr><th>OCSTPSEL[2:0]</th><th>Details</th></tr>
<tr><td>000</td><td>No source</td></tr>
<tr><td>001</td><td>Rising edge of voltage comparator ch. 1</td></tr>
<tr><td>010</td><td>Falling edge of voltage comparator ch. 1</td></tr>
<tr><td>011</td><td>Rising edge and falling edge of voltage comparator ch. 1</td></tr>
<tr><td>100</td><td>Rising edge of voltage comparator ch. 1/2/3</td></tr>
<tr><td>101</td><td>Falling edge of voltage comparator ch. 1/2/3</td></tr>
<tr><td>Other than the above</td><td>Setting prohibited</td></tr>
</table>
*: The specific edge of the voltage comparator channel that occurs first will be used as the start trigger.</td></tr>
</table>

■ Notes on Using Rising/Falling Edge of Voltage Comparator as Hardware Stop Trigger

- Before turning off the voltage comparator, disable OCU output and select no source as a hardware stop trigger by writing "000" to OCUOR:OCSTPSEL[2:0], then clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF).
- After turning on the voltage comparator, clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF), then enable OCU output and select a rising/falling edge of the voltage comparator as a hardware stop trigger.

27.3.1.2 16-bit Output Compare Channel 0/1 Register (OCCP0/OCCP1)

The value of the 16-bit output compare channel 0/1 register is compared with the FRT value. OCCP0 and OCCP1 must be accessed by word access instructions. When the value of OCCP0/OCCP1 matches that of the FRT, a compare signal will be generated and the output compare interrupt flag will be set. In addition, with the output of OCU enabled, upon a match between the value of OCCP0/OCCP1 and that of the FRT, the output pin (OUT0/OUT1) will output a signal of the level selected in OCSH:OTD0/OTD1.
If a compare match coincides with a write operation to OCCP0/OCCP1, the subsequent compare operation may not be executed properly. Therefore, before writing a value to OCCP0/OCCP1, check the counter value of the FRT or initialize the FRT.

■ 16-bit Output Compare Channel 0/1 Register (OCCP0/OCCP1)

Figure 27.3-4 16-bit Output Compare Channel 0/1 Register (OCCP0/OCCP1)

16-bit output compare channel 0 register (upper) (OCCP0H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare channel 0 register (lower) (OCCP0L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0065_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare channel 1 register (upper) (OCCP1H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0066_H									00000000_B
	R	R	R	R	R	R	R	R	

16-bit output compare channel 1 register (lower) (OCCP1L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0067_H									00000000_B
	R	R	R	R	R	R	R	R	

R : Read only

OCCP0/OCCP1 can only be accessed by the 16-bit word access.

The value of OCCP0/OCCP1 is to be compared with the FRT up/down counter value. In up count mode, if the FRT up/down counter value matches the value of OCCP0/OCCP1, the FRT will be reset to "0" at the next count clock edge. In up/down count mode, if the FRT up/down counter value matches the value of OCCP0/OCCP1, a compare signal will be generated.

OCCP0 and OCCP1 are located at the same addresses as their respective corresponding buffer registers, OCCPB0 and OCCPB1, respectively.

A read access to OCCP0/OCCP1 returns the current compare register value.

Use a word access instruction to access OCCP0/OCCP1. Do not use a read-modify-write (RMW) instruction to access OCCP0/OCCP1.

27.3.1.3 16-bit Output Compare Buffer Channel 0/1 Register (OCCPB0/1)

The 16-bit output compare buffer channel 0 register and the 16-bit output compare buffer channel 1 register are 16-bit buffer registers for OCCP0 and OCCP1 respectively.

■ 16-bit Output Compare Buffer Channel 0/1 Register (OCCPB0/1)

Figure 27.3-5 16-bit Output Compare Buffer Channel 0/1 Register (OCCPB0/1)

16-bit output compare buffer channel 0 register (upper) (OCCPB0H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064_H									00000000_B
	W	W	W	W	W	W	W	W	

16-bit output compare buffer channel 0 register (lower) (OCCPB0L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0065_H									00000000_B
	W	W	W	W	W	W	W	W	

16-bit output compare buffer channel 1 register (upper) (OCCPB0H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0066_H									00000000_B
	W	W	W	W	W	W	W	W	

16-bit output compare buffer channel 1 register (lower) (OCCPB0L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0067_H									00000000_B
	W	W	W	W	W	W	W	W	

W : Write only

OCCPB0/OCCPB1 can only be accessed by the 16-bit word access.

These buffer registers are updated by the CPU.

Immediately after the buffer function of a channel is disabled (EOCS:BUF1 = 0 or EOCS:BUF0 = 0) or the FRT is stopped, the value of an output compare buffer register will be transferred to its corresponding output compare register.

With the buffer function of a channel enabled (EOCS:BUF1 = 1 or EOCS:BUF0 = 1), the value of an output compare buffer register will be transferred to its corresponding output compare register when a compare clear match signal or a zero-detect signal is generated, depending on the settings of the BTS1 and BTS0 bits in the EOCS register.

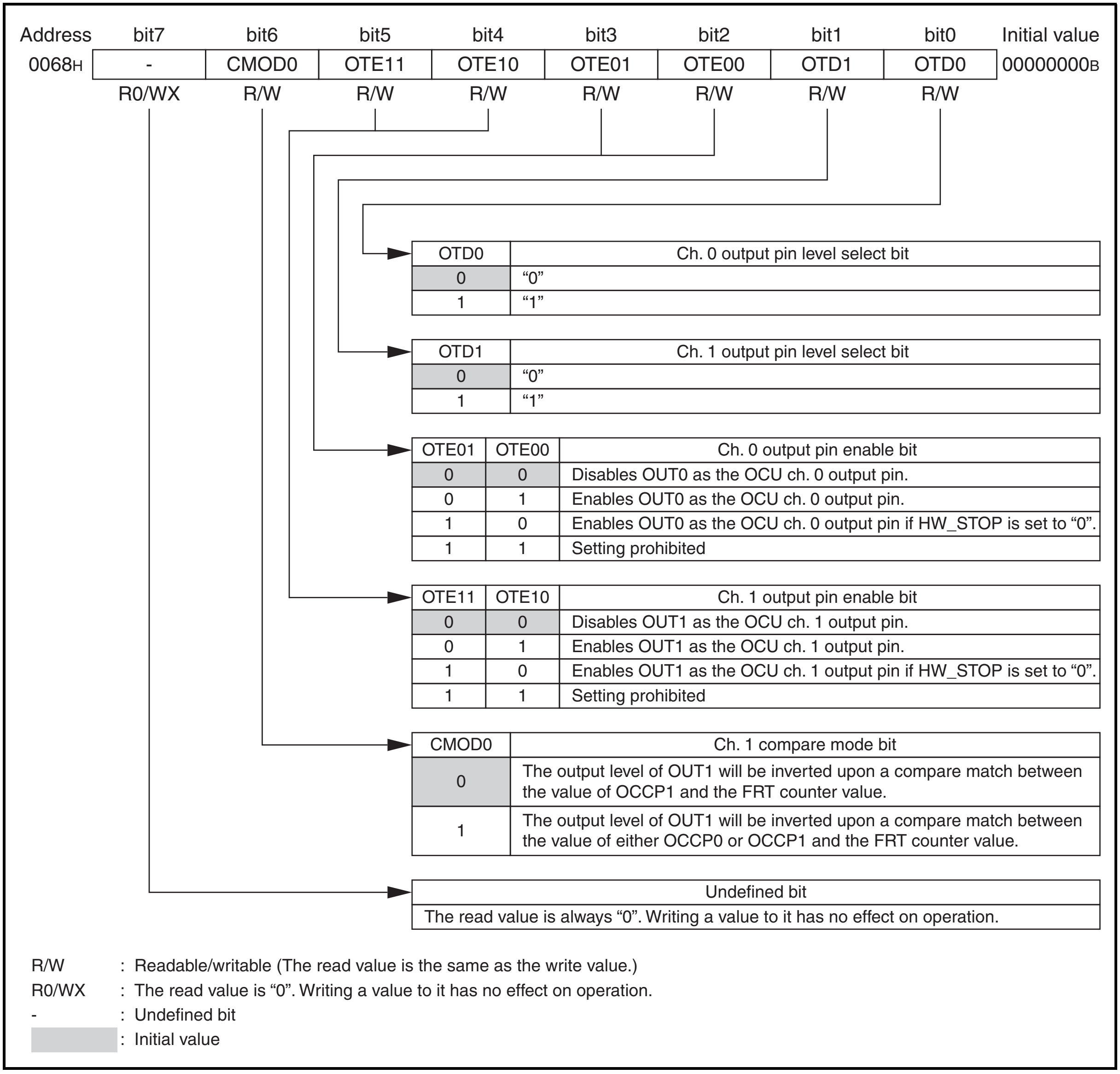
OCCPB0 and OCCPB1 are located at the same addresses as their respective corresponding output compare registers, OCCP0 and OCCP1, respectively.

27.3.1.4 16-bit Output Compare Control Status Register Upper/Lower (OCSH/OCSL)

The 16-bit output compare control status register upper (OCSH) and the 16-bit output compare control status register lower (OCSL) are used to set the OCU operating mode, start and stop output compare, control OCU interrupts and set the levels of external output pins.

■ 16-bit Output Compare Control Status Register Upper (OCSH)

Figure 27.3-6 16-bit Output Compare Control Status Register Upper (OCSH)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0068H	-	CMOD0	OTE11	OTE10	OTE01	OTE00	OTD1	OTD0	00000000B
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

OTD0	Ch. 0 output pin level select bit
0	"0"
1	"1"

OTD1	Ch. 1 output pin level select bit
0	"0"
1	"1"

OTE01	OTE00	Ch. 0 output pin enable bit
0	0	Disables OUT0 as the OCU ch. 0 output pin.
0	1	Enables OUT0 as the OCU ch. 0 output pin.
1	0	Enables OUT0 as the OCU ch. 0 output pin if HW_STOP is set to "0".
1	1	Setting prohibited

OTE11	OTE10	Ch. 1 output pin enable bit
0	0	Disables OUT1 as the OCU ch. 1 output pin.
0	1	Enables OUT1 as the OCU ch. 1 output pin.
1	0	Enables OUT1 as the OCU ch. 1 output pin if HW_STOP is set to "0".
1	1	Setting prohibited

CMOD0	Ch. 1 compare mode bit
0	The output level of OUT1 will be inverted upon a compare match between the value of OCCP1 and the FRT counter value.
1	The output level of OUT1 will be inverted upon a compare match between the value of either OCCP0 or OCCP1 and the FRT counter value.

Undefined bit
The read value is always "0". Writing a value to it has no effect on operation.

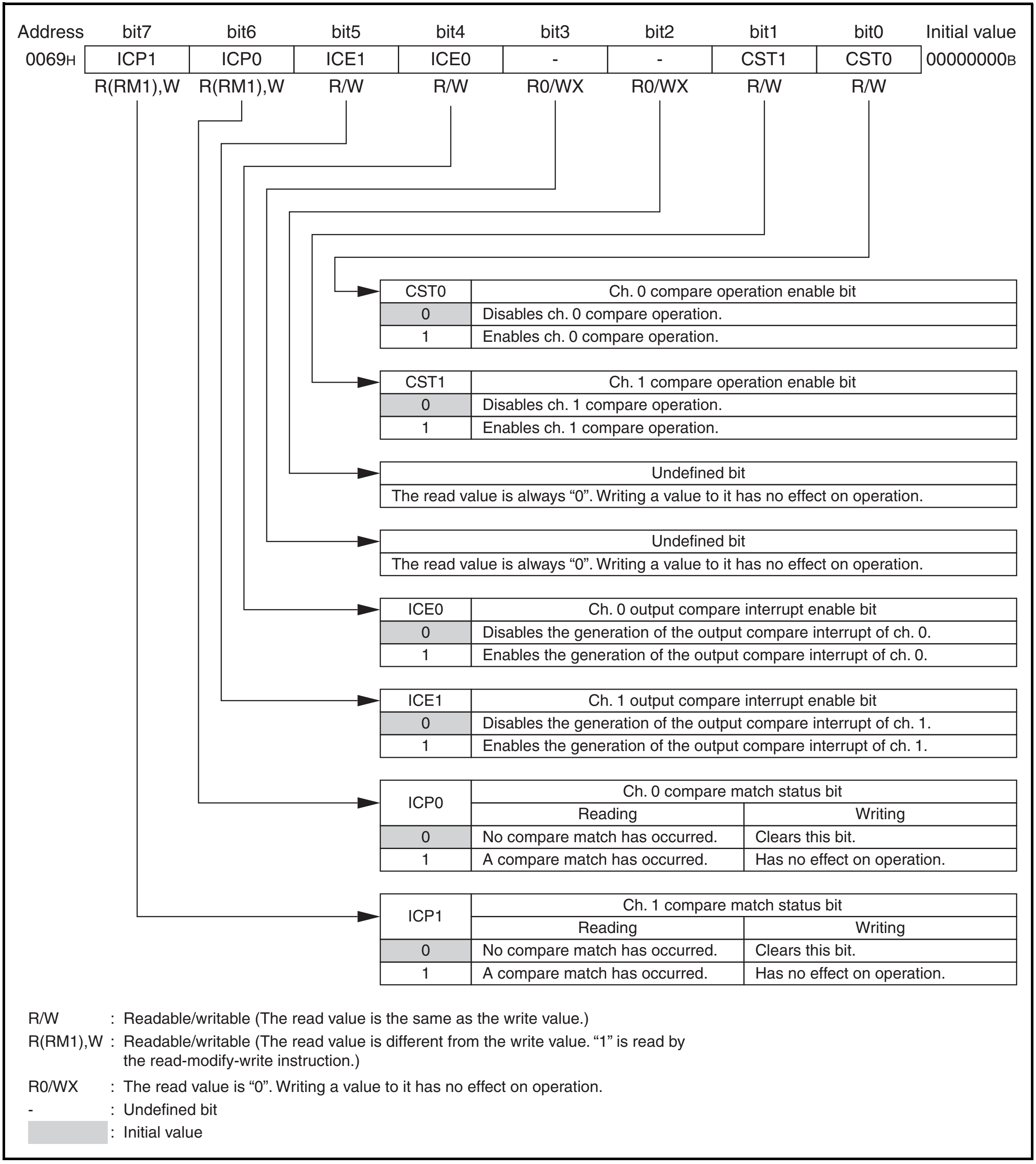
R/W	:	Readable/writable (The read value is the same as the write value.)
R0/WX	:	The read value is "0". Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	Initial value

Table 27.3-2 Functions of Bits in 16-bit Output Compare Control Status Register Upper (OCSH)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The settings of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit6	CMOD0: Ch. 1 compare mode bit	This bit is used to select how the pin output level will be inverted upon a compare match while pin output is enabled (OTE11, OTE10 = 01 or 10; OTE01, OTE00 = 01 or 10). When CMOD0 is set to "0", the output level of a pin will be inverted upon a compare match between the value of the compare register corresponding to that pin and the FRT up/down counter value. • OUT0: The output level will be inverted upon a compare match between the value of OCCP0 and the FRT up/down counter value. • OUT1: The output level will be inverted upon a compare match between the value of OCCP1 and the FRT up/down counter value When CMOD0 is set to "1", the output level of the pin OUT1 will be inverted upon a compare match between the value of OCCP0 or OCCP1 and the FRT up/down counter value. • OUT0: The output level will be inverted upon a compare match between the value of OCCP0 and the FRT up/down counter value. • OUT1: The output level will be inverted upon a compare match between the value of either OCCP0 or OCCP1 and the FRT up/down counter value.
bit5, bit4	OTE11, OTE10: Ch. 1 output pin enable bits	These bits are used to enable or disable the OCU ch. 1 output pin (OUT1). **Writing "00":** Disables OUT1 as the OCU ch. 1 output pin. OUT1 can be used as a general-purpose port or the comparator ch. 0 negative input pin. **Writing "01":** Enables OUT1 as the OCU ch. 1 output pin. The setting of EOCS:HW_STOP has no effect on OCU ch. 1 output pin. **Writing "10":** Enables OUT1 as the OCU ch. 1 output pin if HW_STOP is set to "0". If HW_STOP is set to "1", OUT1 can be used as a general-purpose port or the comparator ch. 0 negative input pin. **Writing "11":** Is prohibited.
bit3, bit2	OTE01, OTE00: Ch. 0 output pin enable bits	These bits are used to enable or disable the OCU ch. 0 output pin (OUT0). **Writing "00":** Disables OUT0 as the OCU ch. 0 output pin. OUT0 can be used as a general-purpose port or the comparator ch. 0 positive input pin. **Writing "01":** Enables OUT0 as the OCU ch. 0 output pin. The setting of EOCS:HW_STOP has no effect on OCU ch. 0 output pin. **Writing "10":** Enables OUT0 as the OCU ch. 0 output pin if HW_STOP is set to "0". If HW_STOP is set to "1", OUT0 can be used as a general-purpose port or the comparator ch. 0 positive input pin. **Writing "11":** Is prohibited.
bit1	OTD1: Ch. 1 output pin level select bit	This bit is used to change the output level of the OUT1 pin when the pin is enabled. The initial value of OUT1 is "0". To modify this bit, disables the compare operation by setting OCSL:CST1 to "0". This bit returns the value of OUT1 when read. The value will be inverted before being output from the OCU if OCMCR:INV1 is set to "1". **Writing "0":** Sets OUT1 to "0". OUT1 will output "0" if OCMCR:INV1 is set to "0", and will output "1" if OCMCR:INV1 is set to "1". **Writing "1":** Sets OUT1 to "1". OUT1 will output "1" if OCMCR:INV1 is set to "0", and will output "0" if OCMCR:INV1 is set to "1".
bit0	OTD0: Ch. 0 output pin level select bit	This bit is used to change the output level of the OUT0 pin when the pin is enabled. The initial value of OUT0 is "0". To modify this bit, disables the compare operation by setting OCSL:CST0 to "0". This bit returns the value of OUT0 when read. The value will be inverted before being output from the OCU if OCMCR:INV0 is set to "1". **Writing "0":** Sets OUT0 to "0". OUT0 will output "0" if OCMCR:INV0 is set to "0", and will output "1" if OCMCR:INV0 is set to "1". **Writing "1":** Sets OUT0 to "1". OUT0 will output "1" if OCMCR:INV0 is set to "0", and will output "0" if OCMCR:INV0 is set to "1".

■ 16-bit Output Compare Control Status Register Lower (OCSL)

Figure 27.3-7 16-bit Output Compare Control Status Register Lower (OCSL)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0069H	ICP1	ICP0	ICE1	ICE0	-	-	CST1	CST0	00000000B
	R(RM1),W	R(RM1),W	R/W	R/W	R0/WX	R0/WX	R/W	R/W	

CST0	Ch. 0 compare operation enable bit
0	Disables ch. 0 compare operation.
1	Enables ch. 0 compare operation.

CST1	Ch. 1 compare operation enable bit
0	Disables ch. 1 compare operation.
1	Enables ch. 1 compare operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

ICE0	Ch. 0 output compare interrupt enable bit
0	Disables the generation of the output compare interrupt of ch. 0.
1	Enables the generation of the output compare interrupt of ch. 0.

ICE1	Ch. 1 output compare interrupt enable bit
0	Disables the generation of the output compare interrupt of ch. 1.
1	Enables the generation of the output compare interrupt of ch. 1.

ICP0	Ch. 0 compare match status bit	
	Reading	Writing
0	No compare match has occurred.	Clears this bit.
1	A compare match has occurred.	Has no effect on operation.

ICP1	Ch. 1 compare match status bit	
	Reading	Writing
0	No compare match has occurred.	Clears this bit.
1	A compare match has occurred.	Has no effect on operation.

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 instruction.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 27.3-3 Functions of Bits in 16-bit Output Compare Control Status Register Lower (OCSL)

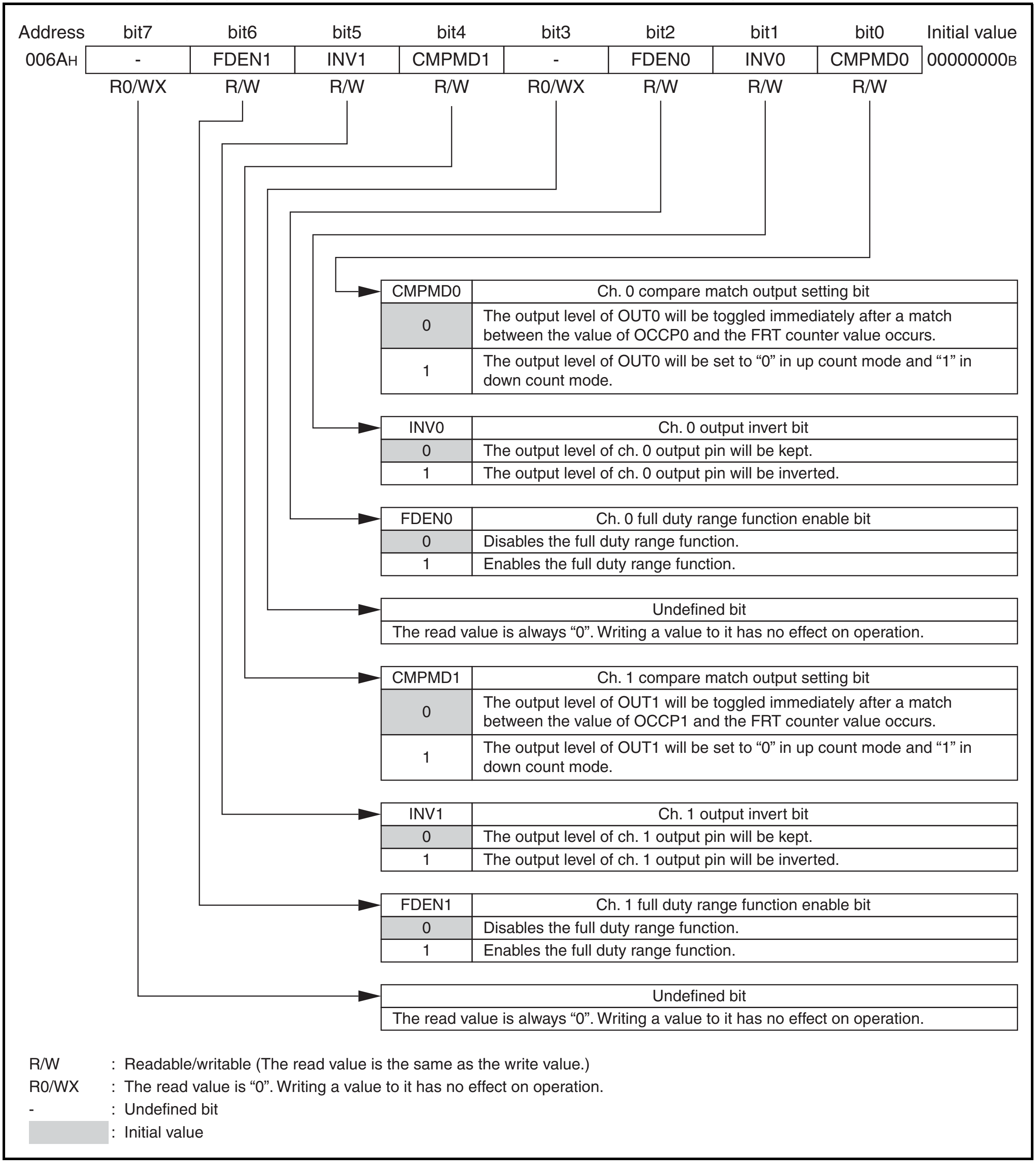
Bit name		Function
bit7	ICP1: Ch. 1 compare match status bit	This bit is used to indicate compare match status. When the value of OCCP1 matches the FRT up/down counter value, ICP1 will be set to "1". **Reading "0":** Indicates that no compare match has occurred. **Reading "1":** Indicates that a compare match has occurred. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by a read-modify-write (RMW) instruction, this bit always returns "1".
bit6	ICP0: Ch. 0 compare match status bit	This bit is used to indicate compare match status. When the value of OCCP0 matches the FRT up/down counter value, ICP0 will be set to "1". **Reading "0":** Indicates that no compare match has occurred. **Reading "1":** Indicates that a compare match has occurred. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by a read-modify-write (RMW) instruction, this bit always returns "1".
bit5	ICE1: Ch. 1 output compare interrupt enable bits	This bit is used to enable or disable the generation of the output compare interrupt of ch. 1. With this bit set to "1", an output compare interrupt of ch. 1 will be generated when a compare match occurs on ch. 1. **Writing "0":** Disables the generation of the output compare interrupt of ch. 1. **Writing "1":** Enables the generation of the output compare interrupt of ch. 1.
bit4	ICE0: Ch. 0 output compare interrupt enable bits	This bit is used to enable or disable the generation of the output compare interrupt of ch. 0. With this bit set to "1", an output compare interrupt of ch. 0 will be generated when a compare match occurs on ch. 0. **Writing "0":** Disables the generation of the output compare interrupt of ch. 0. **Writing "1":** Enables the generation of the output compare interrupt of ch. 0.
bit3, bit2	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit1	CST1: Ch. 1 compare operation enable bit	This bit is used to enable or disable ch. 1 compare operation. **Writing "0":** Disables ch. 1 compare operation. **Writing "1":** Enables ch. 1 compare operation. Note: Write a value to OCCP1 before enabling ch. 1 compare operation. Since the OCU is synchronized with the clock of the FRT, stopping the FRT will also stop the compare operation of the OCU.
bit0	CST0: Ch. 0 compare operation enable bit	This bit is used to enable or disable ch. 0 compare operation. **Writing "0":** Disables ch. 0 compare operation. **Writing "1":** Enables ch. 0 compare operation. Note: Write a value to OCCP0 before enabling ch. 0 compare operation. Since the OCU is synchronized with the clock of the FRT, stopping the FRT will also stop the compare operation of the OCU.

27.3.1.5 16-bit Output Compare Mode Control Register (OCMCR)

This section describes details of the 16-bit output compare mode control register.

■ 16-bit Output Compare Mode Control Register (OCMCR)

Figure 27.3-8 16-bit Output Compare Mode Control Register (OCMCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006A$_H$	-	FDEN1	INV1	CMPMD1	-	FDEN0	INV0	CMPMD0	00000000$_B$
	R0/WX	R/W	R/W	R/W	R0/WX	R/W	R/W	R/W	

CMPMD0	Ch. 0 compare match output setting bit
▨ 0	The output level of OUT0 will be toggled immediately after a match between the value of OCCP0 and the FRT counter value occurs.
1	The output level of OUT0 will be set to “0” in up count mode and “1” in down count mode.

INV0	Ch. 0 output invert bit
▨ 0	The output level of ch. 0 output pin will be kept.
1	The output level of ch. 0 output pin will be inverted.

FDEN0	Ch. 0 full duty range function enable bit
▨ 0	Disables the full duty range function.
1	Enables the full duty range function.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

CMPMD1	Ch. 1 compare match output setting bit
▨ 0	The output level of OUT1 will be toggled immediately after a match between the value of OCCP1 and the FRT counter value occurs.
1	The output level of OUT1 will be set to “0” in up count mode and “1” in down count mode.

INV1	Ch. 1 output invert bit
▨ 0	The output level of ch. 1 output pin will be kept.
1	The output level of ch. 1 output pin will be inverted.

FDEN1	Ch. 1 full duty range function enable bit
▨ 0	Disables the full duty range function.
1	Enables the full duty range function.

Undefined bit
The read value is always “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.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
▨	:	Initial value

Table 27.3-4 Functions of Bits in 16-bit Output Compare Mode Control Register (OCMCR) (1 / 2)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit6	FDEN1: Ch. 1 full duty range function enable bit	This bit is used to enable or disable the full duty range feature. **Writing "0":** Disables the full duty range feature. **Writing "1":** Enables the full duty range feature. In the case of FDEN1 = 0: • The output level of OUT1 will be toggled when a match between the value of OCCP1 and the FRT up/down counter value occurs. In the case of FDEN1 = 1: • When OCCP1 is "0000$_H$", the output level of OUT1 is independent of the FRT up/down counter value, but depends on the setting of OCMCR:INV1. • When the value of OCCP1 is bigger than "0000$_H$" and smaller than the value of CPCLR of the FRT, the output level of OUT1 depends on the setting of OCMCR:CMPDM1. • When the value of OCCP1 is bigger than the value of CPCLR of the FRT, the output level of OUT1 depends on the OCU compare flag (CMP_FLAG) at a compare clear. With CMP_FLAG set to "0", the level of OUT1 depends on the setting OCMCR:INV1. With CMP_FLAG set to "1", the level of OUT1 is to be output as it is. CMP_FLAG will be set at a compare match and reset at a compare clear.
bit5	INV1: Ch. 1 output invert bit	This bit is used to invert the output level of ch. 1 output pin (OUT1). **Writing "0":** The output level of OUT1 will be kept. **Writing "1":** The output level of OUT1 will be inverted.
bit4	CMPMD1: Ch. 1 compare match output setting bit	This bit is used to set how to toggle the output level of the OUT1 pin immediately after a match between the value of OCCP1 and the FRT up/down counter value occurs with OTE11 and OTE10 set to "01 or "10". The above output toggle function can only be used when OCSH:CMOD0 is set to "0" and OCMCR:FDEN1 to "1". In the case of CMPMD1 = 0: • The output level of OUT1 will toggled immediately after a match between the value of OCCP1 and the FRT up/down counter value occurs. In the case of CMPMD1 = 1: • The output level of OUT1 will be set to "0" in up count mode and "1" in down count mode.
bit3	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit2	FDEN0: Ch. 0 full duty range function enable bit	This bit is used to enable or disable the full duty range feature. **Writing "0":** Disables the full duty range feature. **Writing "1":** Enables the full duty range feature. In the case of FDEN0 = 0: • The output level of OUT0 will be toggled when a match between the value of OCCP0 and the FRT up/down counter value occurs. In the case of FDEN0 = 1: • When OCCP0 is "0000$_H$", the output level of OUT0 is independent of the FRT up/down counter value, but depends on the setting of OCMCR:INV0. • When the value of OCCP0 is bigger than "0000$_H$" and smaller than the value of CPCLR of the FRT, the output level of OUT0 depends on the setting of OCMCR:CMPDM0. • When the value of OCCP0 is bigger than the value of CPCLR of the FRT, the output level of OUT0 depends on the OCU compare flag (CMP_FLAG) at a compare clear. With CMP_FLAG set to "0", the level of OUT0 depends on the setting OCMCR:INV0. With CMP_FLAG set to "1", the level of OUT0 is to be output as it is. CMP_FLAG will be set at a compare match and reset at a compare clear.
bit1	INV0: Ch. 0 output invert bit	This bit is used to invert the output level of ch. 0 output pin (OUT0). **Writing "0":** The output level of OUT0 will be kept. **Writing "1":** The output level of OUT0 will be inverted.

Table 27.3-4 Functions of Bits in 16-bit Output Compare Mode Control Register (OCMCR) (2 / 2)

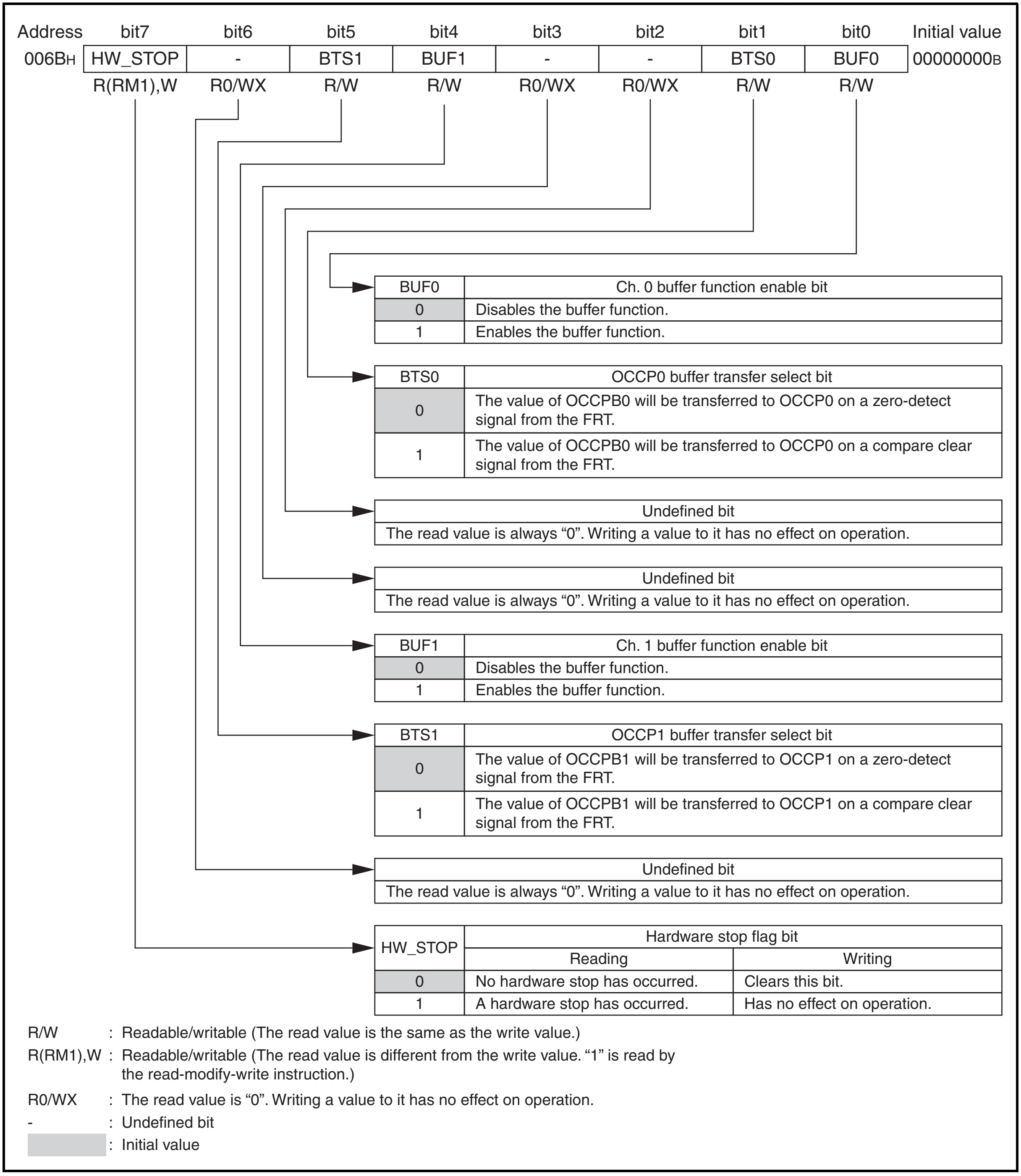
Bit name		Function
bit0	CMPMD0: Ch. 0 compare match output setting bit	This bit is used to set how to toggle the output level of the OUT0 pin immediately after a match between the value of OCCP0 and the FRT up/down counter value occurs with OTE01 and OTE00 set to "01 or "10". The above output toggle function can only be used when OCSH:CMOD0 is set to "0" and OCMCR:FDEN0 to "1". In the case of CMPMD0 = 0: • The output level of OUT0 will toggled immediately after a match between the value of OCCP0 and the FRT up/down counter value occurs. In the case of CMPMD0 = 1: • The output level of OUT0 will be set to "0" in up count mode and "1" in down count mode.

27.3.1.6 16-bit Output Compare Extended Control Status Register (EOCS)

This section describes details of the 16-bit output compare extended control status register.

■ 16-bit Output Compare Extended Control Status Register (EOCS)

Figure 27.3-9 16-bit Output Compare Extended Control Status Register (EOCS)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006BH	HW_STOP	-	BTS1	BUF1	-	-	BTS0	BUF0	00000000B
	R(RM1),W	R0/WX	R/W	R/W	R0/WX	R0/WX	R/W	R/W	

BUF0	Ch. 0 buffer function enable bit
0	Disables the buffer function.
1	Enables the buffer function.

BTS0	OCCP0 buffer transfer select bit
0	The value of OCCPB0 will be transferred to OCCP0 on a zero-detect signal from the FRT.
1	The value of OCCPB0 will be transferred to OCCP0 on a compare clear signal from the FRT.

Undefined bit
The read value is always "0". Writing a value to it has no effect on operation.

Undefined bit
The read value is always "0". Writing a value to it has no effect on operation.

BUF1	Ch. 1 buffer function enable bit
0	Disables the buffer function.
1	Enables the buffer function.

BTS1	OCCP1 buffer transfer select bit
0	The value of OCCPB1 will be transferred to OCCP1 on a zero-detect signal from the FRT.
1	The value of OCCPB1 will be transferred to OCCP1 on a compare clear signal from the FRT.

Undefined bit
The read value is always "0". Writing a value to it has no effect on operation.

HW_STOP	Hardware stop flag bit	
	Reading	Writing
0	No hardware stop has occurred.	Clears this bit.
1	A hardware stop has occurred.	Has no effect on operation.

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 instruction.)
R0/WX : The read value is "0". Writing a value to it has no effect on operation.
- : Undefined bit
▨ : Initial value

Table 27.3-5 Functions of Bits in 16-bit Output Compare Extended Control Status Register (EOCS)

Bit name		Function
bit7	HW_STOP: Hardware stop flag bit	This bit is used to indicate whether a hardware stop has occurred. The hardware stop trigger is generated by the internal voltage comparator. **Reading "0":** Indicates that no hardware stop has occurred. **Reading "1":** Indicates that a hardware stop has occurred. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation.
bit6	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit5	BTS1: OCCP1 buffer transfer select bit	This bit is used to select when the value of OCCPB1 will be transferred to OCCP1. **Writing "0":** The value of OCCPB1 will be transferred to OCCP1 on a zero-detect signal from the FRT. **Writing "1":** The value of OCCPB1 will be transferred to OCCP1 on a compare clear signal from the FRT.
bit4	BUF1: Ch. 1 buffer function enable bit	This bit is used to enable or disable the buffer function of ch. 1. **Writing "0":** Disables the buffer function of ch. 1. **Writing "1":** Enables the buffer function of ch. 1.
bit3, bit2	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation. The setting of this bit has no effect on the read-modify-write (RMW) type of instruction.
bit1	BTS0: OCCP0 buffer transfer select bit	This bit is used to select when the value of OCCPB0 will be transferred to OCCP0. **Writing "0":** The value of OCCPB0 will be transferred to OCCP0 on a zero-detect signal from the FRT. **Writing "1":** The value of OCCPB0 will be transferred to OCCP0 on a compare clear signal from the FRT.
bit0	BUF0: Ch. 0 buffer function enable bit	This bit is used to enable or disable the buffer function of ch. 0. **Writing "0":** Disables the buffer function of ch. 0. **Writing "1":** Enables the buffer function of ch. 0.

27.3.2 Operations of OCU

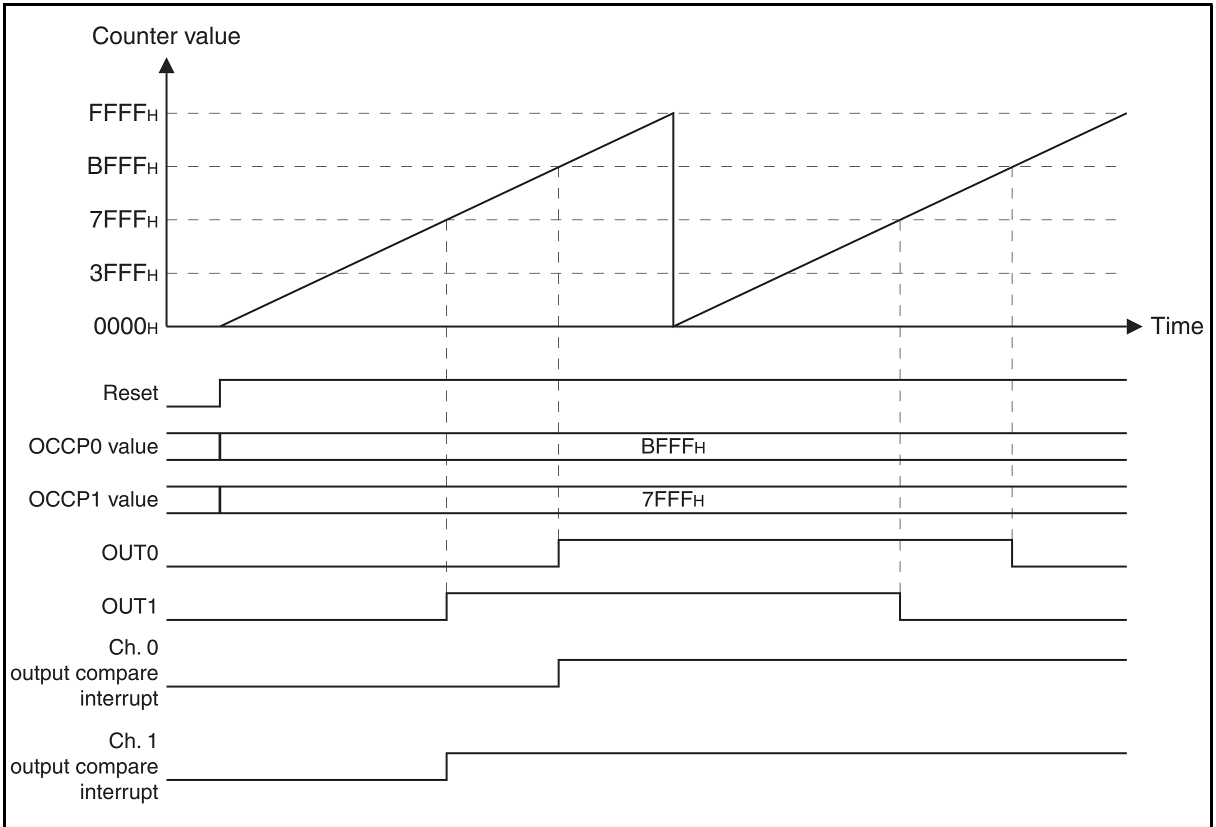
When the value of OCCP0/OCCP1 matches the FRT up/down counter value, an interrupt request flag will be set and the output level of an output pin (OUT0/ OUT1) will be toggled. In addition, the CMOD0 bit can be used to select how the output level of OUT0/OUT1 will be inverted upon a compare match while pin output is enabled.

■ Sample Output Waveform with CMOD0 = 0

With CMOD0 = 0, the output level of a pin corresponding to an output compare register is inverted whenever a compare match between that output compare register and the FRT up/ down counter value occurs. The output level of a pin is governed by only one output compare register.

- OUT0: The output level is inverted whenever a compare match between the value of OCCP0 and the FRT up/down counter value.
- OUT1: The output level is inverted whenever a compare match between the value of OCCP1 and the FRT up/down counter value.

Figure 27.3-10 Sample Output Waveform with CMOD0 = 0



Note: In this figure, the initial values of OUT0 and OUT1 are “0”.

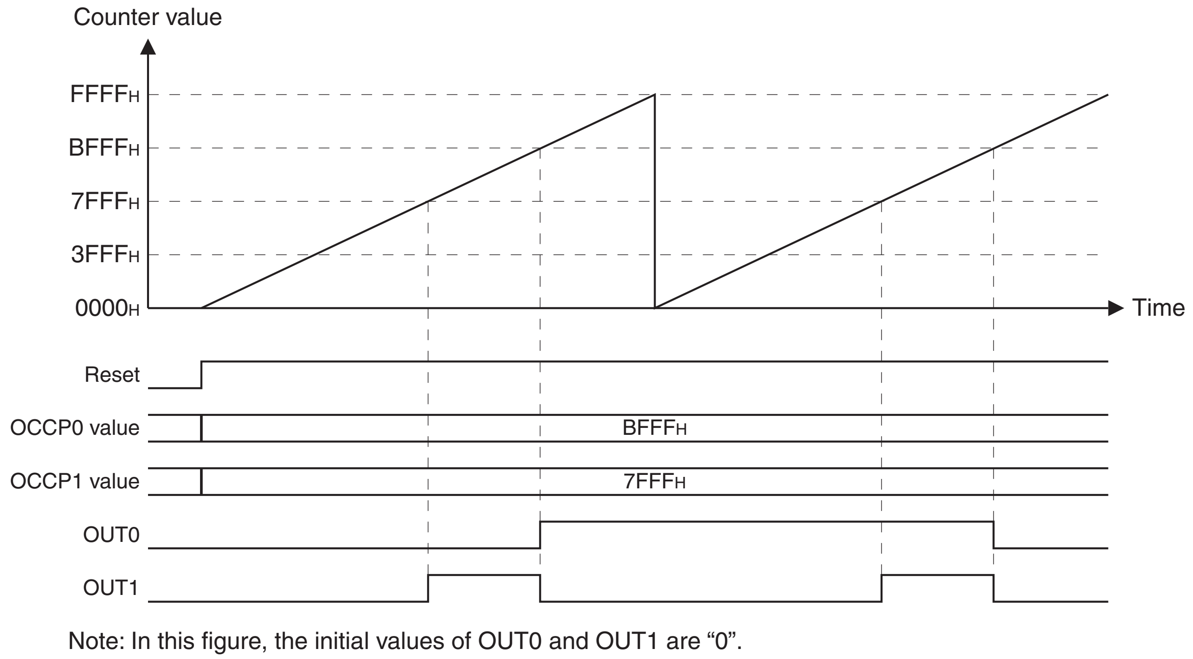
■ Sample Output Waveform of OCCP0 and OCCP1 with CMOD0 = 1

With CMOD0 = 1, the output level of a pin corresponding to an output compare register is inverted whenever a match between that output compare register and the FRT up/down counter value occurs. This is identical to the operation of CMOD0 = 0.

However, the output level of OUT1 behaves differently when CMOD0 = 1. It will be inverted upon a match between the value of OCCP0 or OCCP1 and the FRT up/down counter value. This mechanism enables setting a pulse signal with one edge to be governed by OCCP0 and the other by OCCP1 or vice versa. If both OCCP0 and OCCP1 have the same value, the resulting operation is identical to that of CMOD0 = 0.

A PWM signal with changing frequency can be defined by using together the operation mentioned above (CMOD0 = 1) and the option of initializing the FRT up/down counter upon a compare match (TCCSL:MODE = 1).

Figure 27.3-11 Sample Output Waveform of OCCP0 and OCCP1 with CMOD0 = 1

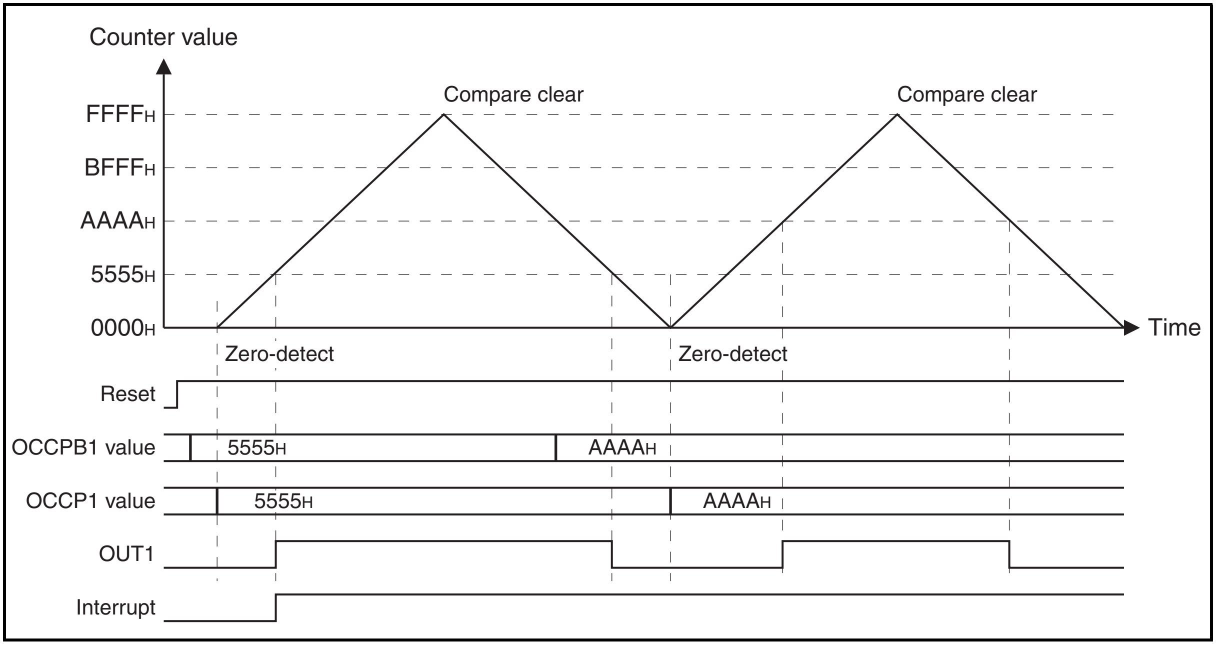


Note: In this figure, the initial values of OUT0 and OUT1 are “0”.

■ Sample Output Waveform with BTS1 = 0 and FDEN1 = 0

With BTS1set to "0", the value of OCCPB1 will be transferred to OCCP1 when the FRT up/down counter value reaches "0000$_H$".

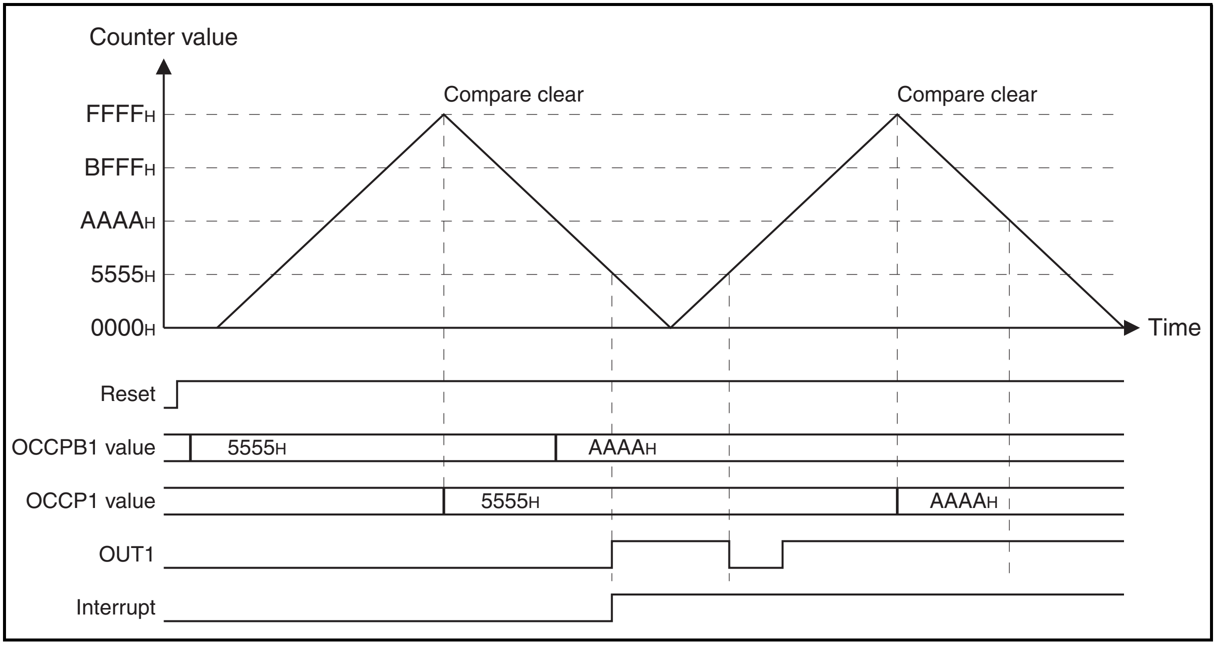
Figure 27.3-12 Sample Output Waveform with BTS1 = 0 and FDEN1 = 0



■ Sample Output Waveform with BTS1 = 1 and FDEN1 = 0

With BTS1set to "1", the value of OCCPB1 will be transferred to OCCP1 when a compare clear match occurs in the FRT.

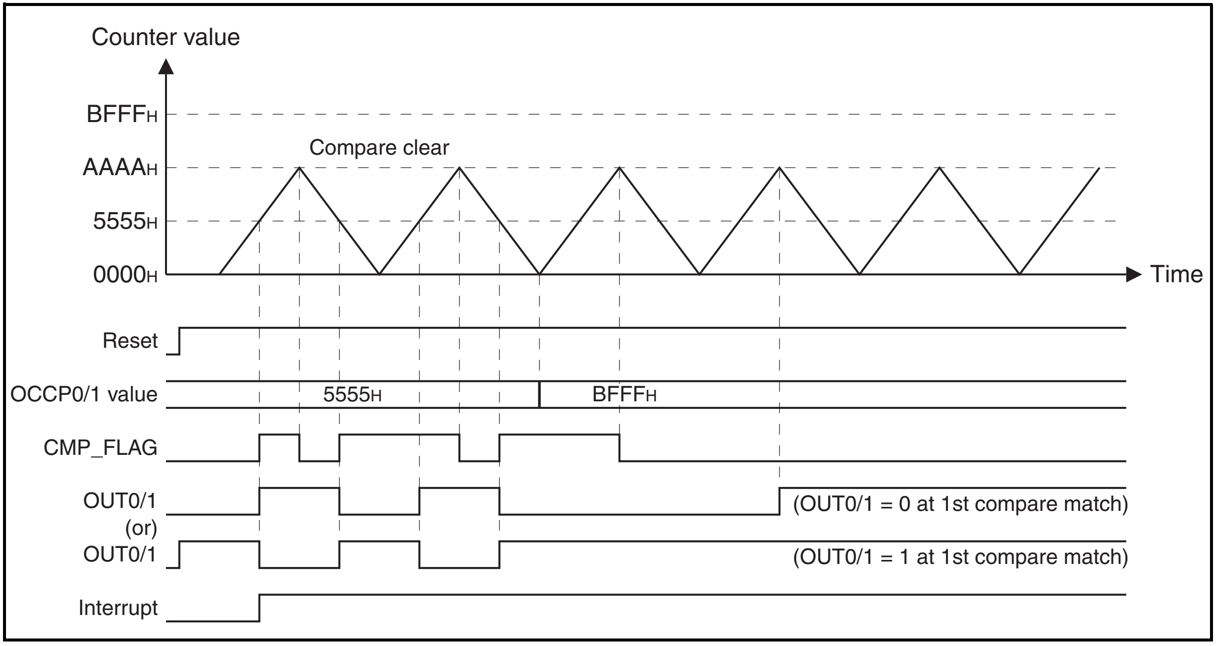
Figure 27.3-13 Sample Output Waveform with BTS1 = 1 and FDEN1 = 0



■ Sample Output Waveform with FDEN0/1 = 1, CMPMD0/1 = 0 and INV0/1 = 0

With CMPMD0/1 set to "0", the output level of OUT0/1 will be toggled when the value of an output compare register (OCCP0/1) of the OCU matches the FRT up/down counter value. The output waveform of OUT0/1 depends on the output level of OUT0/1 at the first match between OCCP0/1 and the FRT up/down counter value. The value of OUT0/1 is controlled by the settings of OCSH:OTD0/1 respectively. If OCMCR:INV0/1 is set to "1", then the output level of OUT0/1 will be inverted. The compare match flag, CMP_FLAG (see details of FDEN0/1 in Table 27.3-4), is set to "1" if a compare match occurs and "0" if a compare clear occurs.

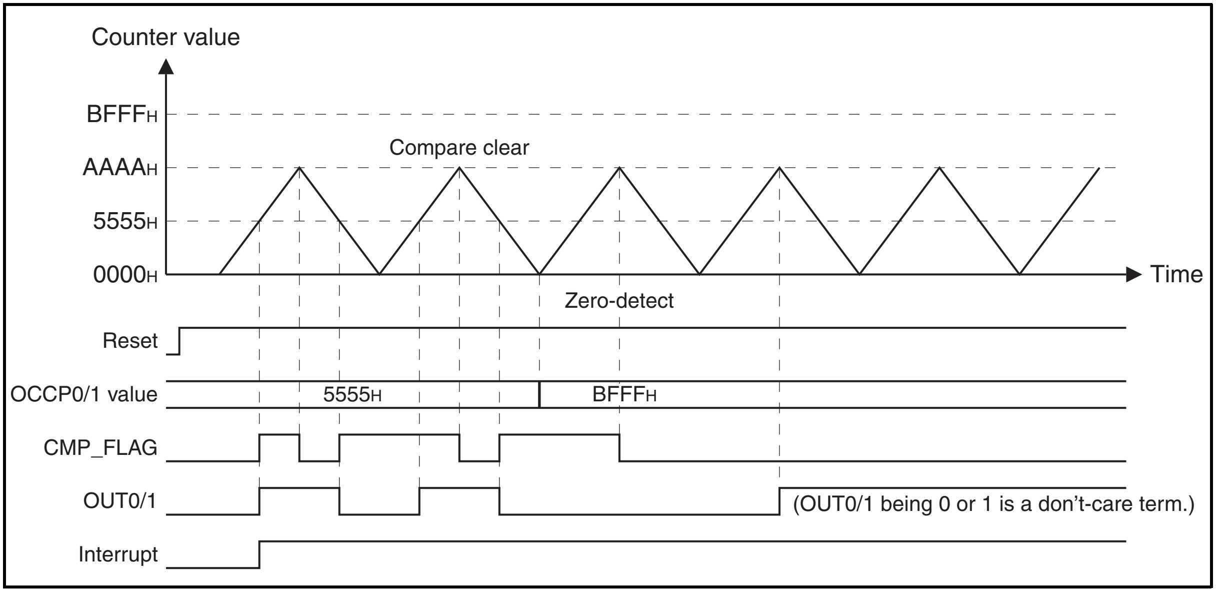
Figure 27.3-14 Sample Output Waveform with FDEN0/1 = 1, CMPMD0/1 = 0 and INV0/1 = 0



■ Sample Output Waveform with FDEN0/1 = 1, CMPMD0/1 = 1 and INV0/1 = 0

With CMPMD0/1 set to "1" and INV0/1 set to "0", OUT0/1 will become "1" when a compare match occurs in up count mode, and will become "0" when a compare match occurs in down count mode. In the waveform (see Figure 27.3-15 below), at the first compare match, the output level of OUT0/1 is a don’t-care term. If OCMCR:INV0/1 is set to "1", then the output level of OUT0/1 will be inverted. The compare match flag, CMP_FLAG (see details of FDEN0/1 in Table 27.3-4), is set to "1" if a compare match occurs and "0" if a compare clear occurs.

Figure 27.3-15 Sample Output Waveform with FDEN0/1 = 1, CMPMD0/1 = 1 and INV0/1 = 0

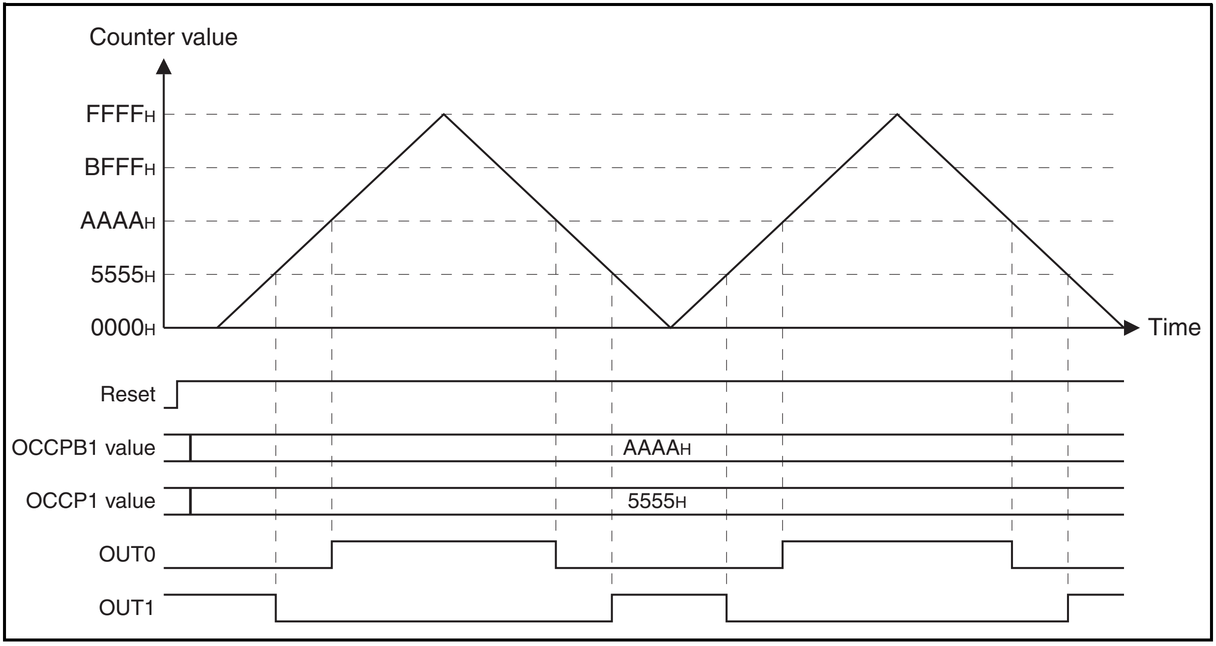


■ Dead Time Feature

In up/down count mode, a waveform as shown in Figure 27.3-16 will form under the following settings of registers and pins.

- CPCLRH and CPCLRL of the FRT: $FFFF_H$
- OCCP0 of the OCU: $AAAA_H$
- OCCP1 of the OCU: 5555_H
- Initial value of OUT0: 1
- Initial value of OUT1: 0

Figure 27.3-16 Dead Time Feature Waveform

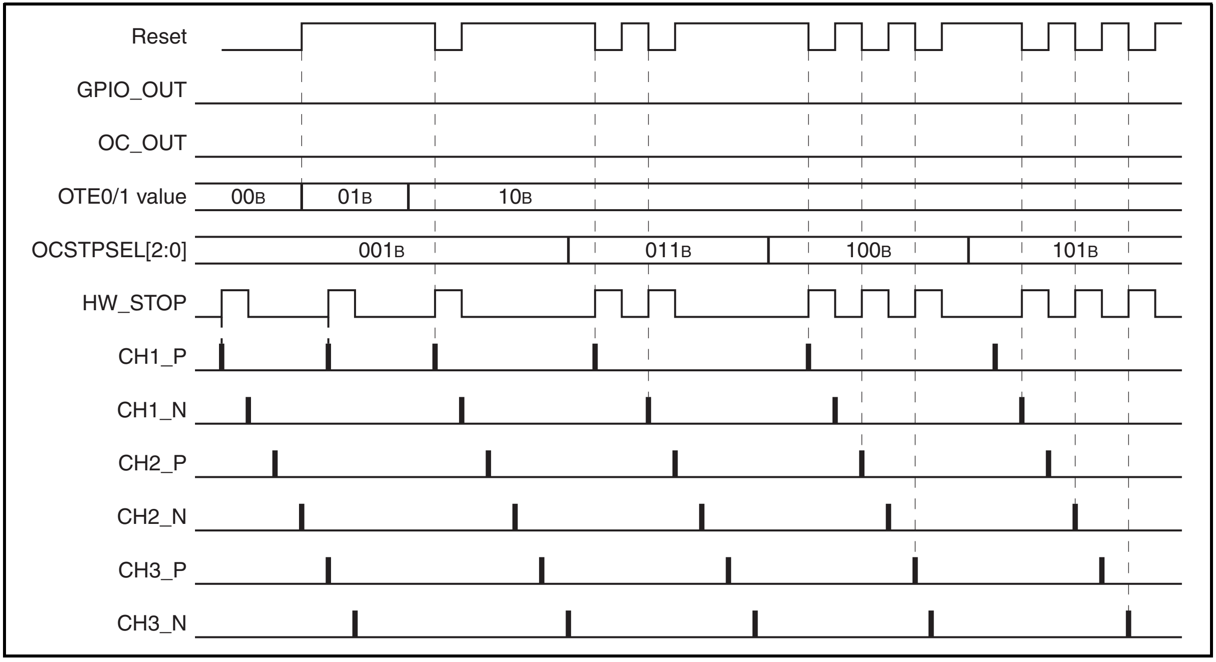


■ Hardware Stop Function

The hardware stop function is used to control how the output compare value is output to the output compare output pin OC_OUT when OCSH:OTE11,OTE10 or OCSH:OTE01,OTE00 are set to "10_B". The hardware stop trigger is generated from a rising edge or a falling edge of CHx_P or CHx_N of the voltage comparator, which can be selected by using OCUOU:OCSTPSEL[2:0]. If "000_B" is written to OCUOU:OCSTPSEL[2:0], no source will be supplied for the hardware stop trigger. OC_OUT and GPIO_OUT share the same pin, and OC_OUT has priority over GPIO_OUT. The waveform of the hardware stop function shown below assumes that GPIO_OUT is "0" and OC_OUT "1" for better illustration.

- If OTE11,10 or OTE01,00 are set to "00_B", outputting OC_OUT to PIN is disabled and PIN is driven by GPIO_OUT.
- If OTE11,10 or OTE01,00 are set to "01_B", outputting OC_OUT to PIN is enabled and PIN is driven by OC_OUT.
- If OTE11,10 or OTE01,00 are set to "10_B", outputting OC_OUT to PIN is enabled only when HW_STOP is set to "0".

Figure 27.3-17 Hardware Stop Function Waveform



Notes:

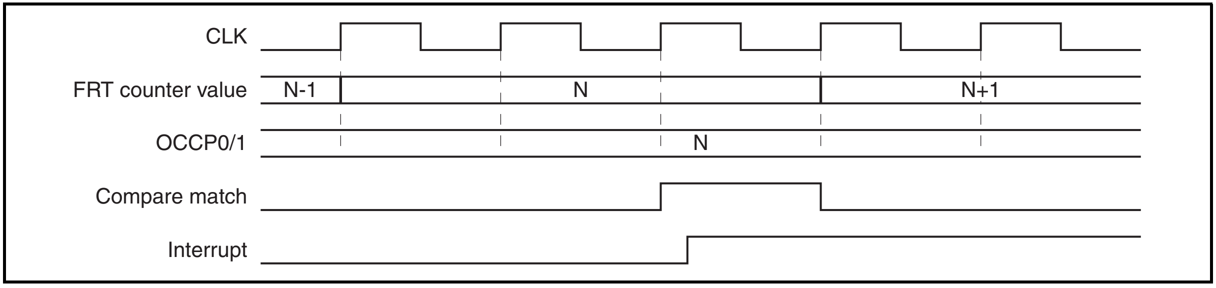
- Before turning off the voltage comparator, disable OCU output and select no source as a hardware stop trigger by writing "000" to OCUOR:OCSTPSEL[2:0], then clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF).
- After turning on the voltage comparator, clear the interrupt flag of the voltage comparator (CMR0/1/2/3:IF), then enable OCU output and select a rising/falling edge of the voltage comparator as a hardware stop trigger.

■ Output Compare Timing

In the output compare operation, a compare match signal is generated when a compare match, in which the value of an output compare register (OCCP0/OCCP1) matches the FRT counter value, occurs. Upon the generation of the compare match signal, depending on settings of registers related to output level inversion and interrupt generation, the output level can be toggled and an interrupt can be generated. The timing of output level inversion upon a compare match is synchronized with the FRT counter timing.

● Interrupt timing

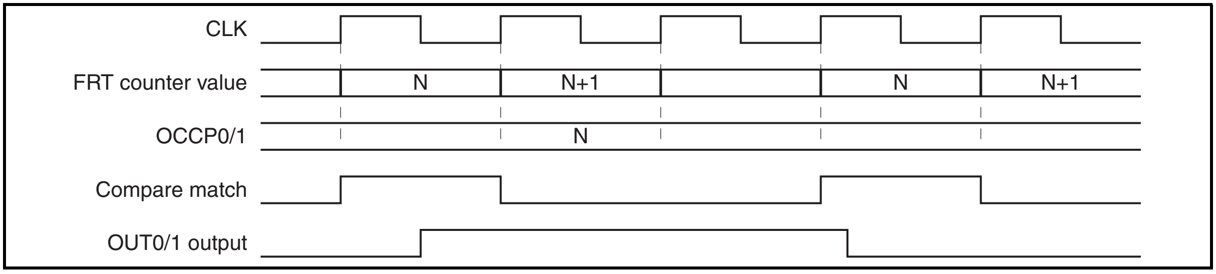
Figure 27.3-18 Interrupt Timing Graph



● Output pin level change timing

The figure below illustrates how the level of the output pin changes in the default configuration of the OCU pin output function.

Figure 27.3-19 Output Pin Level Change Timing Graph



■ Register and Vector Table Addresses Related to Interrupts of OCU

Table 27.3-6 Registers and Vector Table Addresses Related to Interrupts of OCU

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Output compare ch. 0 match	IRQ07	ILR1	L07	$FFEC_H$	$FFED_H$
Output compare ch. 1 match	IRQ08	ILR2	L08	$FFEA_H$	$FFEB_H$

ch.: Channel

See "APPENDIX B Table of Interrupt Sources" for the respective interrupt request numbers and vector table addresses of different peripheral functions.

CHAPTER 28

DUAL OPERATION FLASH MEMORY

This chapter describes the function and operations of the 64/96/160 kbit dual operation Flash memory.

28.1 Overview of Dual operation Flash Memory

Dual operation Flash memory is located at B000$_H$ to BFFF$_H$ and F000$_H$ to FFFF$_H$ for 64 kbit Flash memory, at B000$_H$ to BFFF$_H$ and E000$_H$ to FFFF$_H$ for 96 kbit Flash memory or at B000$_H$ to FFFF$_H$ for 160 kbit Flash memory on the CPU memory map.
The dual operation Flash memory consists of an upper bank and a lower bank*. Unlike conventional Flash products, writing/erasing data to/from one bank and reading data from another bank can be executed simultaneously.

*** MB95F432H/F432K:**
upper bank 4 Kbyte × 1; lower bank: 2 Kbyte × 2
MB95F433H/F433K:
upper bank 8 Kbyte × 1; lower bank: 2 Kbyte × 2
MB95F434H/F434K:
upper bank: 16 Kbyte × 1; lower bank: 2 Kbyte × 2

■ Overview of Dual Operation Flash Memory

The following methods can be used to write data into and erase data from the Flash memory:

- Writing/erasing using a dedicated serial programmer
- Writing/erasing by program execution

Since data can be written into and erased from the Dual operation Flash memory by instructions from the CPU via the Flash memory interface circuit, program code and data can be efficiently updated with the device mounted on a circuit board. The minimum sector size of the dual operation Flash is 2 Kbyte, which is a sector configuration facilitating the management of the program/data area.

Data can be updated by executing a program in RAM or by executing a program in the Flash memory in dual operation mode. The erase/write operation and the read operation can be executed in different banks (upper bank/lower bank) simultaneously.

The dual operation flash can use the following combinations:

Upper bank	Lower bank
Read	
Read	Program/sector erase
Program/sector erase	Read
Chip erase	

The bank on one side cannot be programmed/sector-erased while the bank on the other side is being programmed/erased.

■ Features of Dual Operation Flash Memory

- Sector configuration:
  - 8 Kbyte × 8 bits (4 Kbyte + 2 Kbyte × 2)
  - 12 Kbyte × 8 bits (8 Kbyte + 2 Kbyte × 2)
  - 20 Kbyte × 8 bits (16 Kbyte + 2 Kbyte × 2)
- Two-bank configuration, enabling simultaneous execution of an erase/write operation and a read operation
- Automatic program algorithm (Embedded Algorithm)
- Erase-suspend/erase-resume functions integrated
- Detecting the completion of writing/erasing using the data polling flag or the toggle bit
- Detecting the completion of writing/erasing by CPU interrupts
- Capable of erasing data in specific sectors (any combination of sectors)
- Compatible with JEDEC standard commands
- Erase/write cycle: 100000 times
- Flash read cycle time (minimum): 1 machine cycle

■ Writing and Erasing Flash Memory

- Writing data to and reading data from the same bank of the Flash memory cannot be executed simultaneously.
- To write data to or erase data from a bank in the Flash memory, execute either the program for writing/erasing stored in another bank, or copy the program on the Flash memory to the RAM first and then execute it.
- The dual operation flash memory enables program execution in the Flash memory and write control using interrupts.
  In addition, it is not necessary to download a program to RAM in order to write data to a bank, thereby reducing the time of program download and eliminating the need to protecting RAM data against power interruption.

28.2 Sector/Bank Configuration of Flash Memory

This section shows the sector/bank configuration of the Flash memory.

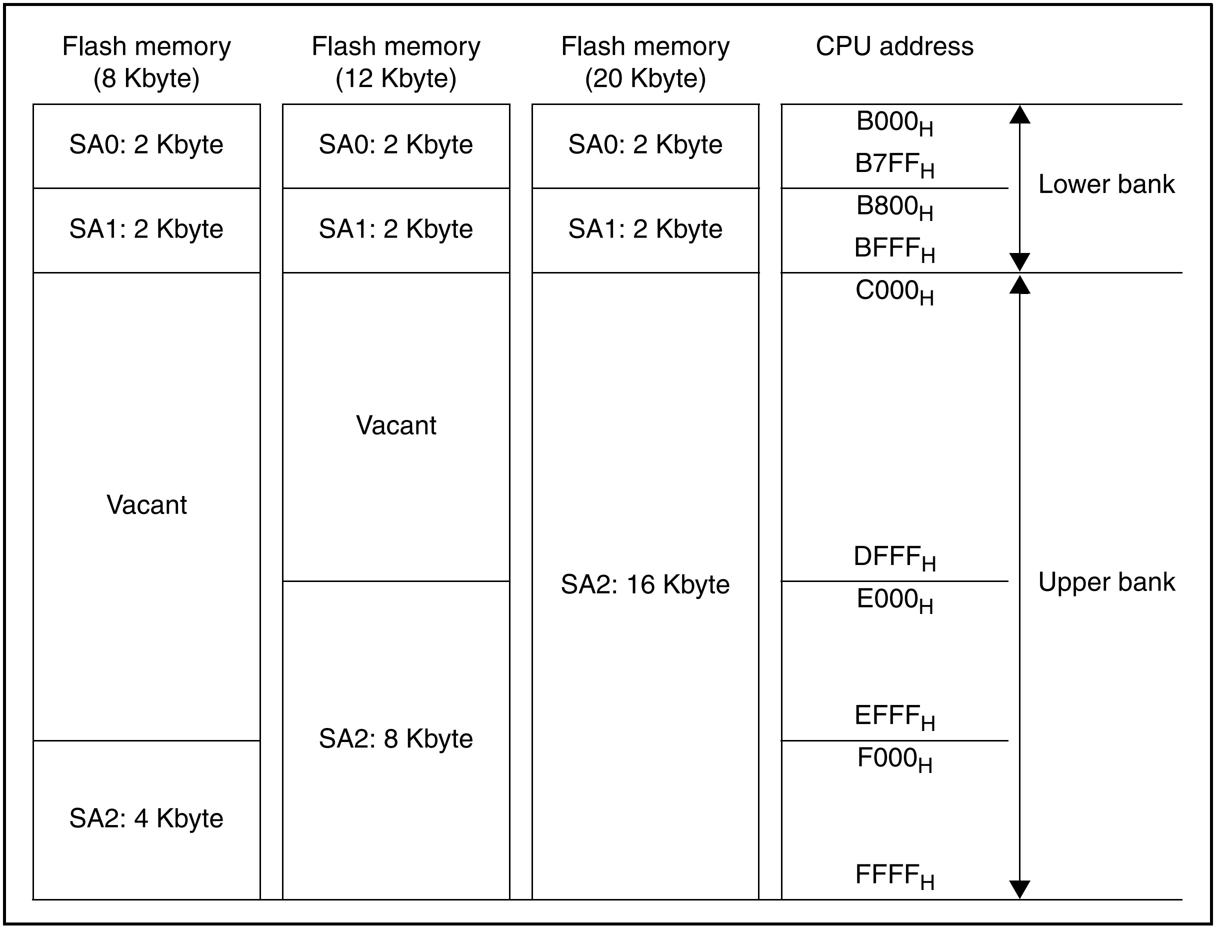
■ Sector/Bank Configuration of Dual Operation Flash Memory

Figure 28.2-1 shows the sector configuration of the Dual operation Flash memory. The upper and lower addresses of each sector are shown in the figure.

● Bank configuration

The lower bank of the Flash memory is SA0 and SA1 and the upper bank SA2.

Figure 28.2-1 Sector/Bank Configuration of Dual Operation Flash Memory



28.3 Registers for Flash Memory

This section shows the registers for the Flash memory.

■ Registers for Flash Memory

Figure 28.3-1 Registers for Flash Memory

Flash memory status register 2 (FSR2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0071_H	PEIEN	PGMEND	PTIEN	PGMTO	EEIEN	ERSEND	ETIEN	ERSTO	00000000_B
	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	

Flash memory status register (FSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072_H	-	-	RDYIRQ	RDY	Reserved	IRQEN	WRE	SSEN	$000X0000_B$
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W	

Flash memory sector write control register 0 (SWRE0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0073_H	Reserved	Reserved	Reserved	Reserved	Reserved	SA2E	SA1E	SA0E	00000000_B
	R/W0	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	

Flash memory status register 3 (FSR3)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0074_H	-	-	-	-	ESPS	SERS	PGMS	HANG	$0000XXXX_B$
	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	

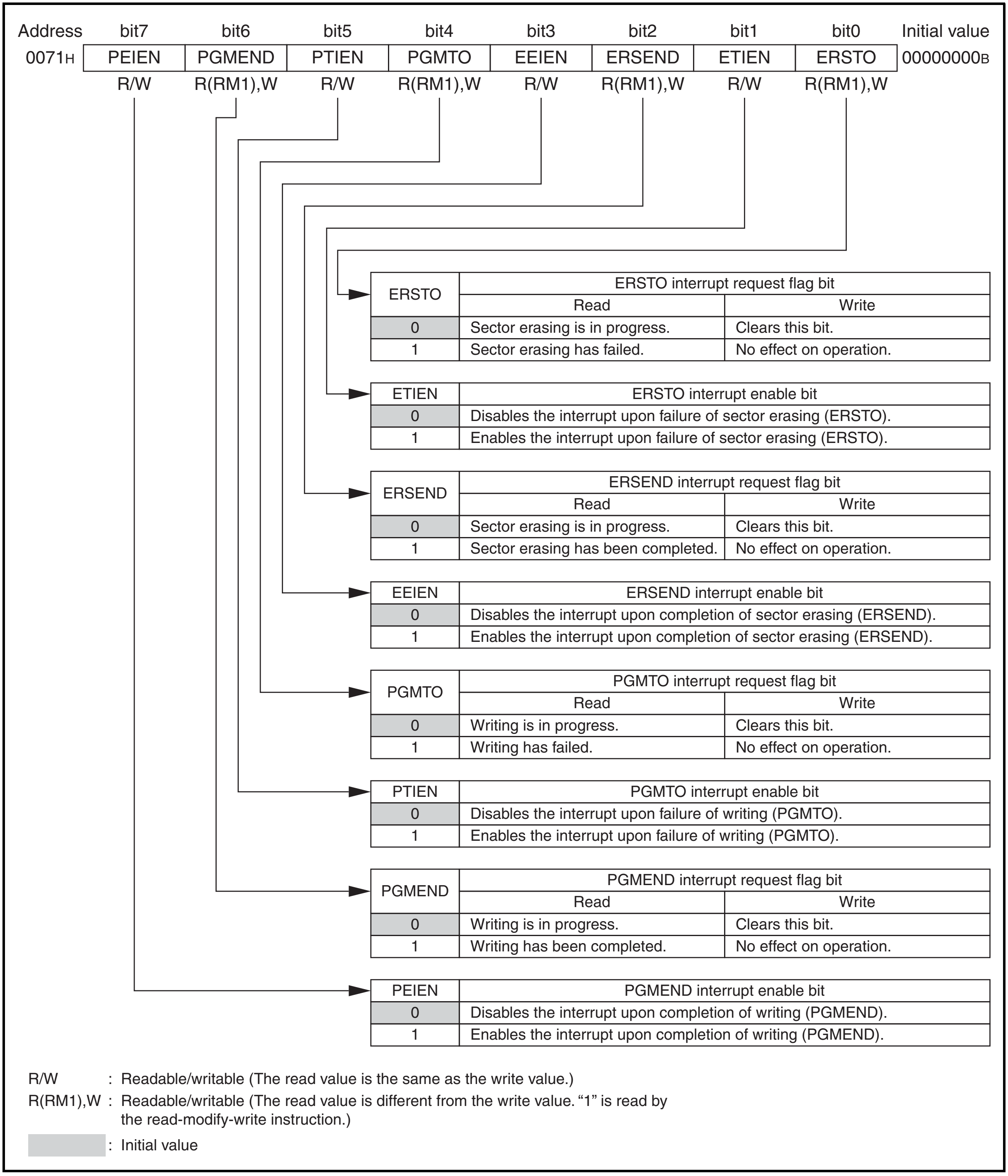
R/W	: Readable/writable (The read value is the same as the write value.)
R(RM1),W	: Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
R/WX	: Read only (Readable. Writing a value to it has no effect on operation.)
R/W0	: The write value is "0"; the read value is the same as the write value.
R0/WX	: The read value is "0". Writing a value to it has no effect on operation.)
-	: Undefined bit
X	: Indeterminate

28.3.1 Flash Memory Status Register 2 (FSR2)

Figure 28.3-2 shows the bit configuration of the flash memory status register 2 (FSR2).

■ Flash Memory Status Register 2 (FSR2)

Figure 28.3-2 Flash Memory Status Register 2 (FSR2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0071H	PEIEN	PGMEND	PTIEN	PGMTO	EEIEN	ERSEND	ETIEN	ERSTO	00000000B
	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	

ERSTO	ERSTO interrupt request flag bit	
	Read	Write
0	Sector erasing is in progress.	Clears this bit.
1	Sector erasing has failed.	No effect on operation.

ETIEN	ERSTO interrupt enable bit
0	Disables the interrupt upon failure of sector erasing (ERSTO).
1	Enables the interrupt upon failure of sector erasing (ERSTO).

ERSEND	ERSEND interrupt request flag bit	
	Read	Write
0	Sector erasing is in progress.	Clears this bit.
1	Sector erasing has been completed.	No effect on operation.

EEIEN	ERSEND interrupt enable bit
0	Disables the interrupt upon completion of sector erasing (ERSEND).
1	Enables the interrupt upon completion of sector erasing (ERSEND).

PGMTO	PGMTO interrupt request flag bit	
	Read	Write
0	Writing is in progress.	Clears this bit.
1	Writing has failed.	No effect on operation.

PTIEN	PGMTO interrupt enable bit
0	Disables the interrupt upon failure of writing (PGMTO).
1	Enables the interrupt upon failure of writing (PGMTO).

PGMEND	PGMEND interrupt request flag bit	
	Read	Write
0	Writing is in progress.	Clears this bit.
1	Writing has been completed.	No effect on operation.

PEIEN	PGMEND interrupt enable bit
0	Disables the interrupt upon completion of writing (PGMEND).
1	Enables the interrupt upon completion of writing (PGMEND).

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 instruction.)
▨ : Initial value

Table 28.3-1 Functions of Bits in Flash Memory Status Register 2 (FSR2) (1 / 2)

Bit name		Function
bit7	PEIEN: PGMEND interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the completion of Flash memory writing. **Writing "0":** Prevents an interrupt request from occurring even when Flash memory writing is completed (FSR2:PGMEND = 1). **Writing "1":** Causes an interrupt request to occur when Flash memory writing is completed (FSR2:PGMEND = 1).
bit6	PGMEND: PGMEND interrupt request flag bit	This bit indicates the completion of Flash memory writing. The PGMEND bit is set to "1" upon completion of the Flash memory automatic algorithm when Flash memory writing is completed. • An interrupt request occurs when the PGMEND bit is set to "1", provided that generating an interrupt request upon completion of Flash memory writing has been enabled (FSR2:PEIEN = 1). • When the PGMEND bit is set to "0" after Flash memory writing is completed, further Flash memory writing is disabled. • When Flash memory writing fails (FSR3:HANG = 1), this bit is cleared to "0". **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".
bit5	PTIEN: PGMTO interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the failure of Flash memory writing. **Writing "0":** Prevents an interrupt request from occurring even when Flash memory writing fails (FSR2:PGMTO = 1). **Writing "1":** Causes an interrupt request to occur when Flash memory writing fails (FSR2:PGMTO = 1).
bit4	PGMTO: PGMTO interrupt request flag bit	This bit indicates that Flash memory writing has failed. The PGMTO bit is set to "1" upon failure of the Flash memory automatic algorithm when Flash memory writing fails. • An interrupt request occurs when the PGMTO bit is set to "1", provided that generating an interrupt request upon failure of Flash memory writing has been enabled (FSR2:PTIEN = 1). • When the PGMTO bit is set to "1" after Flash memory writing is completed, further Flash memory writing is disabled. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. • When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".
bit3	EEIEN: ERSEND interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the completion of Flash memory sector erasing. **Writing "0":** Prevents an interrupt request from occurring even when Flash memory sector erasing is completed (FSR2:ERSEND = 1). **Writing "1":** Causes an interrupt request to occur when Flash memory sector erasing is completed (FSR2:ERSEND = 1).
bit2	ERSEND: ERSEND interrupt request flag bit	This bit indicates the completion of Flash memory sector erasing. The ERSEND bit is set to "1" upon completion of the Flash memory automatic algorithm when Flash memory sector erasing is completed. • An interrupt request occurs when the ERSEND bit is set to "1", provided that generating an interrupt request upon completion of Flash memory sector erasing has been enabled (FSR2:EEIEN = 1). • When the ERSEND bit is set to "0" after Flash memory sector erasing is completed, further Flash memory sector erasing is disabled. • When Flash memory sector erasing fails (FSR3:HANG = 1), this bit is cleared to "0". **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".

Table 28.3-1 Functions of Bits in Flash Memory Status Register 2 (FSR2) (2 / 2)

Bit name		Function
bit1	ETIEN: ERSTO interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the failure of Flash memory sector erasing. **Writing "0":** Prevents an interrupt request from occurring even when Flash memory sector erasing fails (FSR2:ERSTO = 1). **Writing "1":** Causes an interrupt request to occur when Flash memory sector erasing fails (FSR2:ERSTO = 1).
bit0	ERSTO: ERSTO interrupt request flag bit	This bit indicates that Flash memory sector erasing has failed. The ERSTO bit is set to "1" upon failure of the Flash memory automatic algorithm when Flash memory sector erasing fails. • An interrupt request occurs when the ERSTO bit is set to "1", provided that generating an interrupt request upon failure of Flash memory sector erasing has been enabled (FSR2:ETIEN = 1). • When the ERSTO bit is set to "1" after Flash memory sector erasing is completed, further Flash memory sector erasing is disabled. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".

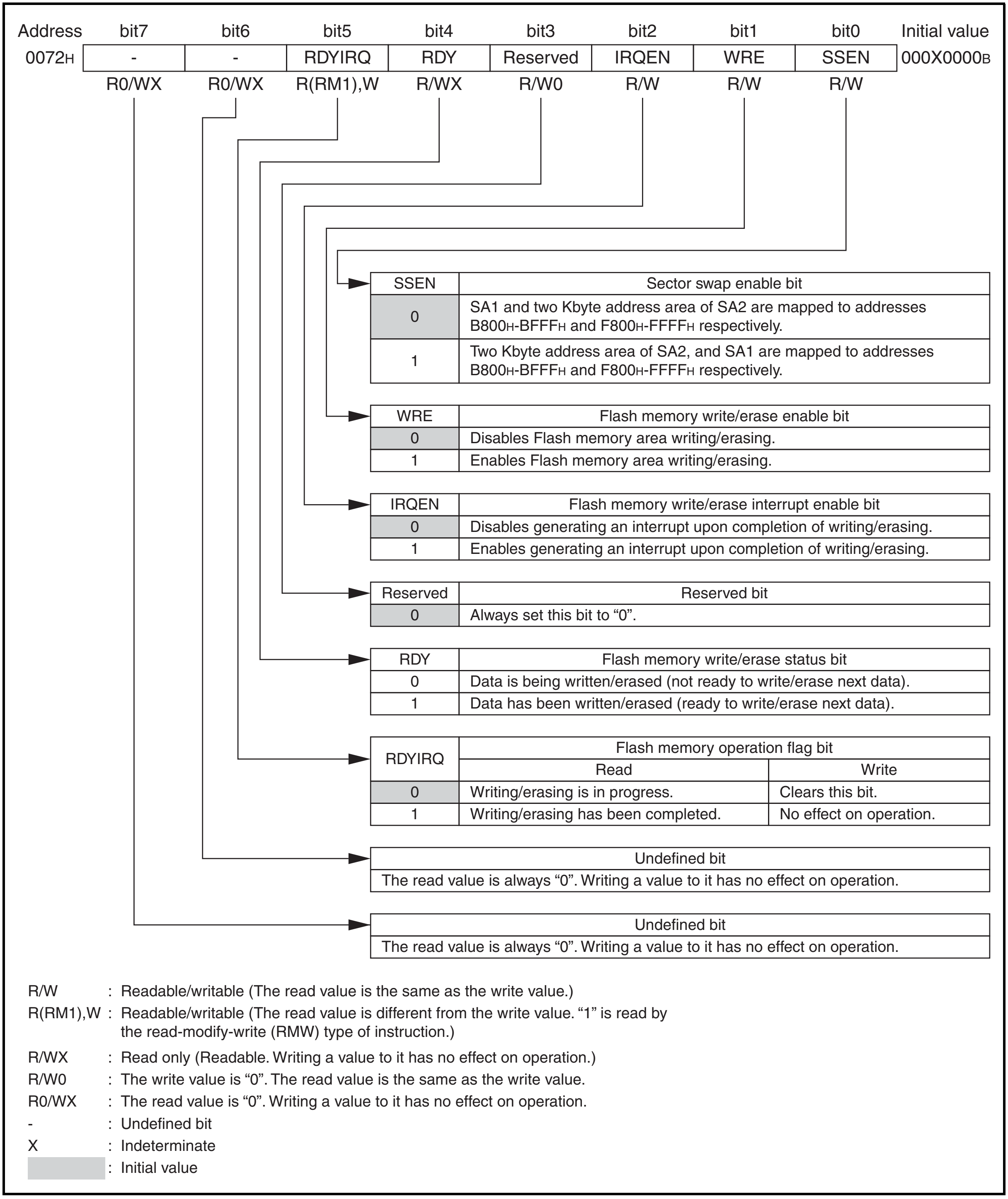
MB95430H Series

28.3.2 Flash Memory Status Register (FSR)

Figure 28.3-3 shows the bit configuration of the flash memory status register (FSR).

■ Flash Memory Status Register (FSR)

Figure 28.3-3 Flash Memory Status Register (FSR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072H	-	-	RDYIRQ	RDY	Reserved	IRQEN	WRE	SSEN	000X0000B
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W	

SSEN	Sector swap enable bit
0	SA1 and two Kbyte address area of SA2 are mapped to addresses B800H-BFFFH and F800H-FFFFH respectively.
1	Two Kbyte address area of SA2, and SA1 are mapped to addresses B800H-BFFFH and F800H-FFFFH respectively.

WRE	Flash memory write/erase enable bit
0	Disables Flash memory area writing/erasing.
1	Enables Flash memory area writing/erasing.

IRQEN	Flash memory write/erase interrupt enable bit
0	Disables generating an interrupt upon completion of writing/erasing.
1	Enables generating an interrupt upon completion of writing/erasing.

Reserved	Reserved bit
0	Always set this bit to “0”.

RDY	Flash memory write/erase status bit
0	Data is being written/erased (not ready to write/erase next data).
1	Data has been written/erased (ready to write/erase next data).

RDYIRQ	Flash memory operation flag bit	
	Read	Write
0	Writing/erasing is in progress.	Clears this bit.
1	Writing/erasing has been completed.	No effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

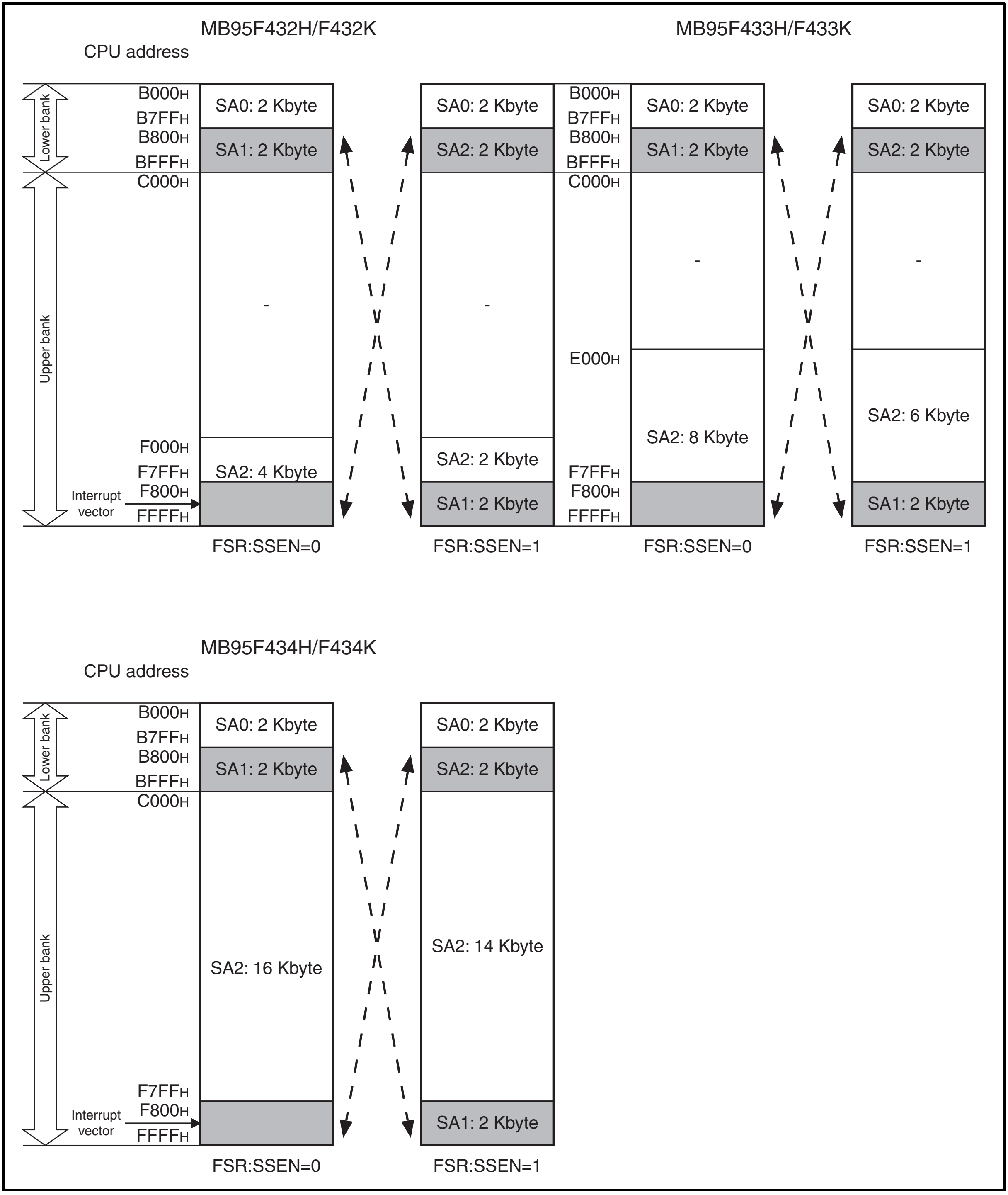
Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

R/W : Readable/writable (The read value is the same as the write value.)
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/W0 : The write value is “0”. The read value is the same as the write value.
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
X : Indeterminate
▨ : Initial value

Table 28.3-2 Functions of Bits in 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 writing/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 writing/erasing having been enabled (FSR:IRQEN = 1), if the RDYIRQ bit is set to "1", an interrupt request occurs. • After Flash memory writing/erasing is completed, if the RDYIRQ bit is set to "0", further Flash memory writing/erasing is disabled. **Writing "0":** Clears this bit. **Writing "1":** Has no effect on operation. When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".
bit4	RDY: Flash memory write/ erase status bit	This bit shows the write/erase status of the Flash memory. • When the RDY bit is "0", writing 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 writing or erasing ends, the RDY bit is set to "1". • After a write/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 write/erase command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit.
bit3	Reserved bit	Always set this bit to "0".
bit2	IRQEN: Flash memory write/ erase interrupt enable bit	This bit enables or disables the generation of interrupt requests triggered by the completion of Flash memory writing/erasing. **Writing "0":** Prevents an interrupt request from occurring even when the flash memory operation flag bit (FSR:RDYIRQ) is set to "1". **Writing "1":** Causes an interrupt request to occur when the flash memory operation flag bit (FSR:RDYIRQ) is set to "1".
bit1	WRE: Flash memory write/ erase enable bit	This bit enables or disables the writing/erasing of data into/from the Flash memory area. Set the WRE bit before invoking a Flash memory write/erase command. **Writing "0":** Prevents write/erase signals from being generated even when a write/erase command is input. **Writing "1":** Enables Flash memory writing/erasing to be executed after a write/erase command is input. • When not writing data into or erasing data from the Flash memory, set the WRE bit to "0" in order to prevent data from being accidentally written into or erased from the Flash memory. • To write data to the Flash memory, set FSR:WRE to "1" to enable writing data to the Flash memory, and set the flash memory sector write control register 0 (SWRE0) according to the Flash memory sector into which data is to be written. When Flash memory writing is disabled (FSR:WRE = 0), no write access to a sector in the Flash memory can be executed even though it has been enabled by setting a bit corresponding to that sector in the flash memory sector write control register 0 (SWRE0) to "1".
bit0	SSEN: Sector swap enable bit	This bit is used to swap the two Kbyte address area of SA2 in the upper bank, which contains an interrupt vector, for SA1 in the lower bank in dual operation mode. **Writing "0":** Maps SA1 to $B800_H$-$BFFF_H$, and the 2 Kbyte address area of SA2 to $F800_H$ to $FFFF_H$. **Writing "1":** Maps the 2 Kbyte address area of SA2 to $B800_H$-$BFFF_H$, and SA1 to $F800_H$ to $FFFF_H$.

Figure 28.3-4 Access Sector Map by FSR:SSEN Value



28.3.3 Flash Memory Sector Write Control Register 0 (SWRE0)

The flash memory sector write control register 0 (SWRE0) is installed in the Flash memory interface for implementing the Flash memory write-protect function.

■ Flash Memory Sector Write Control Register 0 (SWRE0)

The flash memory sector write control register 0 (SWRE0) has bits for enabling/disabling writing data into individual sectors (SA0 to SA2). The initial value of each bit is "0", meaning writing data is disabled. Writing "1" to a bit in SWRE0 enables writing data into the sector corresponding to that bit. Writing "0" to a bit in SWRE0 prevents data from being accidentally written into the sector corresponding to that bit. When "0" is written to a bit in SWRE0, even though "1" is written to that bit afterward, data cannot be written into the sector corresponding to that bit. To enable writing data into that sector, it is necessary to reset the bit corresponding to that sector first.

Figure 28.3-5 Flash Memory Sector Write Control Register 0 (SWRE0)

SWRE0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	Reserved	Reserved	Reserved	Reserved	Reserved	SA2E	SA1E	SA0E	00000000_B
0073_H	R/W0	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)
R/W0 : The write value is "0". The read value is the same as the write value.

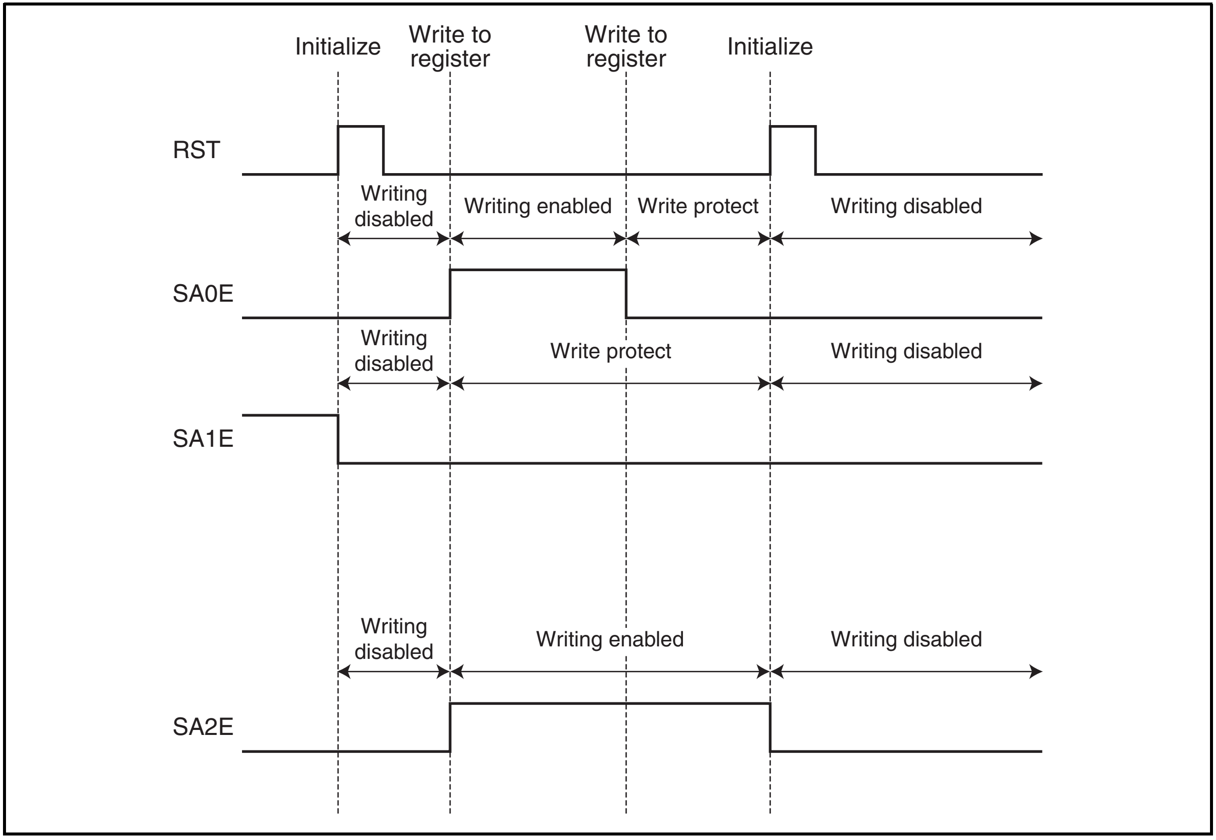
Only write data to SWRE0 by the byte. Setting the bits in SWRE0 using a bit manipulation instruction is prohibited.

Table 28.3-3 Functions of Bits in Flash Memory Sector Write Control Register 0 (SWRE0)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7
to
bit3</td><td>Reserved bits</td><td>Always set these bits to "0".</td></tr>
<tr><td>bit2
to
bit0</td><td>SA2E to SA0E:
Writing function setup bits</td><td>These bits are used to set the function of preventing data from being accidentally written into a sector of the Flash memory. Writing "1" to a bit in SWRE0 enables writing data into the sector corresponding to that bit. Writing "0" to a bit in SWRE0 prevents data from being accidentally written into the sector corresponding to that bit. In addition, resetting this bit initializes it to "0" (writing disabled).

Table of writing function setup bits and their corresponding Flash memory sectors
<table>
<tr><th>Bit Name</th><th>Corresponding Sector in Flash Memory</th></tr>
<tr><td>SA2E</td><td>SA2</td></tr>
<tr><td>SA1E</td><td>SA1</td></tr>
<tr><td>SA0E</td><td>SA0</td></tr>
</table>
<b>Writing disabled</b> : SAxE is "0". With no "0" written to the SAxE bit in the flash memory sector write control register 0 (SWRE0), writing data into a sector can be enabled by setting the SAxE bit corresponding to that sector to "1". (This is the state after SAxE is reset).
<b>Writing enabled</b> : SAxE is "1". Data can be written into a sector corresponding to the SAxE bit.
<b>Write protect</b> : SAxE is "0 ". With "0" written to the SAxE bit in the flash memory sector write control register 0 (SWRE0), writing data into a sector cannot be enabled even though the SAxE bit corresponding to that sector is set to "1".</td></tr>
</table>

Figure 28.3-6 Examples of Flash Memory Writing-disabled, Writing-enabled, and Write-protected States Depending on Flash Memory Sector Write Control Register 0 (SWRE0)



Writing disabled:

SAxE is "0". With no "0" written to the SAxE bit in the flash memory sector write control register 0 (SWRE0), writing data into a sector can be enabled by setting the SAxE bit corresponding to that sector to "1". (This is the state after SAxE is reset).

Writing enabled:

SAxE is "1". Data can be written to a sector corresponding to the SAxE bit.

Write protect:

SAxE is "0 ". With "0" written to the SAxE bit in the flash memory sector write control register 0 (SWRE0), writing data to a sector cannot be enabled even though the SAxE bit corresponding to that sector is set to "1".

■ Note on Setting SWRE0 Register

To write data to or erase data from SA0 ($B000_H$-$B7FF_H$) or SA1 ($B800_H$-$BFFF_H$) of the Flash memory when FSR:SSEN is "0", set both SA0E and SA1E in the SWRE0 register to "1" first.

To write data to or erase data when FSR:SSEN is "1", set SA0E, SA1E and SA2E in the SWRE0 register to "1" first.

For details of the sector map of the Flash memory, see "Figure 28.3-4 Access Sector Map by FSR:SSEN Value".

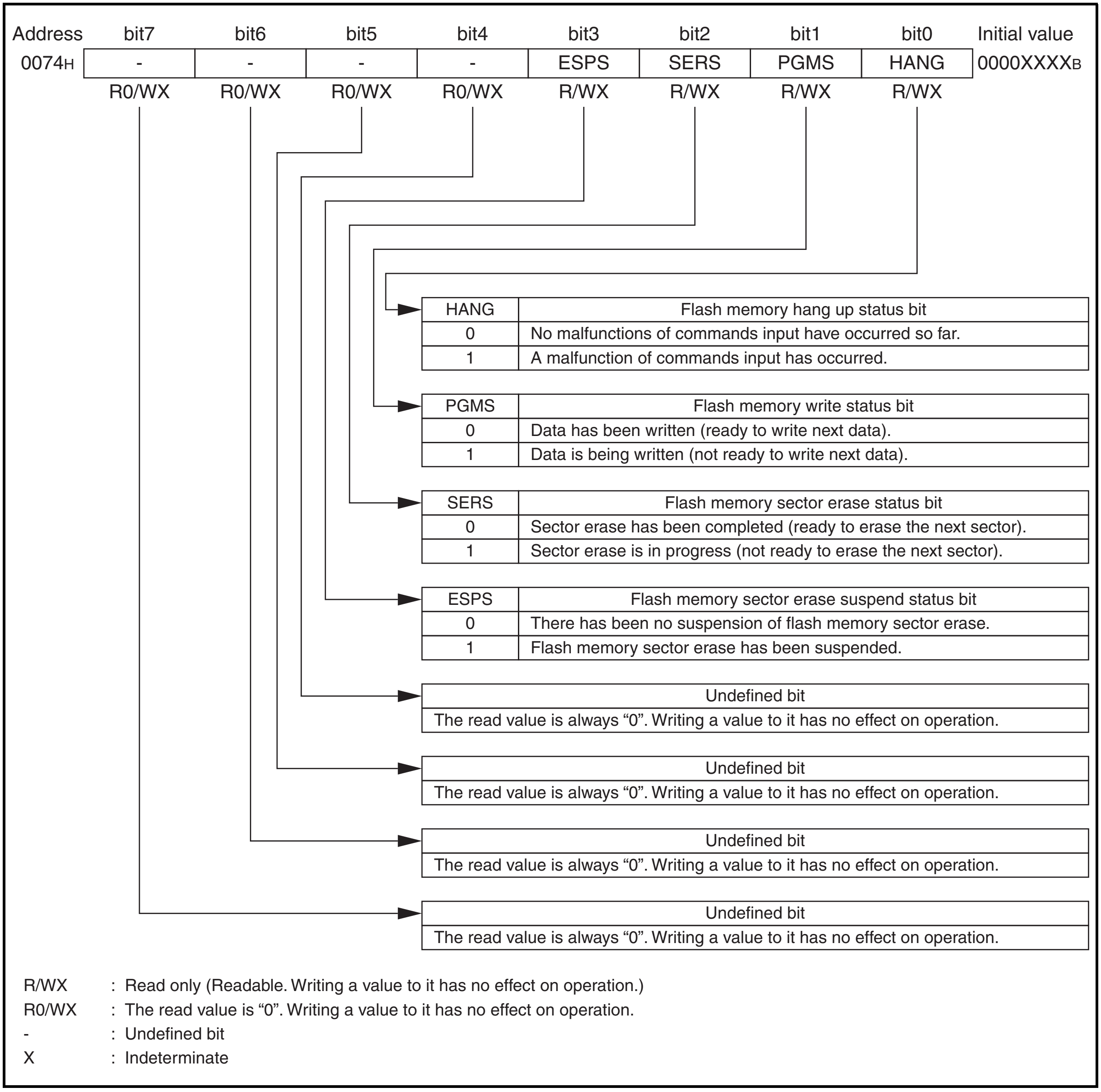
MB95430H Series

28.3.4 Flash Memory Status Register 3 (FSR3)

Figure 28.3-7 shows the bit configuration of the flash memory status register 3 (FSR3).

■ Flash Memory Status Register 3 (FSR3)

Figure 28.3-7 Flash Memory Status Register 3 (FSR3)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0074H	-	-	-	-	ESPS	SERS	PGMS	HANG	0000XXXXB
	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	

HANG	Flash memory hang up status bit
0	No malfunctions of commands input have occurred so far.
1	A malfunction of commands input has occurred.

PGMS	Flash memory write status bit
0	Data has been written (ready to write next data).
1	Data is being written (not ready to write next data).

SERS	Flash memory sector erase status bit
0	Sector erase has been completed (ready to erase the next sector).
1	Sector erase is in progress (not ready to erase the next sector).

ESPS	Flash memory sector erase suspend status bit
0	There has been no suspension of flash memory sector erase.
1	Flash memory sector erase has been suspended.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

Undefined bit
The read value is always “0”. Writing a value to it has no effect on operation.

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
X : Indeterminate

Table 28.3-4 Functions of Bits in Flash Memory Status Register 3 (FSR3)

Bit name		Function
bit7 to bit4	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit3	ESPS: Flash memory sector erase suspend status bit	This bit shows the sector erase suspend status of the Flash memory. • When the ESPS bit is set to "1", that indicates Flash memory sector erase has been suspended. • When the ESPS bit is set to "0", that indicates there has been no suspension of Flash memory sector erase. • There is a delay of two machine clock (MCLK) cycles between the issuance of a sector erase suspend command and the ESPS bit being set to "1". After issuing a sector erase suspend command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit.
bit2	SERS: Flash memory sector erase status bit	This bit shows the sector erase status of the Flash memory. • When the SERS bit is set to "1", that indicates sector erase is in progress. • When the SERS bit is set to "0", that indicates sector erase has been completed. • There is a delay of two machine clock (MCLK) cycles between the issuance of a sector erase command and the SERS bit being set to "1". After issuing a sector erase suspend command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit.
bit1	PGMS: Flash memory write status bit	This bit shows the writing status of the Flash memory. • When the PGMS bit is set to "1", that indicates data is being written to the Flash memory. • When the PGMS bit is set to "0", that indicates data has been written to the Flash memory. • There is a delay of two machine clock (MCLK) cycles between the issuance of a write command and the PGMS bit being set to "1". After issuing a sector erase suspend command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit. • The PGMS bit will never be asserted under the condition that the machine clock (MCLK) cycle is longer than 1µs. Use this bit with the machine clock (MCLK) cycle shorter than 1 µs.
bit0	HANG: Flash memory hang up status bit	This bit shows whether the Flash memory has malfunctioned or not. • When the HANG bit is set to "1", that indicates a malfunction of commands input has occurred. • When the HANG bit is set to "0", that indicates no malfunctions of commands input have occurred so far. • There is a delay of two machine clock (MCLK) cycles between the issuance of a reset command and the HANG bit being cleared to "0". After issuing a sector erase suspend command, wait for those two machine clock cycles to elapse (e.g. inserting NOP twice) before reading this bit.

■ Examples of Status of Flash Memory Status Register 2, Flash Memory Status Register 3 and RDY Bit (FSR:bit4)

Figure 28.3-8 FSR2:PGMEND during Flash Memory Write

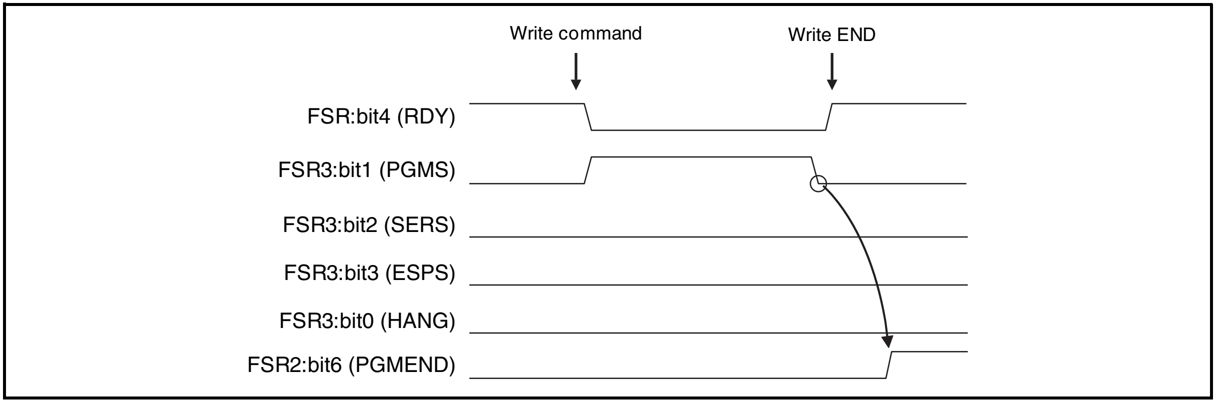


Figure 28.3-9 FSR2:PGMTO when Flash Memory Write Failed

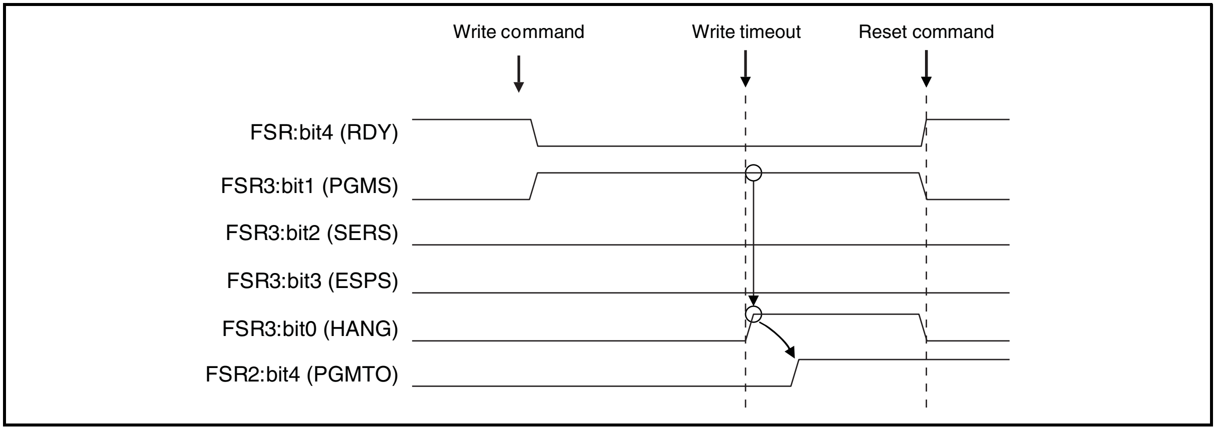


Figure 28.3-10 FSR2:ERSEND during Flash Memory Sector Erase

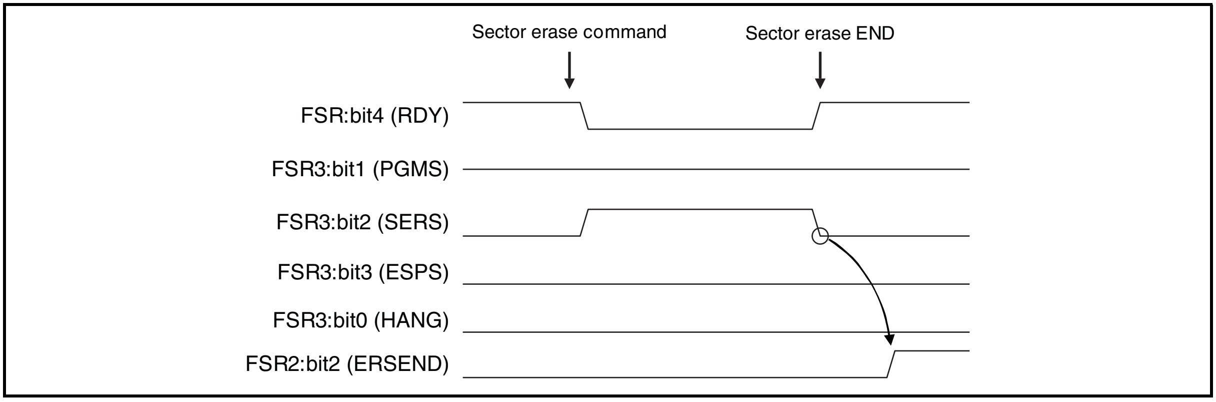


Figure 28.3-11 FSR2:ERSTO when Flash Memory Sector Erase Failed

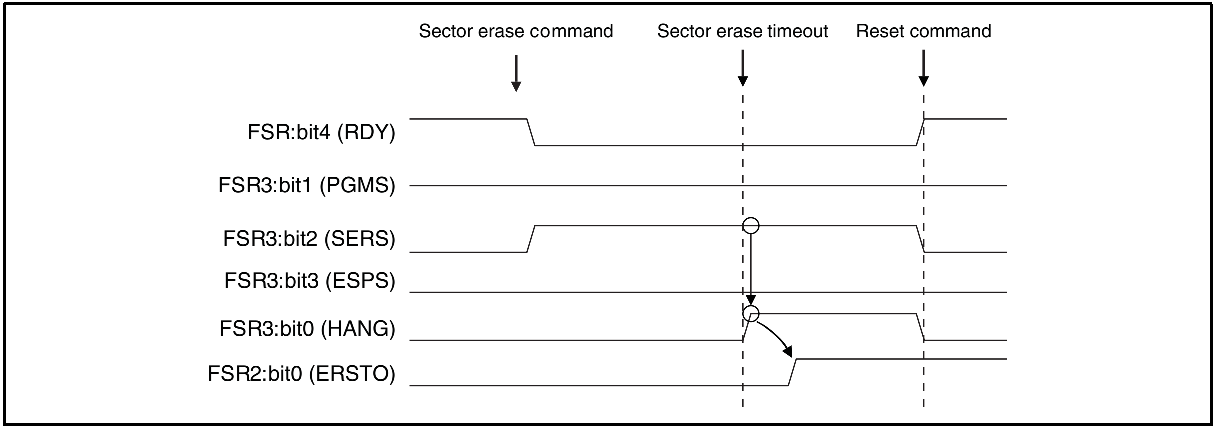


Figure 28.3-12 FSR2:PGMEND and FSR2:ERSEND when Flash Memory Write Is in Progress with Flash Memory Sector Erase Suspended

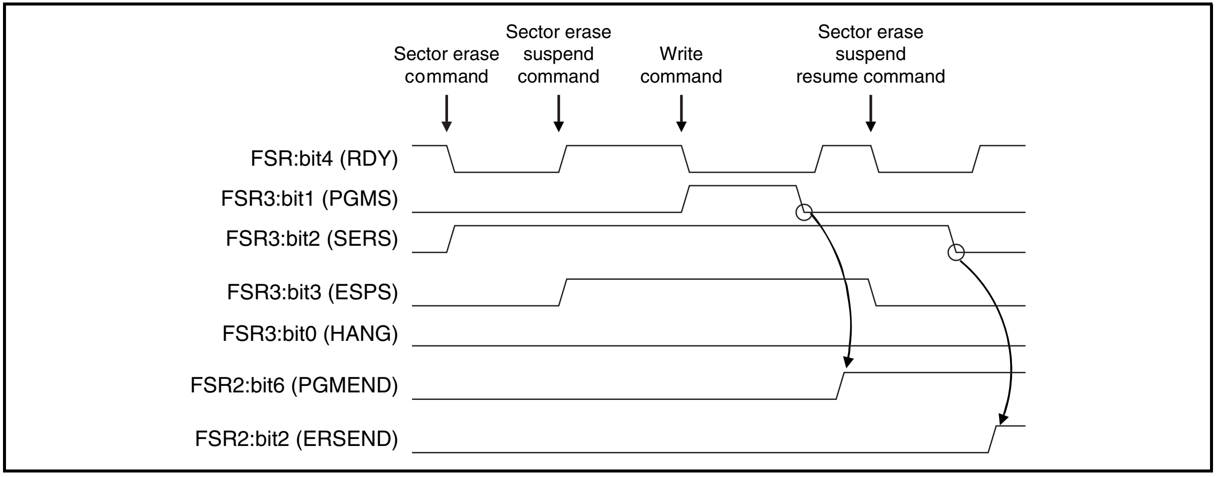


Figure 28.3-13 FSR2:PGMTO and FSR2:ERSEND when Flash Memory Write Failed with Flash Memory Sector Erase Suspended

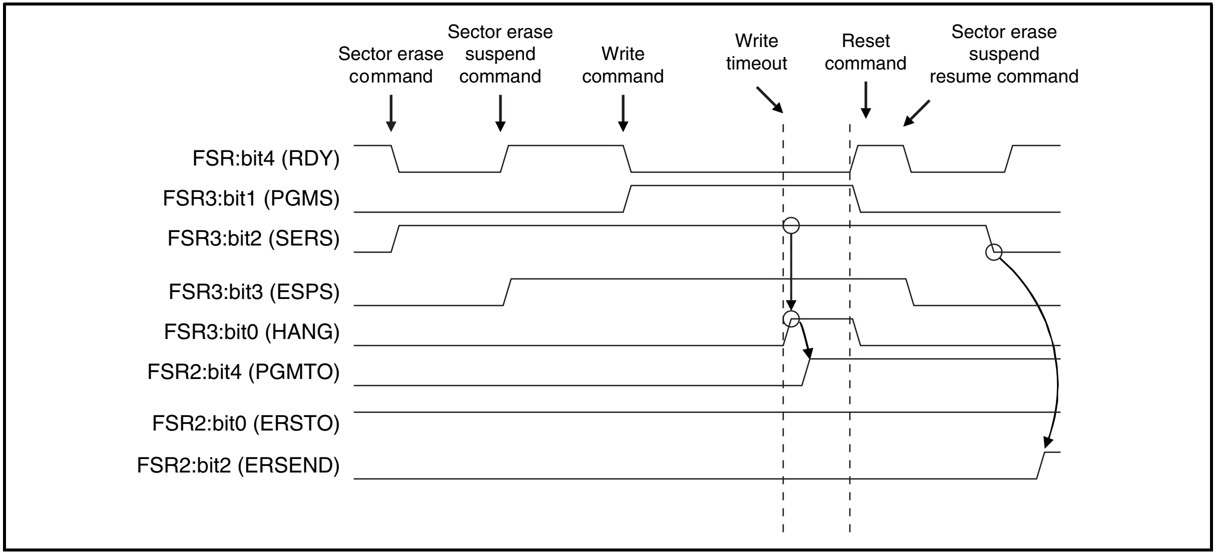


Figure 28.3-14 FSR2:ERSEND when Flash Memory Read Is in Progress with Flash Memory Sector Erase Suspended

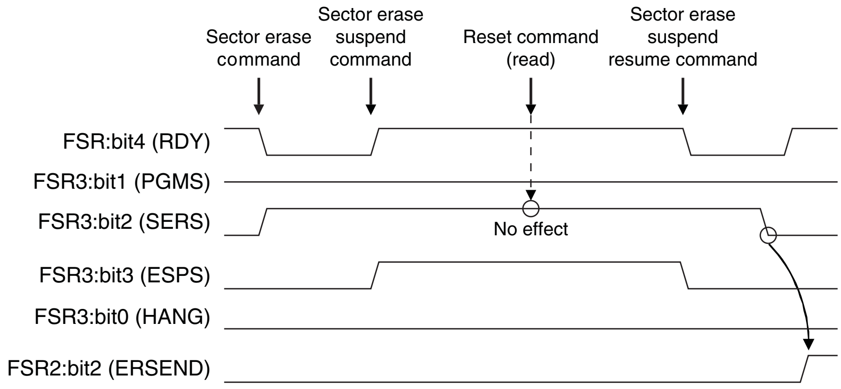
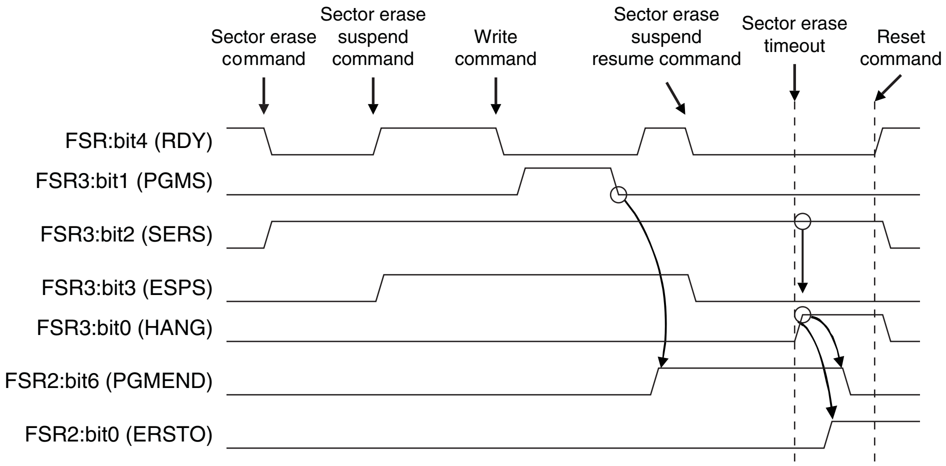


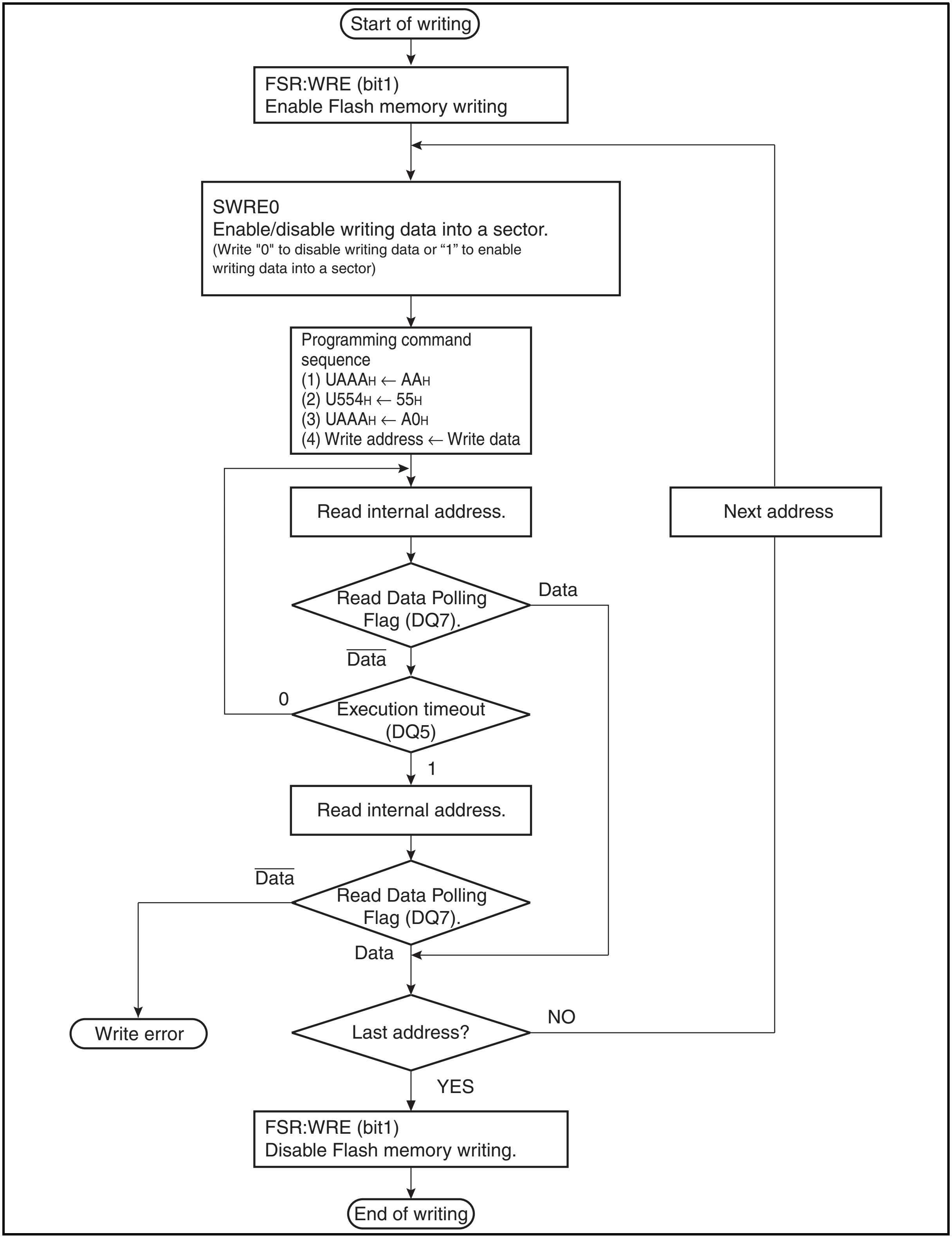
Figure 28.3-15 FSR2:PGMEND and FSR2:ERSTO when Flash Memory Sector Erase Failed after Sector Erase Has Resumed



■ Flash Memory Sector Write Control Register 0 (SWRE0) Setup Flow Chart

Set the FSR:WRE bit to "1" to enable Flash memory writing, then enable or disable writing data into a sector by setting the corresponding bit in the SWRE0 register to "1" or "0" respectively.

Figure 28.3-16 Sample Procedure for Enabling/Disabling Flash Memory Writing


Start of writing
FSR:WRE (bit1)
Enable Flash memory writing
SWRE0
Enable/disable writing data into a sector.
(Write "0" to disable writing data or "1" to enable writing data into a sector)
Programming command sequence
(1) UAAAH ← AAH
(2) U554H ← 55H
(3) UAAAH ← A0H
(4) Write address ← Write data
Read internal address.
Next address
Read Data Polling Flag (DQ7).
Data
$\overline{\text{Data}}$
0
Execution timeout (DQ5)
1
Read internal address.
$\overline{\text{Data}}$
Read Data Polling Flag (DQ7).
Data
Write error
Last address?
NO
YES
FSR:WRE (bit1)
Disable Flash memory writing.
End of writing

■ Note on Setting (FSR:WRE)

To write data to the Flash memory, set FSR:WRE to "1" to enable Flash memory writing and then set the bit in the SWRE0 register corresponding to a sector to which data is to be written. When Flash memory writing is disabled by setting FSR:WRE to "0", no write access to a sector in the Flash memory can be executed even though it has been enabled by setting a bit corresponding to that sector in the SWRE0 register to "1".

28.4 Invoking Flash Memory Automatic Algorithm

There are four commands that invoke the Flash memory automatic algorithm: read/reset, write, chip-erase, and sector-erase. The sector-erase command is capable of suspending and resuming sector erase.

■ Command Sequence Table

Table 28.4-1 lists commands used in writing/erasing Flash memory.

Table 28.4-1 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$_X$XX$_H$	F0$_H$	-	-	-	-	-	-	-	-	-	-
	4	UAAA$_H$	AA$_H$	U554$_H$	55$_H$	UAAA$_H$	F0$_H$	RA	RD	-	-	-	-
Write	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$_H$
Sector erase	6	UAAA$_H$	AA$_H$	X554$_H$	55$_H$	UAAA$_H$	80$_H$	UAAA$_H$	AA$_H$	U554$_H$	55$_H$	SA	30$_H$
Sector erase suspend			Writing data "B0$_H$" to address "UXXX$_H$" suspends erasing during sector erasing.										
Sector erase resume			Writing data "30$_H$" to address "UXXX$_H$" resumes suspended sector erasing.										

RA	: Read address
PA	: Write address
SA	: Sector address (specify arbitrary one address in sector)
RD	: Read data
PD	: Write data
U	: Upper 4 bits same as RA, PA, and SA
F$_X$	: FF/FE
X	: Arbitrary address

*: Both types of read/reset command sequence can reset the Flash memory to read mode.

Note:

- 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 in RA, PA and SA.

  Example: If RA = C48E$_H$, U = C; if PA = 1024$_H$, U=1
  If SA = 3000$_H$, U = 3
- The chip erase command is accepted only when writing data into all sectors has been enabled. The chip erase command is ignored if the bit for any sector in the flash memory sector write control register 0 (SWRE0) has been set to "0" (to disable writing data to that sector).

■ Note on Issuing Commands

Pay attention to the following two points when issuing commands in command sequence table:

- Enable writing data into a required sector before issuing the first command.
- Ensure that since the first command, the value "U", which represents the upper four bits (bit 15 to bit 12) of an address, the same as the upper four bits in RA, PA and SA.

If the two points above are not observed, commands cannot be recognized properly. When commands are not recognized properly, execute a reset to initialize the command sequencer in the Flash memory.

28.5 Checking Automatic Algorithm Execution Status

Since the Flash memory uses the automatic algorithm to execute the write/erase flow, its internal operating status can be checked through the hardware sequence flags.

■ Hardware Sequence Flags

● Overview of hardware sequence flags

The hardware sequence flag consists of the following 4-bit output:

- Data polling flag (DQ7)
- Toggle bit flag (DQ6)
- Execution timeout flag (DQ5)
- Sector erase timer flag (DQ3)

The hardware sequence flags can tell whether a write command, a chip-erase command or a sector-erase command has been terminated and whether an erase code can be written.

The value of a hardware sequence flag can be checked by a read access to the address of a target sector in the Flash memory after a command sequence is set. Note that a hardware sequence flag is output only to the bank from which a command has been issued.

Table 28.5-1 shows the bit allocation of the hardware sequence flags.

Table 28.5-1 Bit Allocation of Hardware Sequence Flag

Bit no.	7	6	5	4	3	2	1	0
Hardware sequence flag	DQ7	DQ6	DQ5	-	DQ3	-	-	-

- To decide whether an automatic write command, a chip-erase command or a sector-erase command is being executed or has been terminated, check the respective hardware sequence flags or the flash memory write/erase status bit in the flash memory status register (FSR:RDY). After writing/erasing is terminated, the Flash memory returns to the read/reset state.
- When creating a write/erase program, read data after confirming the termination of automatic writing/erasing using the DQ3, DQ5, DQ6 and DQ7 flags.
- The hardware sequence flags can also be used to check whether the second sector erase code write and those to be executed afterward are valid or not.

- Description of hardware sequence flags

Table 28.5-2 lists the functions of the hardware sequence flags.

Table 28.5-2 List of Hardware Sequence Flag Functions

State		DQ7	DQ6	DQ5	DQ3
State transition during normal operation	Writing → Writing completed (when write address has been specified)	$\overline{DQ7}$ → DATA: 7	Toggle → DATA: 6	0 → DATA: 5	0 → DATA: 3
	Chip/sector erasing → Erasing completed	0 → 1	Toggle → Stop	0 → 1	1
	Sector erasing wait → Erasing started	0	Toggle	0	0 → 1
	Erasing → Sector erasing suspended (Sector being erased)	0 → 1	Toggle → 1	0	1 → 0
	Sector erasing suspended → Erasing resumed (Sector being erased)	1 → 0	1 → Toggle	0	0 → 1
	Sector erasing being suspended (Sector not being erased)	DATA: 7	DATA: 6	DATA: 5	DATA: 3
Abnormal operation	Writing	$\overline{DQ7}$	Toggle	1	0
	Chip/sector erasing	0	Toggle	1	1

28.5.1 Data Polling Flag (DQ7)

The data polling flag (DQ7) is a hardware sequence flag used to indicate that the automatic algorithm is being executing or has been completed using the data polling function.

■ Data Polling Flag (DQ7)

Table 28.5-3 and Table 28.5-4 show the state transition of the data polling flag during normal operation and the one during abnormal operation respectively.

Table 28.5-3 State Transition of Data Polling Flag (During Normal Operation)

Operating state	Writing → Writing completed	Chip/sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ7	$\overline{\text{DQ7}}$ → DATA: 7	0 → 1	0	0 → 1	1 → 0	DATA: 7

Table 28.5-4 State Transition of Data Polling Flag (During Abnormal Operation)

Operating state	Writing	Chip/sector erasing
DQ7	$\overline{\text{DQ7}}$	0

- At writing

When read access takes place during execution of the automatic write algorithm, the Flash memory outputs the inverted value of bit7 in the last data written to DQ7.

If read access takes place on completion of the automatic write algorithm, the Flash memory outputs bit7 of the value read from the read-accessed address to DQ7.

- At chip/sector erasing

When read access is made to the sector currently being erased during execution of the chip/sector erase algorithm, bit7 of Flash memory outputs "0". Bit7 of Flash memory outputs "1" upon completion of chip/sector erasing.

● At sector erasing suspension

- When read access takes place with a sector-erase operation suspended, the Flash memory outputs "1" to DQ7 if the read address is the sector being erased. If not, the Flash memory outputs bit7 (DATA:7) of the value read from the read address to DQ7.
- Referring the data polling flag (DQ7) together with the toggle bit flag (DQ6) permits a decision on whether Flash memory is in the sector erase suspended state or which sector is being erased.

Note:

Once the automatic algorithm has been started, read access to the specified address is ignored. Data reading is allowed after the data polling flag (DQ7) is set to "1". Data reading after the end of the automatic algorithm should be performed following read access made to confirm the completion of data polling.

28.5.2 Toggle Bit Flag (DQ6)

The toggle bit flag (DQ6) is a hardware sequence flag indicating whether the automatic algorithm is being executed or terminates using the toggle bit function.

■ Toggle Bit Flag (DQ6)

Table 28.5-5 and Table 28.5-6 show the state transition of the toggle bit flag during normal operation and the one during abnormal operation respectively.

Table 28.5-5 State Transition of Toggle Bit Flag (During Normal Operation)

Operating state	Writing → Writing completed	Chip/sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ6	Toggle → DATA: 6	Toggle → Stop	Toggle	Toggle → 1	1 → Toggle	DATA: 6

Table 28.5-6 State Transition of Toggle Bit Flag (During Abnormal Operation)

Operating state	Writing	Chip/sector erasing
DQ6	Toggle	Toggle

● At writing and chip/sector erasing

- When read accesses are made continuously while the automatic write algorithm or the chip-erase/sector-erase algorithm is being executed, the Flash memory toggles the output between "1" and "0" at each read access.
- When read accesses are made continuously after the automatic write algorithm or the chip-erase/sector-erase algorithm terminates, the Flash memory outputs bit6 (DATA:6) of the value read from the read address at each read access.

● At sector erasing suspension

When a read access is made with a sector-erase operation suspended, the Flash memory outputs "1" if the read address is the sector being erased. Otherwise, the Flash memory outputs bit6 (DATA: 6) of the value read from the read address.

Note:

When using dual-operation Flash memory (flash memory write control program is executed on the flash memory), the toggle bit flag (DQ6) cannot be used to check the operating state of writing/erasing. See the notes in "28.9 Notes on Using Dual Operation Flash Memory" when writing a program.
The note above does not apply if the flash memory write control program is executed on the RAM.

28.5.3 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 writing/erasing) in the Flash memory.

■ Execution Timeout Flag (DQ5)

Table 28.5-7 and Table 28.5-8 show the state transition of the execution timeout flag during normal operation and the one during abnormal operation respectively.

Table 28.5-7 State Transition of Execution Timeout Flag (During Normal Operation)

Operating state	Writing → Writing completed	Chip/sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ5	0 → DATA: 5	0 → 1	0	0	0	DATA: 5

Table 28.5-8 State Transition of Execution Timeout Flag (During Abnormal Operation)

Operating state	Writing	Chip/sector erasing
DQ5	1	1

● At writing and chip/sector erasing

When a read access is made with the write or chip-erase/sector-erase automatic algorithm invoked, the flag outputs "0" when the algorithm execution time is within the specified time (required for writing/erasing) or "1" when it exceeds that time.

The execution time-out flag (DQ5) can be used to check whether writing/erasing has succeeded or failed regardless of whether the automatic algorithm has been running or terminated. When the execution timeout flag (DQ5) outputs "1", it can be judged that writing fails if flash memory write/erase status bit (RDY) of flash memory status register (FSR) is "0".

If an attempt is made to write "1" to a Flash memory address holding "0", for example, the Flash memory is locked, the time limit is exceeded and the execution time-out flag (DQ5) outputs "1". The state in which the execution time-out flag (DQ5) outputs "1" means that the Flash memory has not been used correctly; it does not mean that the Flash memory is defective. When this state occurs, execute the reset command.

28.5.4 Sector Erase Timer Flag (DQ3)

The sector erase timer flag (DQ3) is a hardware sequence flag indicating whether the Flash memory is waiting for sector erasing after the sector erase command has started.

■ Sector Erase Timer Flag (DQ3)

Table 28.5-9 and Table 28.5-10 show the state transition of the sector erase timer flag during normal operation and the one during abnormal operation respectively.

Table 28.5-9 State Transition of Sector Erase Timer Flag (During Normal Operation)

Operating state	Writing → Writing completed	Chip/sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ3	0 → DATA: 3	1	0 → 1	1 → 0	0 → 1	DATA: 3

Table 28.5-10State Transition of Sector Erase Timer Flag (During Abnormal Operation)

Operating state	Writing	Chip/sector erasing
DQ3	0	1

● At sector erasing

- When a read access is made after the sector erase command has started, the sector erase timer flag (DQ3) outputs "0" within the sector erase wait period. The flag outputs "1" if the sector erase wait period has elapsed.
- With the data polling function or the toggle bit function indicating that the erase algorithm is being executed (DQ7 = 0, DQ6 indicates toggle output), that the sector erase timer flag (DQ3) is "1" indicates that sector erasing is in progress. If any command other than the sector erase suspend command is set subsequently, it is ignored until sector erasing is terminated.
- If the sector erase timer flag (DQ3) is "0", the Flash memory can accept the sector erase command. Before writing the sector erase command to the Flash memory, make sure that the sector erase timer flag (DQ3) is "0". If the flag is "1", the Flash memory may not accept the sector erase command suspended.

● At sector erasing suspension

When a read access is made with the sector erase operation suspended, the Flash memory outputs "1" if the read address of that read access is the address of a sector being erased. If the read address is not the address of a sector being erased, the Flash memory outputs bit3 (DATA: 3) of the value read from the read address.

28.6 Writing/Erasing Flash Memory

This section describes the respective procedures for reading/resetting the Flash memory, writing, chip-erasing, sector-erasing, sector erase suspending and sector-erase resuming by entering respective commands to invoke the automatic algorithm.

■ Details of Writing/Erasing Flash Memory

The automatic algorithm can be invoked by writing the read/reset, write, chip-erase, sector-erase, sector-erase suspend, and sector-erase resume command sequence to the Flash memory from the CPU. Always write the commands of a command sequence continuously from the CPU to the Flash memory. The termination of the automatic algorithm can be checked by the data polling function. After the automatic algorithm terminates normally, the Flash memory returns to the read/reset state.

The operations are explained in the following order:

- Enter the read/reset state.
- Write data.
- Erase all data (chip-erase).
- Erase arbitrary data (sector-erase).
- Suspend sector erasing.
- Resume sector erasing.

28.6.1 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 the 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 write access, in the same way as the masked ROM.
- In the case of a read access to the Flash memory, no read/reset commands are required. If a command does not terminate normally, use a read/reset command to initialize the automatic algorithm.

28.6.2 Writing Data to Flash Memory

This section explains the procedure for entering the write command to write data to the Flash memory.

■ Writing Data to Flash Memory

- To invoke the automatic algorithm for writing data to the Flash memory, send write commands in the command sequence table consecutively from the CPU to the Flash memory.
- When writing data to a target address ends in the fourth cycle, the automatic algorithm is invoked and starts automatic writing.

● Addressing method

- Writing can be performed in any order of addresses and across a sector boundary. The size of data that can be written by a single write command is one byte only.

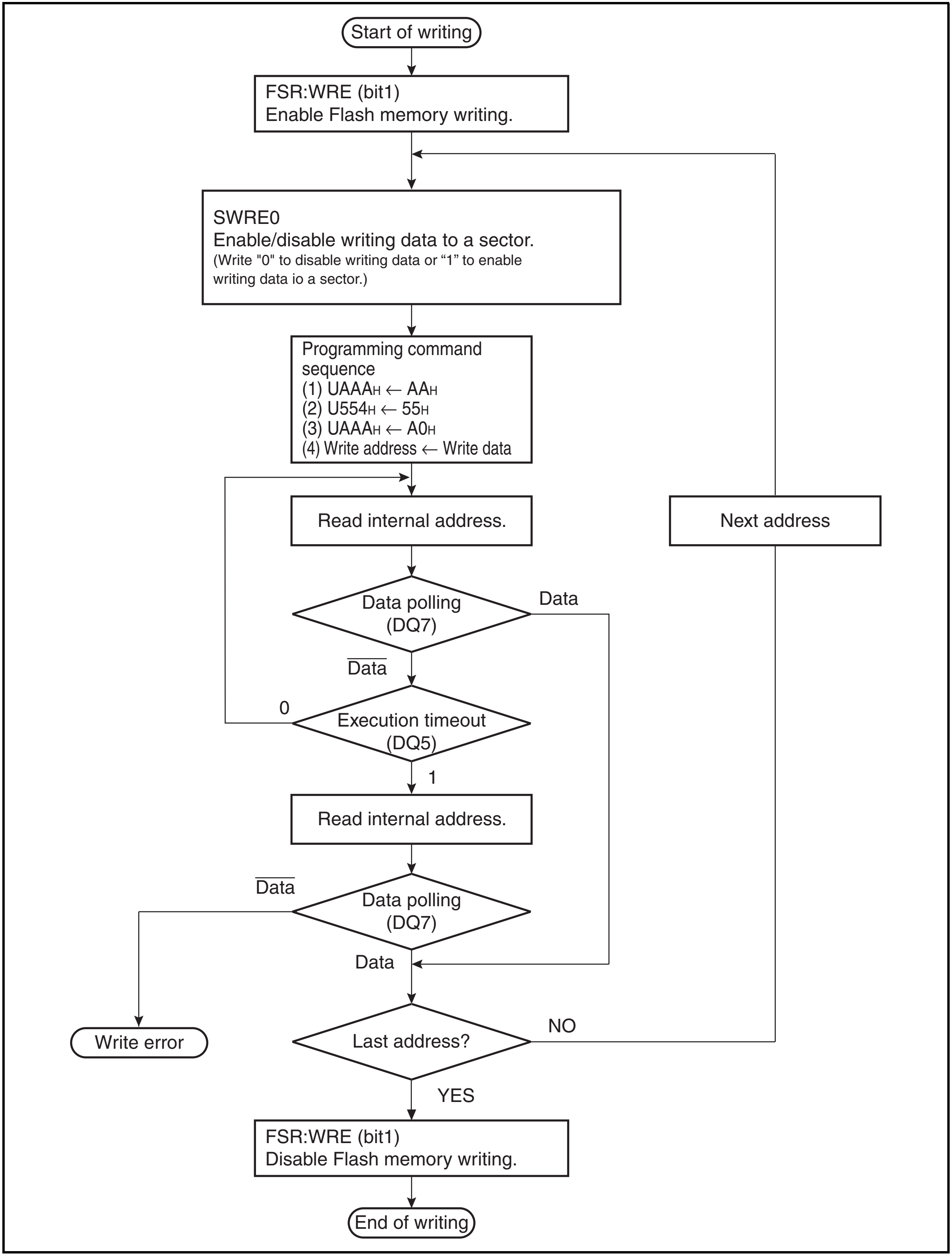
● Note on writing data

- Bit data cannot be returned from "0" to "1" by writing. When "1" is written to bit data that is currently "0", the data polling function (DQ7) or toggle operation (DQ6) is not terminated, 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 writing 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 writing.
- During writing, if a hardware reset occurs, the integrity of data being written to the current address is not guaranteed. Start writing the data from the chip-erase command again.

■ Flash Memory Writing Procedure

- Figure 28.6-1 gives an example of the procedure for writing data to the Flash memory. The hardware sequence flag can be used to check the operating state of the automatic algorithm in the Flash memory. The data polling flag (DQ7) is used for checking the end of writing data into Flash memory in this example.
- Data for flag checking is read from the address to which data has been last written.
- Since the data polling flag (DQ7) and the execution timeout flag (DQ5) are changed simultaneously, check the data polling flag (DQ7) even when the execution timeout flag (DQ5) is "1".
- Similarly, since the toggle bit flag (DQ6) stops toggling at the same time as the execution timeout flag (DQ5) changes to "1", check DQ6 after DQ5 changes to "1".

Figure 28.6-1 Sample Procedure for Writing to Flash Memory



28.6.3 Erasing All Data from Flash Memory (Chip Erase)

This section explains the procedure for issuing the chip erase command to erase all data in the Flash memory.

■ Erasing Data from Flash Memory (Chip Erase)

- To erase all data from the Flash memory, send the chip erase command mentioned in the command sequence table continuously 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 writing commands is complete.
- In chip erase, the user does not need to write data to the Flash memory before starting erasing data. While the automatic erase algorithm is running, it automatically writes "0" to all cells in the Flash memory before erasing data.

■ Note on Chip Erase

- The chip erase command is accepted only when writing data to all sectors has been enabled. The chip erase command is ignored if the bit for any sector in the flash memory sector write control register 0 (SWRE0) has been set to "0" (to disable writing data to that sector).
- During chip erase, if a hardware reset occurs, the integrity of data in the Flash memory is not guaranteed.

28.6.4 Erasing Specific Data from Flash Memory (Sector Erase)

This section explains the procedure for entering the sector erase command to erase a specific sector in the Flash memory. Sector-by-sector erasing is enabled and multiple sectors can also be specified at a time.

■ Erasing Specific Data from Flash Memory (Sector Erase)

To erase data from a specific sector in the Flash memory, send the sector erase command mentioned in the command sequence table continuously from the CPU to the Flash memory.

● Specifying a sector

- The sector erase command is executed in six bus operations. A minimum of 50 µs sector erase wait time starts as an address in the sector to be erased is specified as the address for the sixth cycle and the sector erase code (30_H) is written as data.
- To erase data from more than one sector, write the erase code (30_H) to an address in sector to be erased after writing the sector erase code to the address of the first sector to be erased as explained above.

● Note on specifying multiple sectors

- Sector erasing starts as a 50 µs sector erase wait time elapses after the last sector erase code has been written.
- To erase data from multiple sectors simultaneously, input the sector addresses and the erase code (in the sixth cycle of the command sequence) within a minimum of 50 µs sector erase wait time. If the erase code is input after the sector erase wait time elapses, it will not be accepted.
- The sector erase timer flag (DQ3) can be used to check whether it is valid to write sector erase codes continuously.
- Specify the address of a sector to be erased as the address at which the sector erase timer flag (DQ3) is read.

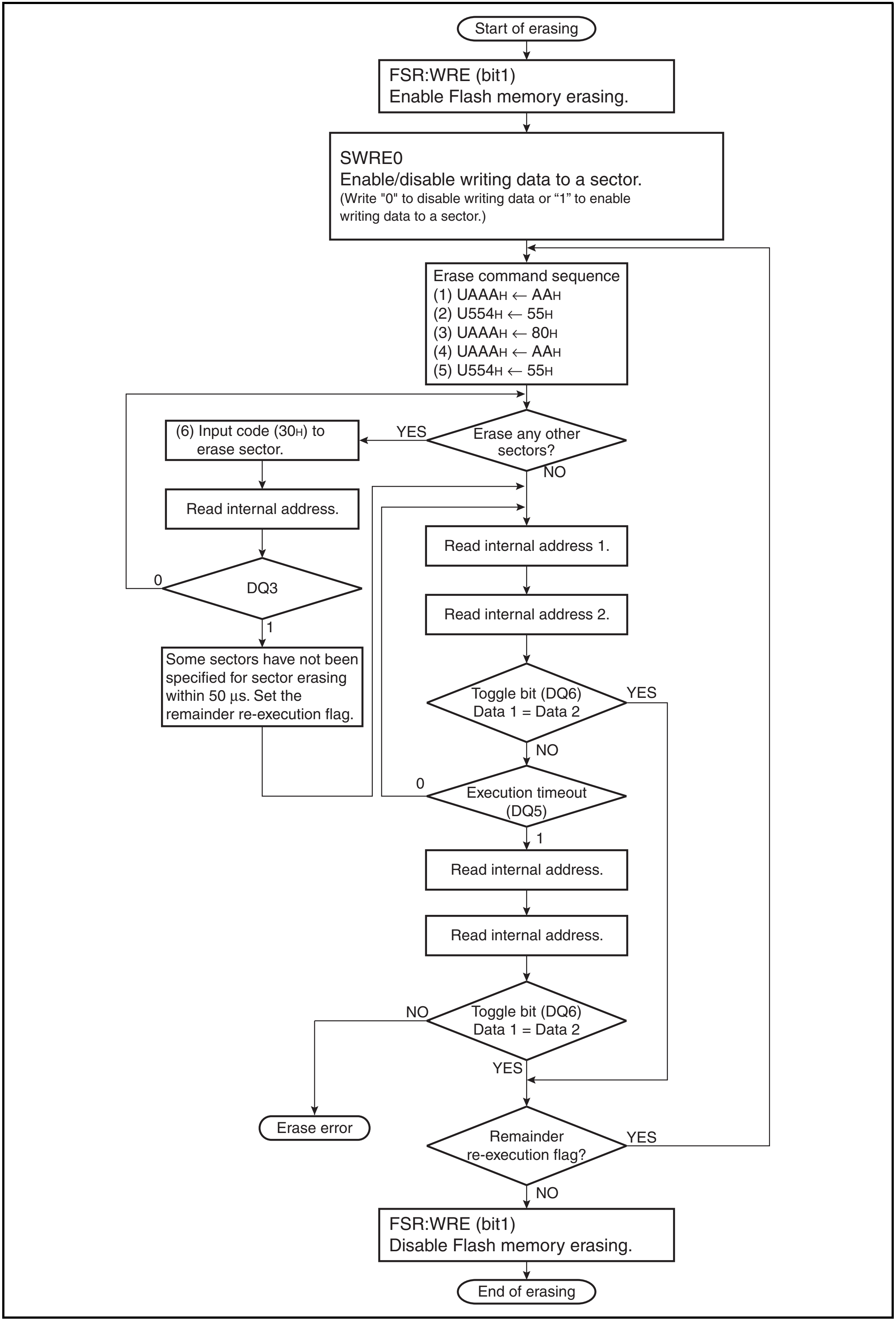
■ Flash Memory Sector Erasing Procedure

- Hardware sequence flags can be used to check the state of the automatic algorithm in the Flash memory. Figure 28.6-2 gives an example of the Flash memory sector erasing procedure. In this example, the toggle bit flag (DQ6) is used to check the end of sector erasing.
- The toggle bit flag (DQ6) stops toggling the output at the same time as the execution timeout flag (DQ5) changes to "1". Do check the toggle bit flag (DQ6) even when the execution timeout flag (DQ5) is "1".
- Since the data polling flag (DQ7) and the execution timeout flag (DQ5) are changed simultaneously, check the data polling flag (DQ7) when the execution timeout flag (DQ5) is "1".

■ Note on Erasing Data from Sectors

If a hardware reset occurs while data is being erased, the integrity of data in the Flash memory is not guaranteed. Run the sector erasing procedure again after a hardware reset occurs.

Figure 28.6-2 Sample Procedure for Erasing Data from Sectors in Flash Memory


Start of erasing
FSR:WRE (bit1)
Enable Flash memory erasing.
SWRE0
Enable/disable writing data to a sector.
(Write "0" to disable writing data or “1” to enable writing data to a sector.)
Erase command sequence
(1) UAAAH ← AAH
(2) U554H ← 55H
(3) UAAAH ← 80H
(4) UAAAH ← AAH
(5) U554H ← 55H
Erase any other sectors?
YES
NO
(6) Input code (30H) to erase sector.
Read internal address.
DQ3
0
1
Some sectors have not been specified for sector erasing within 50 µs. Set the remainder re-execution flag.
Read internal address 1.
Read internal address 2.
Toggle bit (DQ6) Data 1 = Data 2
YES
NO
Execution timeout (DQ5)
0
1
Read internal address.
Read internal address.
Toggle bit (DQ6) Data 1 = Data 2
NO
YES
Erase error
Remainder re-execution flag?
YES
NO
FSR:WRE (bit1)
Disable Flash memory erasing.
End of erasing

28.6.5 Suspending Sector Erasing from Flash Memory

This section explains the procedure for entering the sector erase suspend command to suspend sector erasing from the Flash memory. Data can be read from sectors not being erased.

■ Suspending Sector Erasing from Flash Memory

- To suspend the Flash memory sector erasing, send the sector erase suspend command mentioned in the command sequence table from the CPU to the Flash memory.
- The sector erase suspend command suspends the current sector erase operation, allowing data to be read from sectors that are not being erased.
- The sector erase suspend command is only enabled during the sector erase period including the erase wait time; it is ignored in chip erasing or writing.
- The sector erase suspend command is executed when the sector erase suspend code ($B0_H$) is written. Specify an address in the sector selected to be erased. If an attempt is made to execute the sector erase suspend command again when sector erasing has been suspended, the new sector erase suspend command input is ignored.
- When a sector erase suspend command is input during the sector erase wait period, the sector erase wait time ends immediately, the sector erase operation is stopped, and the Flash memory enters the erase stop state.
- When a sector erase suspend command is input during sector erasing after the sector erase wait period, the erase suspend state occurs after a maximum of 20 μs has elapsed since the issue of the sector erase suspend command.

Note:

To suspend sector erasing by issuing a sector erase suspend command, issue the command after 20 ms or longer has elapsed since the issue of a sector erase command or a sector erase resume command.

28.6.6 Resuming Sector Erasing from Flash Memory

This section explains the procedure for entering the sector erase resume command to resume suspended erasing of a sector in the Flash memory.

■ Resuming Sector Erasing from Flash Memory

- To resume suspended sector erasing, send the sector erase resume command mentioned in the command sequence table from the CPU to the Flash memory.
- The sector erase resume command resumes a sector erase operation suspended by the sector erase suspend command. The sector erase resume command is executed by writing erase resume code (30_H). Specify an address in the sector selected to be erased.
- A sector erase resume command input during sector erasing is ignored.

28.7 Operations of Dual Operation Flash

Pay attention in particular to the following points when using dual operation flash:

- **Interrupt generated when upper banks are updated**
- **Procedure of setting the sector swap enable bit in the flash memory status register (FSR:SSEN)**

■ Interrupt Generated When the Upper Bank Is Updated

The dual operation flash consists of two banks. Like conventional flash products, however, it cannot be erased/programmed and read at the same time in banks on the same side.

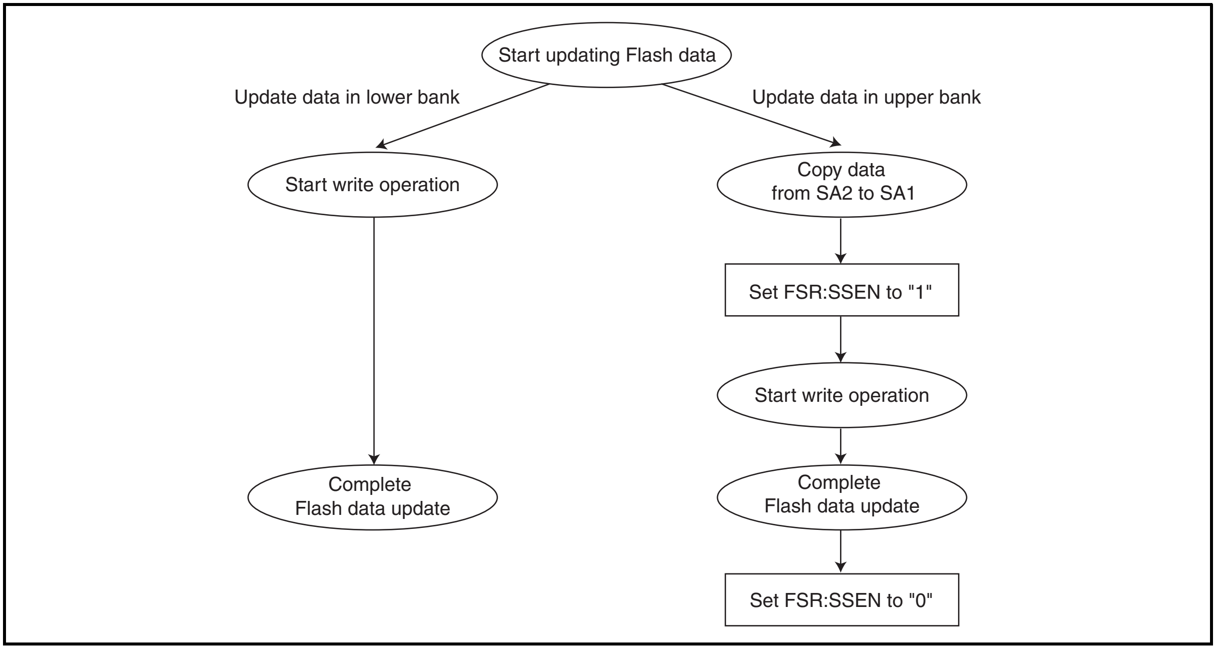
As SA2 contains an interrupt vector, an interrupt vector from the CPU cannot be read normally when an interrupt occurs during a write to the upper bank. Before the upper bank can be updated, the sector swap enable bit must be set to "1" (FSR:SSEN = 1). When an interrupt occurs, therefore, SA1 is accessed to read interrupt vector data. The same data must be copied to SA1 and SA2 before the sector swap enable bit (FSR:SSEN) is set.

■ Procedure for Setting Sector Swap Enable Bit (FSR:SSEN)

Figure 28.7-1 shows a sample procedure of setting the sector swap enable bit (FSR:SSEN).

To modify data in the upper bank, it is necessary to set FSR:SSEN to "1". While data is being written to the Flash memory, modifying the setting of FSR:SSEN is prohibited. The setting of FSR:SSEN can only be modified before the start of writing data to the Flash memory or after the completion of writing data to the Flash memory. In addition, control the Flash memory interrupts while setting FSR:SSEN as follows: before setting FSR:SSEN, disable the Flash memory interrupts; after setting FSR:SSEN, enable the interrupts.

Figure 28.7-1 Sample Procedure of Setting the Sector Swap Enable Bit (FSR:SSEN)



■ Operation during Writing/Erasing

It is prohibited to write data to the Flash memory within an interrupt routine when an interrupt occurs during Flash memory writing/erasing.
When two or more write/erase routines exist, wait for one write/erase routine to finish before executing another write/erase routine.

While data is being written to or erased from the Flash memory, state transition in the current mode (clock mode or standby mode) is prohibited. Ensure that writing data to or erasing data from the Flash memory ends before state transition occurs.

■ Register and Vector Table Addresses Related to Dual Operation Flash Memory Interrupts

Table 28.7-1 Register and Vector Table Addresses Related to Dual Operation Flash Memory Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Flash memory	IRQ23	ILR5	L23	$FFCC_H$	$FFCD_H$

28.8 Flash Security

The flash security controller function prevents contents of the Flash memory from being read by external pins.

■ 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 a chip erase command operation is executed.

It is advisable to write the protection code at the end of flash writing to avoid enabling unnecessary protection during writing.

Once flash security is enabled, a chip erase operation must be executed before data can be written to the Flash memory again.

28.9 Notes on Using Dual Operation Flash Memory

This section provides notes on using the Dual operation Flash memory.

■ Restriction on Using Toggle Bit Flag (DQ6)

When using the dual-operation Flash memory (The Flash memory write control program is executed on the Flash memory), the toggle bit flag (DQ6) cannot be used to check the operating state of the Flash memory during writing or erasing. Therefore, use the data polling flag (DQ7) to check the internal operating state of the Flash memory after writing data to the Flash memory or erasing data from the Flash memory as shown in the examples in Figure 28.6-1 and Figure 28.6-2.

The restriction above does not apply if the Flash memory write control program is executed on the RAM.

CHAPTER 29
EXAMPLE OF SERIAL PROGRAMMING CONNECTION

This chapter describes the example of serial programming connection.

29.1 Basic Configuration of Serial Programming Connection

The MB95430H Series support flash ROM serial on-board programming. This section describes the configuration.

■ Basic Configuration of Serial Programming Connection

The BGM adaptor MB2146-08-E, manufactured by , is used for serial onboard programming.

Figure 29.1-1 shows the basic configuration of serial programming connection.

Figure 29.1-1 Basic Configuration of Serial Programming Connection

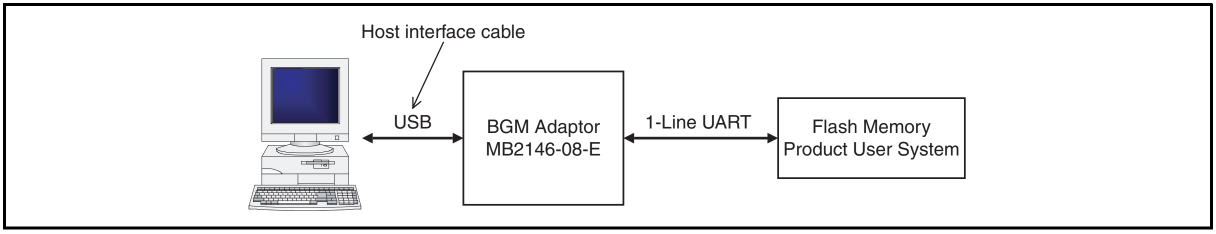


Table 29.1-1 Pins Used for Fujitsu Semiconductor Standard Serial Onboard Programming

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.
$\overline{\text{RST}}$	Reset	The $\overline{\text{RST}}$ pin is pulled up to V_{CC}.
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 29.2-1.)

- Oscillation Clock Frequency

The UART clock is provided by the main 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.

29.2 Example of Serial Programming Connection

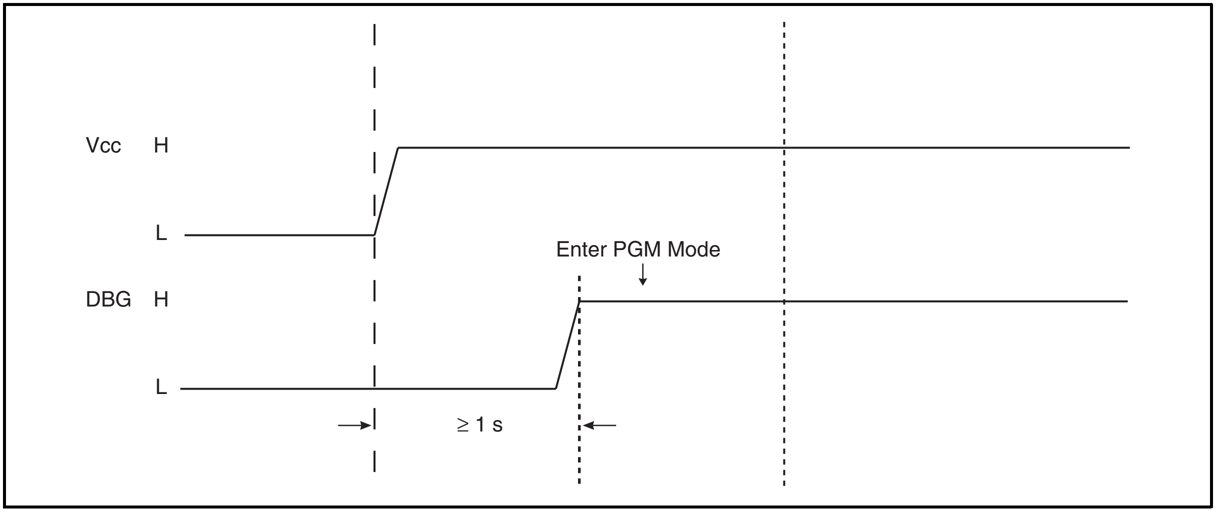
The microcontroller enters the PGM mode at the following timing.

■ 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 29.2-1 Timing Diagram

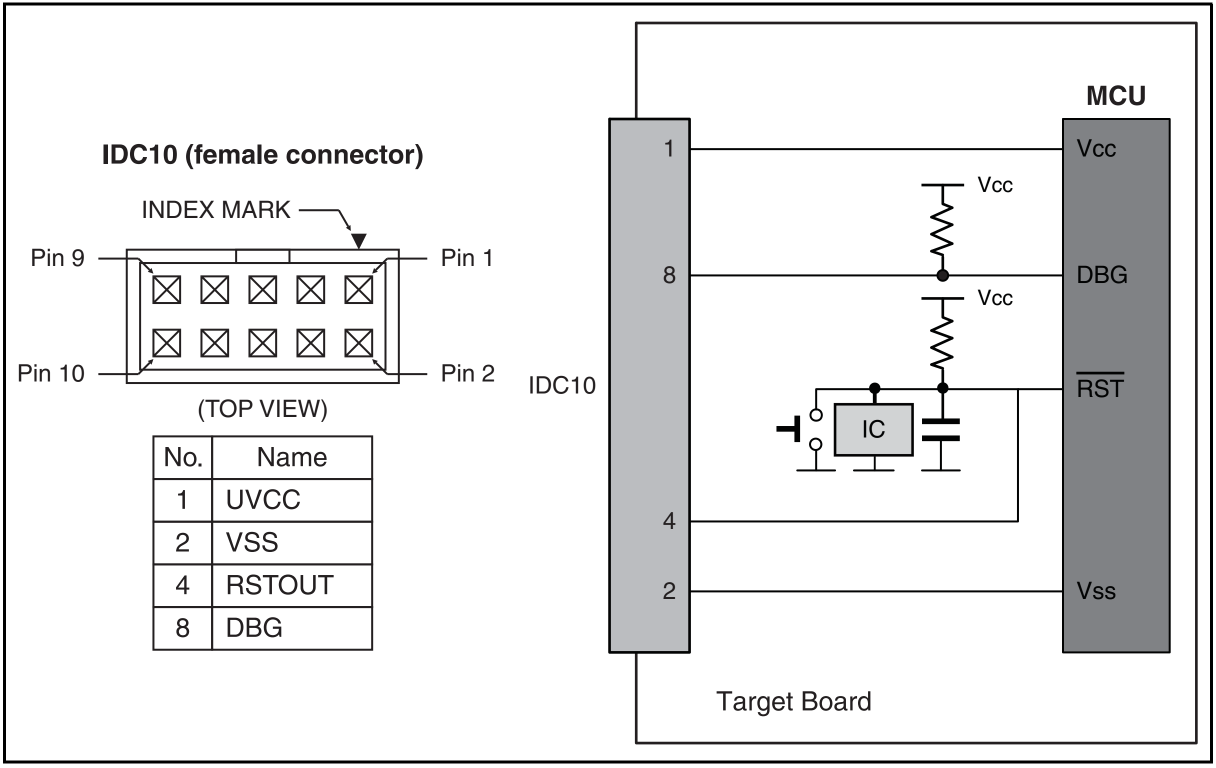


■ Example of Serial Programming Connection

Figure 29.2-2 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 29.2-2 Example of Serial Programming Connection



No.	Name
1	UVCC
2	VSS
4	RSTOUT
8	DBG

CHAPTER 30
NON-VOLATILE REGISTER FUNCTION (NVR)

This chapter describes the functions and operations of the NVR interface.

30.1 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 MB95430H 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 (6 bits)**
- **Watchdog Timer Selection ID (16 bits)**

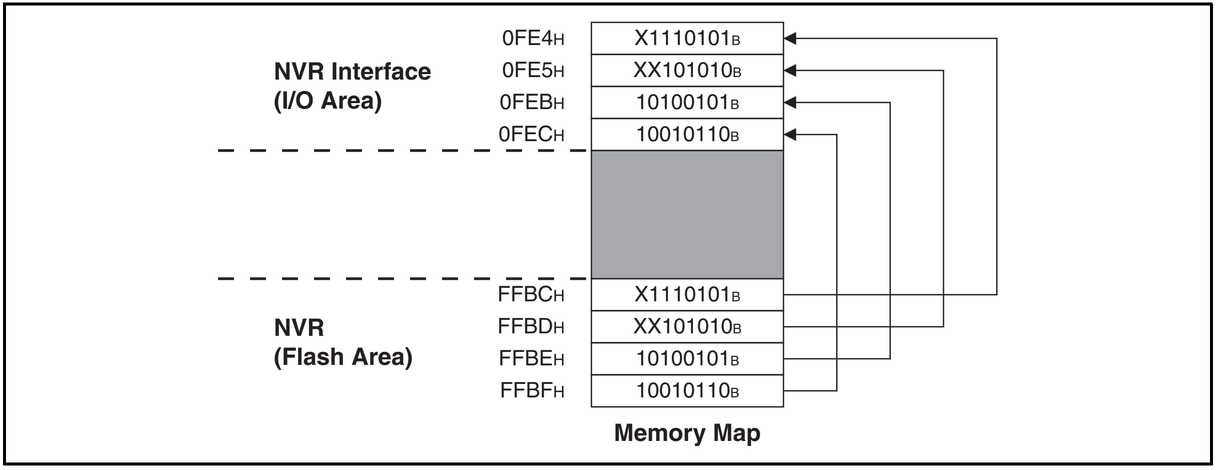
■ Functions of NVR Interface

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 30.1-1 and Figure 30.2-1 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 30.1-1 shows the retrieval of NVR during a reset.

Figure 30.1-1 Retrieval of NVR during Reset



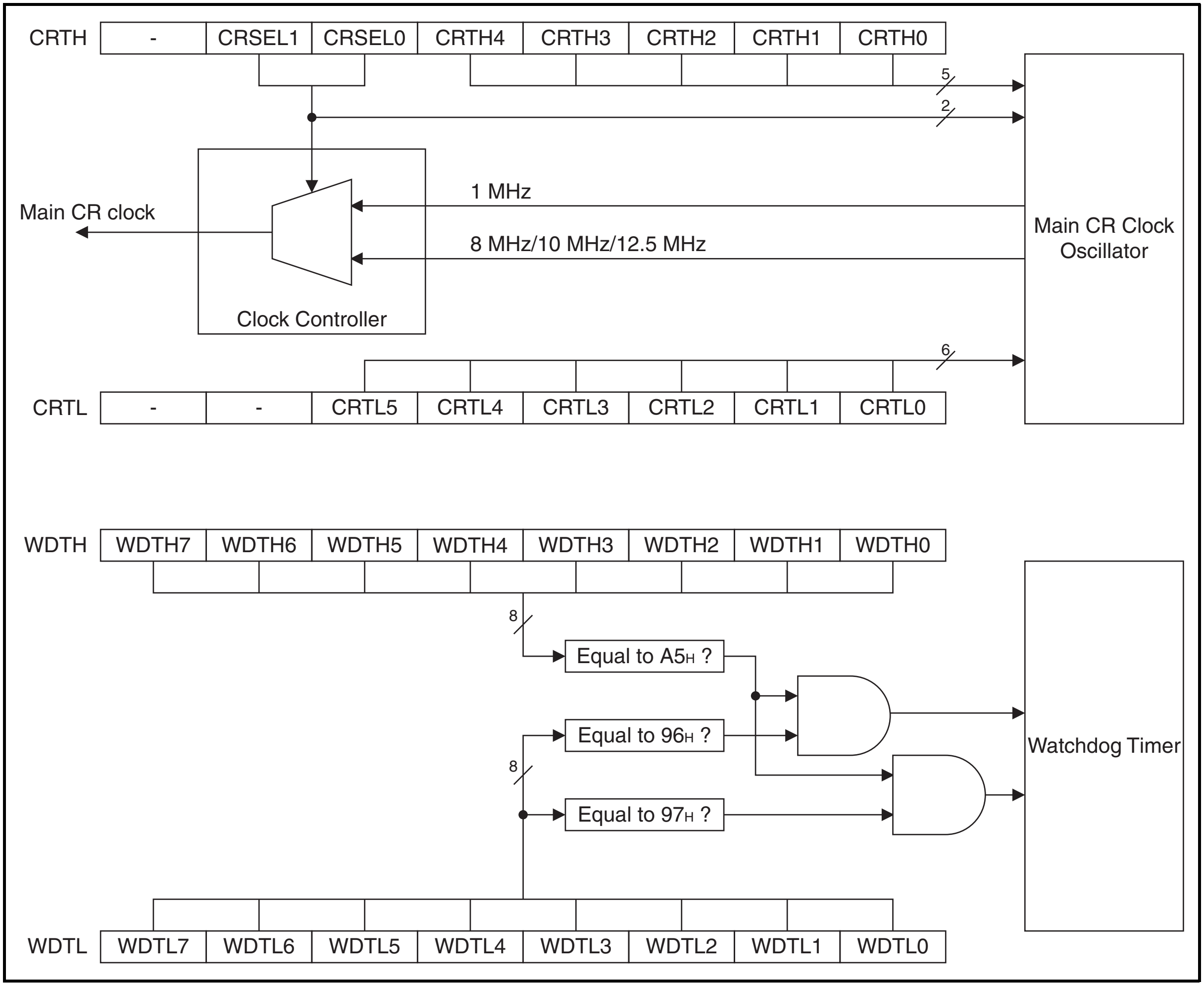
30.2 Configuration of NVR Interface

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 30.2-1 Block Diagram of NVR Interface



30.3 Registers of NVR Interface

This section lists the registers of the NVR interface.

■ Registers of NVR Interface

Figure 30.3-1 Registers of NVR Interface

Address	Register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE4$_H$	CRTH	–	CRSEL1	CRSEL0	CRTH4	CRTH3	CRTH2	CRTH1	CRTH0	0XXXXXXX$_B$
		R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0FE5$_H$	CRTL	–	–	CRTL5	CRTL4	CRTL3	CRTL2	CRTL1	CRTL0	00XXXXXX$_B$
		R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	
0FEB$_H$	WDTH	WDTH7	WDTH6	WDTH5	WDTH4	WDTH3	WDTH2	WDTH1	WDTH0	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	
0FEC$_H$	WDTL	WDTL7	WDTL6	WDTL5	WDTL4	WDTL3	WDTL2	WDTL1	WDTL0	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

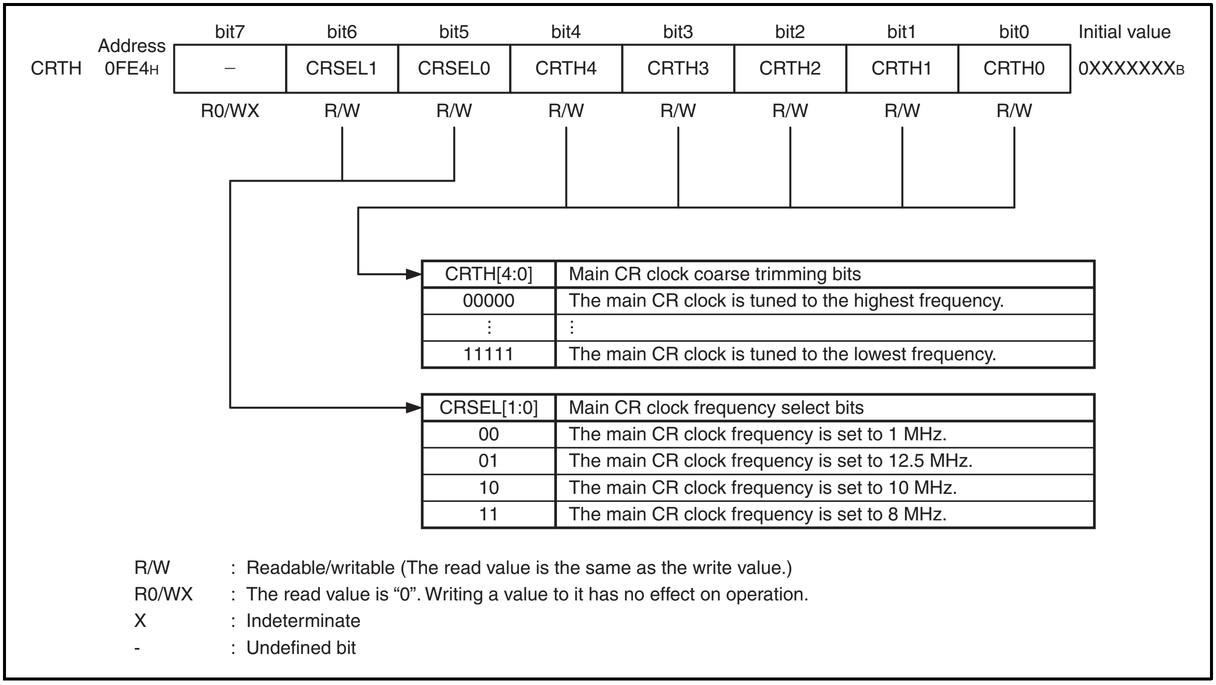
R/W	:	Readable/writable (The read value is the same as the write value.)
R/WX	:	Read only (Readable. Writing a value to it has no effect on operation.)
R0/WX	:	The read value is “0”. Writing a value to it has no effect on operation.
-	:	Undefined bit
X	:	Indeterminate

30.3.1 Main CR Clock Trimming Register (Upper) (CRTH)

Figure 30.3-2 shows the main CR clock trimming register (upper) (CRTH).

■ Main CR Clock Trimming Register (Upper) (CRTH)

Figure 30.3-2 Main CR Clock Trimming Register (Upper) (CRTH)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CRTH	0FE4H	–	CRSEL1	CRSEL0	CRTH4	CRTH3	CRTH2	CRTH1	CRTH0	0XXXXXXXB
		R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CRTH[4:0]	Main CR clock coarse trimming bits
00000	The main CR clock is tuned to the highest frequency.
⋮	⋮
11111	The main CR clock is tuned to the lowest frequency.

CRSEL[1:0]	Main CR clock frequency select bits
00	The main CR clock frequency is set to 1 MHz.
01	The main CR clock frequency is set to 12.5 MHz.
10	The main CR clock frequency is set to 10 MHz.
11	The main CR clock frequency is set to 8 MHz.

R/W : Readable/writable (The read value is the same as the write value.)
R0/WX : The read value is “0”. Writing a value to it has no effect on operation.
X : Indeterminate
- : Undefined bit

Table 30.3-1 Functions of Bits in Main CR Clock Trimming Register (Upper) (CRTH)

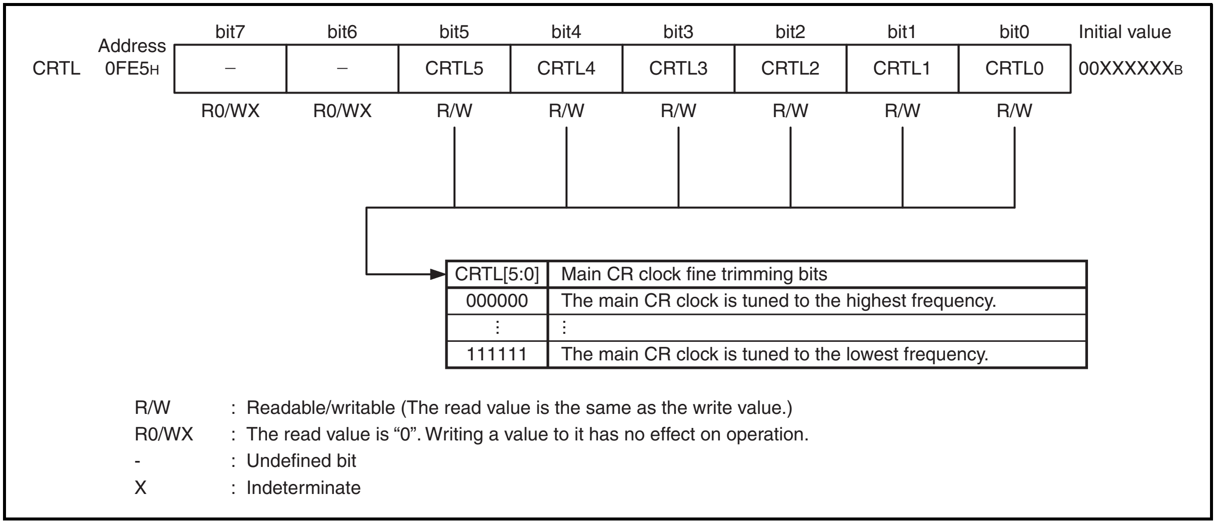
<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Undefined bit</td><td>The read value is always "0". Writing a value to it has no effect on operation.</td></tr>
<tr><td>bit6,
bit5</td><td>CRSEL[1:0]:
Main CR clock frequency select bits</td><td>These two bits are loaded from the flash address FFBC$_H$ (bit6, 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 "30.5 Notes on Using NVR" for notes on changing the main CR frequency selection.</td></tr>
<tr><td>bit4
to
bit0</td><td>CRTH[4:0]:
Main CR coarse trimming bits</td><td>These five bits are loaded from the flash address FFBC$_H$ (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 "30.4 Notes on Main CR Clock Trimming" and "30.5 Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.</td></tr>
</table>

30.3.2 Main CR Clock Trimming Register (Lower) (CRTL)

Figure 30.3-3 shows the main CR clock trimming register (lower) (CRTL).

■ Main CR Clock Trimming Register (Lower) (CRTL)

Figure 30.3-3 Main CR Clock Trimming Register (Lower) (CRTL)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CRTL	0FE5H	–	–	CRTL5	CRTL4	CRTL3	CRTL2	CRTL1	CRTL0	00XXXXXXB
		R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

CRTL[5:0]	Main CR clock fine trimming bits
000000	The main CR clock is tuned to the highest frequency.
⋮	⋮
111111	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 : The read value is “0”. Writing a value to it has no effect on operation.
- : Undefined bit
X : Indeterminate

Table 30.3-2 Functions of Bits in Main CR Trimming Register (Lower) (CRTL)

Bit Name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5 to bit0	CRTL[5:0]: Main CR fine trimming bits	These six bits are loaded from the flash address FFBD$_H$ (bit5 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. **CRTL[5:0]** / **Main CR clock frequency**: 000000$_B$ Highest; : :; 111111$_B$ Lowest See "30.4 Notes on Main CR Clock Trimming" and "30.5 Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.

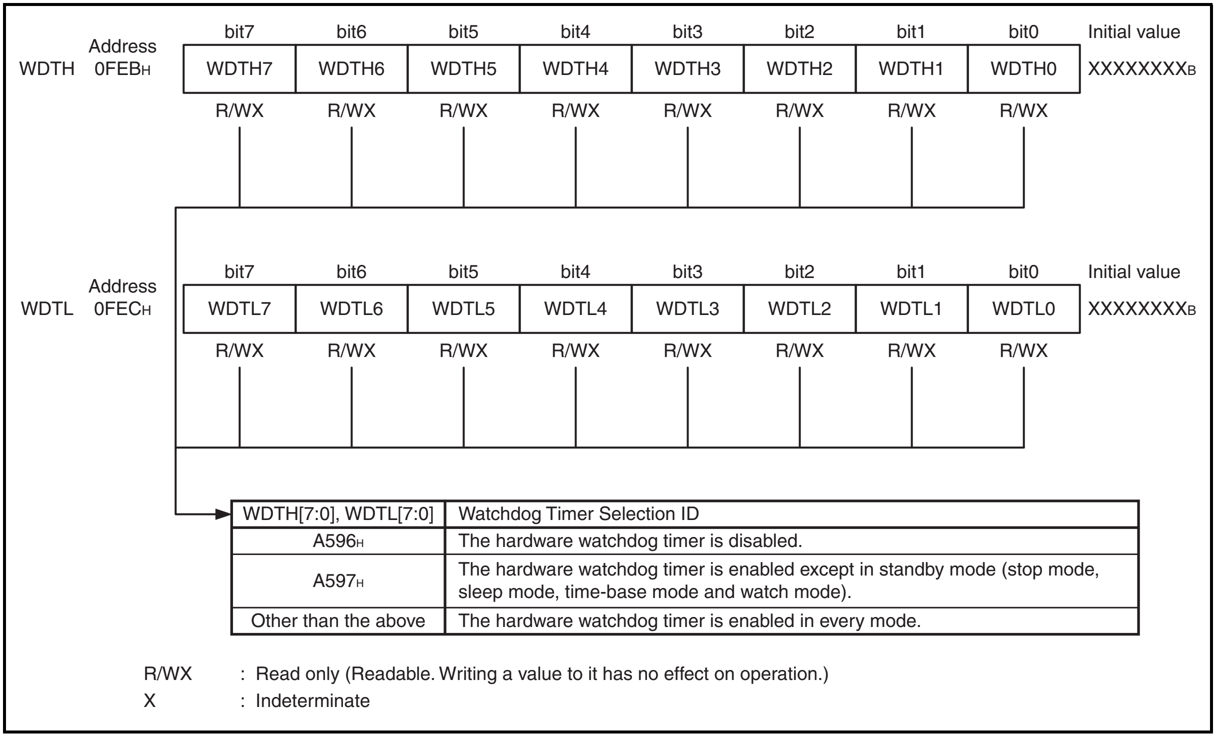
CRTL[5:0]	Main CR clock frequency
000000$_B$	Highest
:	:
111111$_B$	Lowest

30.3.3 Watchdog Timer Selection ID Registers (WDTH,WDTL)

Figure 30.3-4 shows watchdog timer selection ID registers (WDTH, WDTL).

■ Watchdog Timer Selection ID Registers (WDTH, WDTL)

Figure 30.3-4 Watchdog Timer Selection ID Registers (WDTH,WDTL)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WDTH	0FEB$_H$	WDTH7	WDTH6	WDTH5	WDTH4	WDTH3	WDTH2	WDTH1	WDTH0	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WDTL	0FEC$_H$	WDTL7	WDTL6	WDTL5	WDTL4	WDTL3	WDTL2	WDTL1	WDTL0	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

WDTH[7:0], WDTL[7:0]	Watchdog Timer Selection ID
A596$_H$	The hardware watchdog timer is disabled.
A597$_H$	The hardware watchdog timer is enabled except in standby mode (stop mode, sleep mode, time-base mode and watch mode).
Other than the above	The hardware watchdog timer is enabled in every mode.

R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
X : Indeterminate

Table 30.3-3 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 FFBE$_H$ (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 30.3-5 for watchdog timer selection. See "30.5 Notes on Using NVR" for notes on writing NVR values.

Table 30.3-4 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 FFBF$_H$ (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 30.3-5 for Watchdog Timer Selection. See "30.5 Notes on Using NVR" for notes on writing NVR values.

Table 30.3-5 Watchdog Timer Selection ID

WDTH[7:0], WDTL[7:0]	Function
$A596_H$	The hardware watchdog timer becomes invalid and the software watchdog timer becomes effective.
$A597_H$	The hardware watchdog timer is selected and the software watchdog timer becomes invalid. It is possible to stop in the standby mode (stop, sleep, time-base timer, and watch mode).
Other than the above	The hardware watchdog timer is selected and the software watchdog timer becomes invalid. Operation is continued in the standby mode (stop, sleep, time-base timer, and watch mode).

30.4 Notes on Main CR Clock Trimming

This section provides notes on main CR clock trimming.

After a hardware reset, the 11-bit CR clock trimming value will be loaded from the NVR flash area to registers in the NVR IO area.

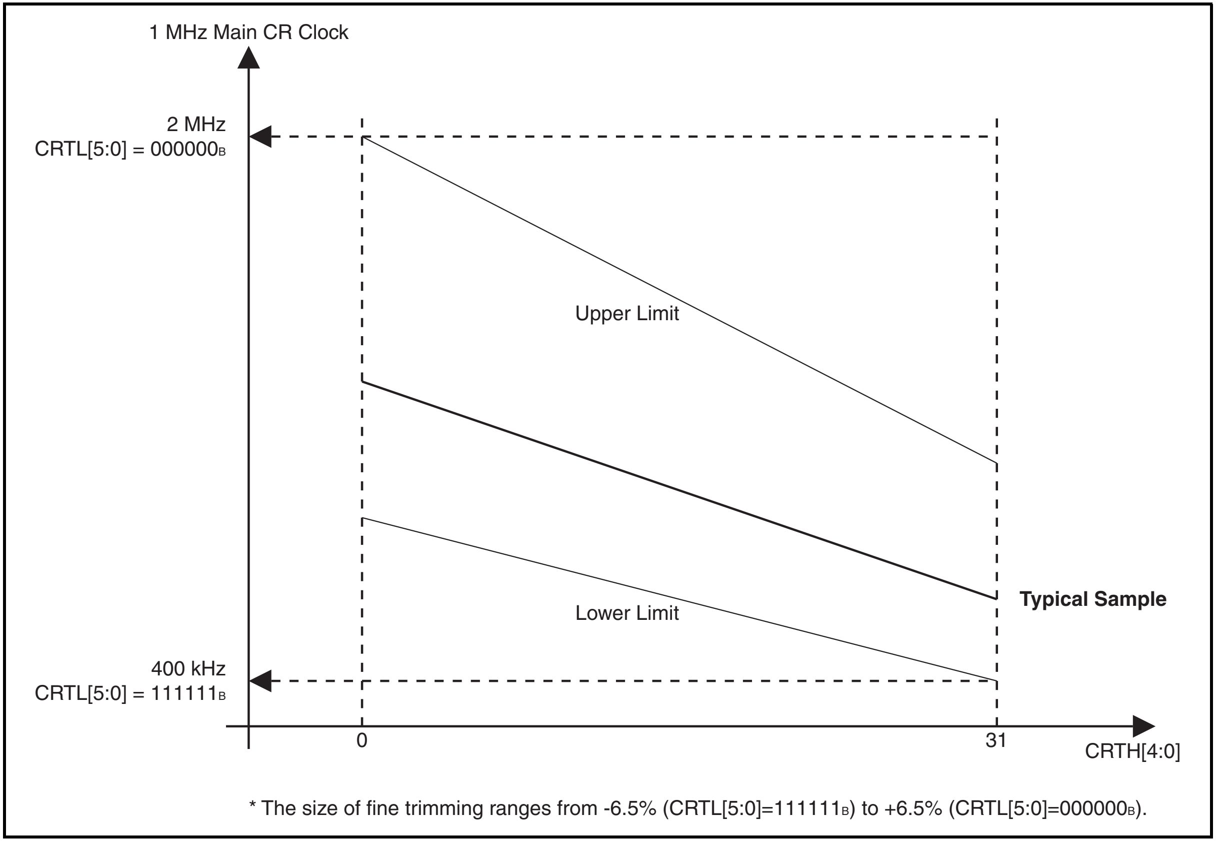
Table 30.4-1 shows the step size of CR Trimming.

Table 30.4-1 Step Size of CR Trimming

Function	Coarse trimming value CRTH[4:0]	Fine trimming value CRTL[5:0]
To achieve minimum frequency	11111_B	111111_B
To achieve maximum frequency	00000_B	000000_B
Step Size	20 kHz to 60 kHz	Non-linear

The relationship between coarse trimming step size and CR frequency is shown in the diagram below.

Figure 30.4-1 Coarse Trimming Step Size



30.5 Notes on Using NVR

This section provides notes on using NVR.

■ Note on Changing Main CR Frequency

1. The frequency of the main CR clock can be selected by writing different values to the bits CRTH:CRSEL[1:0]. 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.

■ Note 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:CRTH[4:0] and CRTL:CRTL[5:0].

   (2) Erase the flash.

   (3) Restore all data in CRTH:CRTH[4:0] and CRTL:CRTL[5:0] to the NVR flash area.

   If there is new data in CRTH:CRTH[4:0] and CRTL:CRTL[5:0], 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 31
SYSTEM CONFIGURATION CONTROLLER

This chapter describes the functions and operations of the system configuration controller (called the "controller" in this chapter).

31.1 Overview of Controller

The controller consists of two 8-bit registers, the SYSC1 register and the SYSC2 register, which are used to configure the clock and reset system and resources re-direction.

■ Functions of SYSC1

- Selection of the port/reset function for the PF2/$\overline{\text{RST}}$ pin
- Enabling/disabling reset output for the $\overline{\text{RST}}$ 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 EC0 input pin as the external count clock input pin for the 8/16-bit composite timer
- Selection of the BZ output for the buzzer

■ Functions of SYSC2

- Selection of the A/D converter channel AN16
- Selection of the OUT0 and OUT1 output pins for the output comparator
- Selection of the UCK, UI and UO pins for the UART/SIO
- Selection of the TRG and PPG pins for the 16-bit PPG
- Selection of the SCL and SDA pins for the I^{2}C

31.2 Registers of Controller

This section shows the registers of the controller.

■ Registers of Controller

Figure 31.2-1 Registers of Controller

System Configuration Register 1 (SYSC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE8$_H$	PGSEL	PFSEL	Reserved	Reserved	EC0SL	BZSEL	RSTOE	RSTEN	11000011$_B$
	R/W	R/W	R/W0	R/W0	R/W	R/W	R/W	R/W	

System Configuration Register 2 (SYSC2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004E$_H$	OPADSEL	OUTSEL1	OUTSEL0	I2CE_SEL1	I2CE_SEL0	UARTSEL	PPGSEL	TRGSEL	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/W0 : The write value is "0". The read value is the same as the write value.

31.2.1 System Configuration Register 1 (SYSC1)

This section provides details of the SYSC register 1.

■ System Configuration Register 1 (SYSC1)

Figure 31.2-2 System Configuration Register 1 (SYSC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE8$_H$	PGSEL	PFSEL	Reserved	Reserved	EC0SL	BZSEL	RSTOE	RSTEN	11000011$_B$
	R/W	R/W	R/W0	R/W0	R/W	R/W	R/W	R/W	

RSTEN	PF2 reset/GPIO function select bit
0	Enables the general-purpose I/O port function of the $\overline{\text{RST}}$/PF2 pin.
▨ 1	Enables the reset input function of the $\overline{\text{RST}}$/PF2 pin.

RSTOE	Reset output enable/disable bit
0	Disables the reset output function of the $\overline{\text{RST}}$/PF2 pin.
▨ 1	Enables the reset output function of the $\overline{\text{RST}}$/PF2 pin.

BZSEL	Buzzer output select bit
▨ 0	Selects PG1/BZ as the buzzer output pin.
1	Selects P01/BZ as the buzzer output pin.

EC0SL	EC0 clock select bit
▨ 0	Selects the P12/EC0 pin as the external count clock input pin.
1	Selects the P07/EC0 pin as the external count clock input pin.

Reserved	Reserved bit
▨ 0	Always set this bit to “0”.

Reserved	Reserved bit
▨ 0	Always set this bit to “0”.

PFSEL	PF1/PF0 function select bit
0	Selects the PF1/PF0 pins as main clock oscillator pins.
▨ 1	Selects the PF1/PF0 pins as general-purpose I/O ports.

PGSEL	PG2/PG1 function select bit
0	Selects the PG2/PG1 pins as subclock oscillator pins.
▨ 1	Selects the PG2/PG1 pins as general-purpose I/O ports.

R/W : Readable/writable (The read value is the same as the write value.)
R/W0 : The write value is “0”. The read value is the same as the write value.
▨ : Initial value

Table 31.2-1 Functions of Bits in SYSC1 Register (SYSC1)

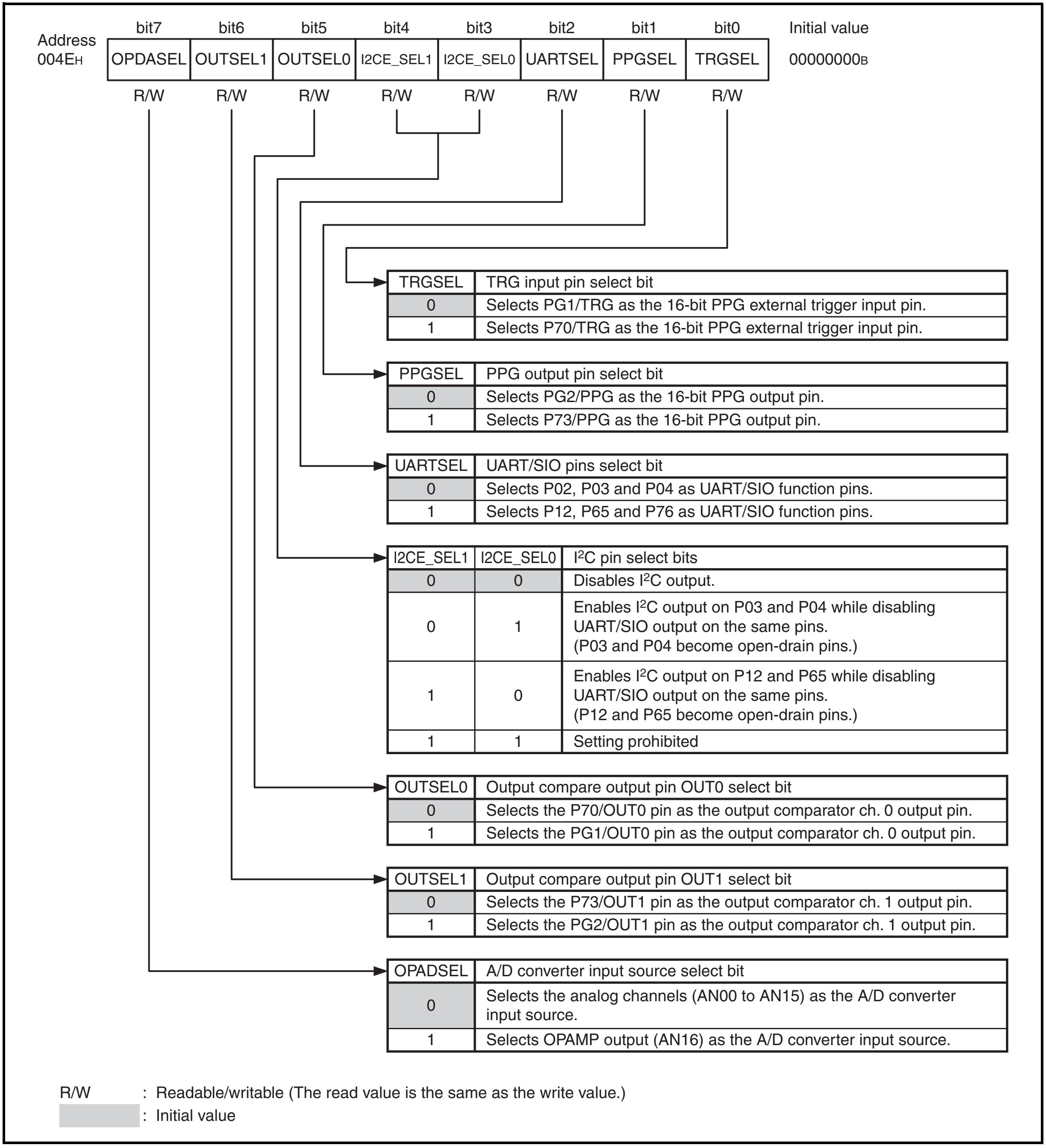
Bit name		Function
bit7	PGSEL: PG2/PG1 function select bit	This bit is used to select the function of the PG2/PG1 pins. If this bit is set to "0", the PG2/PG1 pins are selected as subclock oscillator pins, and the subclock oscillation is enabled or disabled by the subclock oscillation enable bit (SYCC2:SOSCE). If this bit is set to "1", the PG2/PG1 pins are selected as general-purpose I/O ports.
bit6	PFSEL: PF1/PF0 function select bit	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.
bit5, bit4	Reserved bits	Always set these bits to "0".
bit3	EC0SL: EC0 clock select bit	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 first. 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.
bit2	BZSEL: Buzzer output select bit	This bit is used to select the buzzer output pin from the P01 pin and the PG1 pin. If this bit is set to "0", the PG1 pin is selected as the buzzer output pin. If this bit is set to "1", the P01 pin is selected as the buzzer output pin.
bit1	RSTOE: Reset output enable/ disable bit	This bit is used to enable and disable the reset output function of the $\overline{\text{RST}}$/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 description of the reset input enable/disable bit (SYSC1:RSTEN) in this register. If this bit is set to "0", the reset output function of the $\overline{\text{RST}}$/PF2 pin is disabled. If this bit is set to "1", the reset output function of the $\overline{\text{RST}}$/PF2 pin is enabled.
bit0	RSTEN: PF2 reset/GPIO function select bit	This bit is used to enable and disable the reset input function of the $\overline{\text{RST}}$/PF2 pin. The reset input function is always enabled in MB95F434H regardless of the setting of this bit. If this bit is set to "0", the reset input function of the $\overline{\text{RST}}$/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 $\overline{\text{RST}}$/PF2 pin is enabled, and the general-purpose I/O port function is disabled. Set bit2 in the PDRF register to "1" before modifying this bit.

31.2.2 System Configuration Register 2 (SYSC2)

This section provides details of the SYSC register 2.

■ System Configuration Register 2 (SYSC2)

Figure 31.2-3 System Configuration Register 2 (SYSC2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004EH	OPDASEL	OUTSEL1	OUTSEL0	I2CE_SEL1	I2CE_SEL0	UARTSEL	PPGSEL	TRGSEL	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TRGSEL	TRG input pin select bit
0	Selects PG1/TRG as the 16-bit PPG external trigger input pin.
1	Selects P70/TRG as the 16-bit PPG external trigger input pin.

PPGSEL	PPG output pin select bit
0	Selects PG2/PPG as the 16-bit PPG output pin.
1	Selects P73/PPG as the 16-bit PPG output pin.

UARTSEL	UART/SIO pins select bit
0	Selects P02, P03 and P04 as UART/SIO function pins.
1	Selects P12, P65 and P76 as UART/SIO function pins.

I2CE_SEL1	I2CE_SEL0	I^{2}C pin select bits
0	0	Disables I^{2}C output.
0	1	Enables I^{2}C output on P03 and P04 while disabling UART/SIO output on the same pins. (P03 and P04 become open-drain pins.)
1	0	Enables I^{2}C output on P12 and P65 while disabling UART/SIO output on the same pins. (P12 and P65 become open-drain pins.)
1	1	Setting prohibited

OUTSEL0	Output compare output pin OUT0 select bit
0	Selects the P70/OUT0 pin as the output comparator ch. 0 output pin.
1	Selects the PG1/OUT0 pin as the output comparator ch. 0 output pin.

OUTSEL1	Output compare output pin OUT1 select bit
0	Selects the P73/OUT1 pin as the output comparator ch. 1 output pin.
1	Selects the PG2/OUT1 pin as the output comparator ch. 1 output pin.

OPADSEL	A/D converter input source select bit
0	Selects the analog channels (AN00 to AN15) as the A/D converter input source.
1	Selects OPAMP output (AN16) as the A/D converter input source.

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

Table 31.2-2 Functions of Bits in SYSC2 Register (SYSC2)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>OPADSEL:
A/D converter input source select bit</td><td>This bit is used to select the A/D converter analog input source.
If this bit is set to "0", the analog channels AN00-AN15 will be used as the A/D converter analog input source.
If this bit is set to "1", the analog channel OPAMP output (AN16) will be used as the A/D converter analog input source.</td></tr>
<tr><td>bit6</td><td>OUTSEL1:
OUT1 select bit</td><td>This bit is used to select which pin is used as the output comparator ch. 1 output pin.
If this bit is set to "0", P73/OUT1 is used as the output comparator ch. 1 output pin.
If this bit is set to "1", PG2/OUT1 is used as the output comparator ch. 1 output pin.</td></tr>
<tr><td>bit5</td><td>OUTSEL0:
OUT0 select bit</td><td>This bit is used to select which pin is used as the output comparator ch. 0 output pin.
If this bit is set to "0", P70/OUT0 is used as the output comparator ch. 0 output pin.
If this bit is set to "1", PG1/OUT0 is used as the output comparator ch. 0 output pin.</td></tr>
<tr><td>bit4,
bit3</td><td>I2CE_SEL[1:0]:
I^{2}C pin select bits</td><td>These bits are used to select which pins are to be used as SCL and SDA of I2C.
<table>
<tr><th>I2CE_SEL1</th><th>I2CE_SEL0</th><th>I^{2}C pin select bits</th></tr>
<tr><td>0</td><td>0</td><td>Disables I^{2}C output.</td></tr>
<tr><td>0</td><td>1</td><td>Enables I^{2}C output on P03 and P04 while disabling UART/SIO output on the same pins.
(P03 and P04 become open-drain pins.)</td></tr>
<tr><td>1</td><td>0</td><td>Enables I^{2}C output on P12 and P65 while disabling UART/SIO output on the same pins.
(P12 and P65 become open-drain pins.)</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
<tr><td>bit2</td><td>UARTSEL:
UART/SIO pin select bit</td><td>This bit is used to select which pins are to used as the UCK, UO and UI pins of the UART/SIO.
If this bit is set to "0", the P02, P03 and P04 pins are used as the UCK, UO and UI pins of the UART/SIO respectively.
If this bit is set to "1", the P12, P65 and P76 pins are used as the UCK, UO and UI pins of the UART/SIO respectively.</td></tr>
<tr><td>bit1</td><td>PPGSEL:
Buzzer output select bit</td><td>This bit is used to select which pin is to used as PPG, the 16-bit PPG output pin.
If this bit is set to "0", PG2 is used as PPG, the 16-bit PPG output pin.
If this bit is set to "1", P73 is used as PPG, the 16-bit PPG output pin.</td></tr>
<tr><td>bit0</td><td>TRGSEL:
PPG trigger pin select bit</td><td>This bit is used to select which pin is to used as TRG of the 16-bit PPG.
If this bit is set to "0", PG1 is used as TRG of the 16-bit PPG.
If this bit is set to "1", P70 is used as TRG of the 16-bit PPG.</td></tr>
</table>

Note:

To keep the reset input/output function after the reset, SYSC1:RSTEN and SYSC1:RSTOE are initialized to "1" after the power is switched on. They will not be initialized by any other type of reset.

If the reset input/output functions have to be used in the system, it is strongly recommended that SYSC1: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.

When I^{2}C output on P03 and P04 is enabled, UART/SIO output on the same pins will be disabled.

When I^{2}C output on P65 and P12 is enabled, UART/SIO output on the same pins will be disabled.

31.3 Notes on Using Controller

This section provides notes on using the controller.

■ Notes on Using Controller

● Settings for pins X0, X0A, X1 and X1A

To use pins X0 and X1 as main oscillation clock pins, set SYSC1:PFSEL to "0".

To use pins X0A and X1A as sub-oscillation clock pins, set SYSC1:PGSEL to "0".

APPENDIX

This chapter 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 MB95430H Series.

■ I/O Map

Table A-1 I/O MAP (1 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0000_H	PDR0	Port 0 data register	R/W	00000000_B
0001_H	DDR0	Port 0 direction register	R/W	00000000_B
0002_H	PDR1	Port 1 data register	R/W	00000000_B
0003_H	DDR1	Port 1 direction register	R/W	00000000_B
0004_H	—	(Disabled)	—	—
0005_H	WATR	Oscillation stabilization wait time setting register	R/W	11111111_B
0006_H	—	(Disabled)	—	—
0007_H	SYCC	System clock control register	R/W	$0000X011_B$
0008_H	STBC	Standby control register	R/W	$00000XXX_B$
0009_H	RSRR	Reset source register	R/W	$000XXXXX_B$
$000A_H$	TBTC	Time-base timer control register	R/W	00000000_B
$000B_H$	WPCR	Watch prescaler control register	R/W	00000000_B
$000C_H$	WDTC	Watchdog timer control register	R/W	$00XX0000_B$
$000D_H$	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	PDR6	Port 7 data register	R/W	00000000_B
0019_H	DDR6	Port 7 direction register	R/W	00000000_B
0020_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

Table A-1 I/O MAP (2 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0038_H	BUZZ	Buzzer control register	R/W	00000000_B
0039_H	—	(Disabled)	—	—
$003A_H$	CMR0	Voltage comparator control register ch. 0	R/W	$000X0001_B$
$003B_H$	CMR1	Voltage comparator control register ch. 1	R/W	$000X0001_B$
$003C_H$	CMR2	Voltage comparator control register ch. 2	R/W	$000X0001_B$
$003D_H$	CMR3	Voltage comparator control register ch. 3	R/W	$000X0001_B$
$003E_H$	OPCR	OPAMP control register	R/W	00000011_B
$003F_H$ to 0041_H	—	(Disabled)	—	—
0042_H	PCNTH0	16-bit PPG status control register upper ch. 0	R/W	00000000_B
0043_H	PCNTL0	16-bit PPG status control register lower ch. 0	R/W	00000000_B
0044_H	PTGS0	16-bit PPG trigger source control register ch. 0	R/W	00000000_B
0045_H	—	(Disabled)	—	—
0046_H	OCUOC	16-bit output compare stop trigger control register	R/W	00000000_B
0047_H	—	(Disabled)	—	—
0048_H	EIC20	External interrupt circuit control register ch. 0/ch. 1	R/W	00000000_B
0049_H	EIC30	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
$004C_H$, $004D_H$	—	(Disabled)	—	—
$004E_H$	SYSC2	System control register 2	R/W	00000000_B
$004F_H$	—	(Disabled)	—	—
0050_H	IBCR00	I^2C bus control register 0 ch. 0	R/W	00000000_B
0051_H	IBCR10	I^2C bus control register 1 ch. 0	R/W	00000000_B
0052_H	IBSR0	I^2C bus status register ch. 0	R/W	00000000_B
0053_H	IDDR0	I^2C data register ch. 0	R/W	00000000_B
0054_H	IAAR0	I^2C address register ch. 0	R/W	00000000_B
0055_H	ICCR0	I^2C clock control register ch. 0	R/W	00000000_B
0056_H	SMC10	UART/SIO serial mode control register 1 ch. 0	R/W	00000000_B
0057_H	SMC20	UART/SIO serial mode control register 2 ch. 0	R/W	00100000_B
0058_H	SSR0	UART/SIO serial status and data register ch. 0	R/W	00000000_B
0059_H	TDR0	UART/SIO serial output data register ch. 0	R/W	00000000_B
$005A_H$	RDR0	UART/SIO serial input data register ch. 0	R/W	00000000_B
$005B_H$	—	(Disabled)	—	—
$005C_H$	TCDTH	16-bit free-run timer data register (upper)	R/W	00000000_B
$005D_H$	TCDTL	16-bit free-run timer data register (lower)	R/W	00000000_B
$005E_H$	CPCLRH	16-bit free-run timer compare clear register (upper)	R/W	11111111_B
$005F_H$	CPCLRL	16-bit free-run timer compare clear register (lower)	R/W	11111111_B

Table A-1 I/O MAP (3 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0060_H	TCCSH	16-bit free-run timer control status register (upper)	R/W	01000000_B
0061_H	TCCSL	16-bit free-run timer control status register (lower)	R/W	00000000_B
0062_H	ETCCSH	16-bit free-run timer extended control status register (upper)	R/W	00000000_B
0063_H	ETCCSL	16-bit free-run timer extended control status register (lower)	R/W	00000000_B
0064_H	OCCP0H	16-bit output compare channel 0 register (upper)	R/W	00000000_B
0065_H	OCCP0L	16-bit output compare channel 0 register (lower)	R/W	00000000_B
0066_H	OCCP1H	16-bit output compare channel 1 register (upper)	R/W	00000000_B
0067_H	OCCP1L	16-bit output compare channel 1 register (lower)	R/W	00000000_B
0068_H	OCSH	16-bit output compare control status register (upper)	R/W	00000000_B
0069_H	OCSL	16-bit output compare control status register (lower)	R/W	00000000_B
$006A_H$	OCMCR	16-bit output compare mode control register	R/W	00000000_B
$006B_H$	EOCS	16-bit output compare extended control status register	R/W	00000000_B
$006C_H$	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	—	(Disabled)	—	—
0071_H	FSR2	Flash memory status register 2	R/W	00000000_B
0072_H	FSR	Flash memory status register	R/W	$000X0000_B$
0073_H	SWRE0	Flash memory sector write control register 0	R/W	00000000_B
0074_H	FSR3	Flash memory status register 3	R	$0000XXXX_B$
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
$007F_H$ to $0F7F_H$	—	(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 (4 / 5)

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
$0F97_H$ to $0FA9_H$	—	(Disabled)	—	—
$0FAA_H$	PDCRH0	16-bit PPG down counter register (upper) ch. 0	R/W	00000000_B
$0FAB_H$	PDCRL0	16-bit PPG down counter register (lower) ch. 0	R/W	00000000_B
$0FAC_H$	PCSRH0	16-bit PPG cycle setting buffer register (upper) ch. 0	R/W	11111111_B
$0FAD_H$	PCSRL0	16-bit PPG cycle setting buffer register (lower) ch. 0	R/W	11111111_B
$0FAE_H$	PDUTH0	16-bit PPG duty setting buffer register (upper) ch. 0	R/W	11111111_B
$0FAF_H$	PDUTL0	16-bit PPG duty setting buffer register (lower) ch. 0	R/W	11111111_B
$0FB0_H$ to $0FBD_H$	—	(Disabled)	—	—
$0FBE_H$	PSSR0	UART/SIO prescaler select register ch. 0	R/W	00000000_B
$0FBF_H$	BRSR0	UART/SIO baud rate setting register ch. 0	R/W	00000000_B
$0FC0_H$, $0FC1_H$	—	(Disabled)	—	—
$0FC2_H$	AIDRH	A/D input disable register (upper)	R/W	00000000_B
$0FC3_H$	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	$0XXXXXXX_B$
$0FE5_H$	CRTL	Main CR clock trimming register (lower)	R/W	$00XXXXXX_B$
$0FE6_H$, $0FE7_H$	—	(Disabled)	—	—
$0FE8_H$	SYSC1	System configuration register 1	R/W	11000011_B
$0FE9_H$	CMCR	Clock monitoring control register	R/W	00000000_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)	—	—

Table A-1 I/O MAP (5 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
$0FEE_H$	ILSR	Input level select register	R/W	00000000_B
$0FEF_H$	WICR	Interrupt pin control register	R/W	01000000_B
$0FF0_H$ to $0FFF_H$	—	(Disabled)	—	—

- **R/W access symbols**
  R/W : Readable / Writable
  R : Read only
  W : Write only
- **Initial value symbols**
  0 : The initial value of this bit is "0".
  1 : The initial value of this bit is "1".
  X : The initial value of this bit is indeterminate.

Note:

Do not write to an address that is "(Disabled)". If a "(Disabled)" address is read, an undefined value is returned.

MB95430H Series

APPENDIX B Table of Interrupt Sources

This section shows the table of interrupt sources used in the MB95430H Series.

■ Table of Interrupt Sources

See "CHAPTER 5 CPU" for interrupt operation.

Table B-1 MB95430 Series

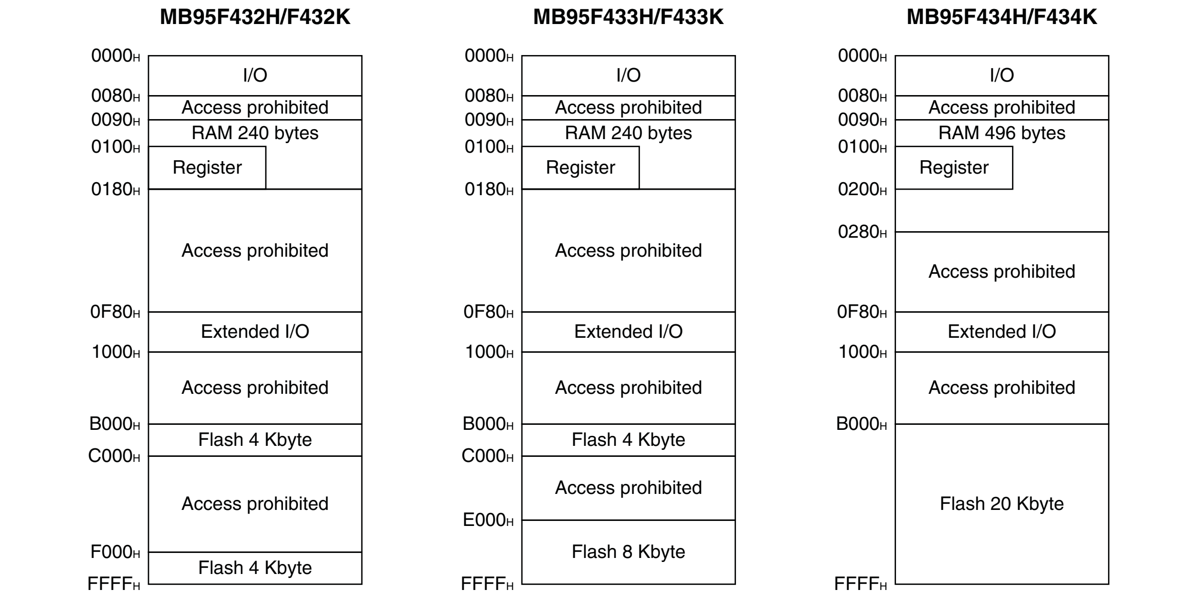
Interrupt source	Interrupt request number	Vector table address		Bit name of interrupt level setting register	Priority order of interrupt sources of the same level (occurring simultaneously)
		Upper	Lower		
External interrupt ch. 0	IRQ00	FFFA$_H$	FFFB$_H$	L00 [1:0]	High
External interrupt ch. 4					
External interrupt ch. 1	IRQ01	FFF8$_H$	FFF9$_H$	L01 [1:0]	
External interrupt ch. 5					
External interrupt ch. 2	IRQ02	FFF6$_H$	FFF7$_H$	L02 [1:0]	
External interrupt ch. 6					
External interrupt ch. 3	IRQ03	FFF4$_H$	FFF5$_H$	L03 [1:0]	
External interrupt ch. 7					
UART/SIO	IRQ04	FFF2$_H$	FFF3$_H$	L04 [1:0]	
8/16-bit composite timer ch. 0 (lower)	IRQ05	FFF0$_H$	FFF1$_H$	L05 [1:0]	
8/16-bit composite timer ch. 0 (upper)	IRQ06	FFEE$_H$	FFEF$_H$	L06 [1:0]	
Output compare ch. 0 match	IRQ07	FFEC$_H$	FFED$_H$	L07 [1:0]	
Output compare ch. 1 match	IRQ08	FFEA$_H$	FFEB$_H$	L08 [1:0]	
—	IRQ09	FFE8$_H$	FFE9$_H$	L09 [1:0]	
Voltage comparator ch. 0	IRQ10	FFE6$_H$	FFE7$_H$	L10 [1:0]	
Voltage comparator ch. 1	IRQ11	FFE4$_H$	FFE5$_H$	L11 [1:0]	
Voltage comparator ch. 2	IRQ12	FFE2$_H$	FFE3$_H$	L12 [1:0]	
Voltage comparator ch. 3	IRQ13	FFE0$_H$	FFE1$_H$	L13 [1:0]	
16-bit free-run timer (compare match/ zero-detect/overflow)	IRQ14	FFDE$_H$	FFDF$_H$	L14 [1:0]	
16-bit PPG	IRQ15	FFDC$_H$	FFDD$_H$	L15 [1:0]	
I^{2}C	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]	
Time-base 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 MB95430H Series.

■ Memory Map

Figure C-1 Memory Maps of Different Products



Parameter Part number	Flash memory	RAM
MB95F432H/F432K	8 Kbyte	240 bytes
MB95F433H/F433K	12 Kbyte	240 bytes
MB95F434H/F434K	20 Kbyte	496 bytes

MB95430H Series

APPENDIX D Pin States of MB95430H Series

Table D-1 below shows the pin states of the MB95430H Series in each mode.

■ State of Pins in Each Mode

Table D-1 Pin States in Each Mode (1 / 4)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		On reset
			SPL=0	SPL=1	SPL=0	SPL=1	
PF0/X0	OSC input	OSC input	Hi-Z	Hi-Z	Hi-Z	Hi-Z	—
	I/O port[*4]	I/O port[*4]	- Retain - Input interception[*2*4]	- Hi-Z - Input interception[*2*4]	- Retain - Input interception[*2*4]	- Hi-Z - Input interception[*2*4]	- Hi-Z - Input enabled[*1] (However, it does not function.)
PF1/X1	OSC output	OSC output	Hi-Z	Hi-Z	Hi-Z	Hi-Z	—
	I/O port[*4]	I/O port[*4]	- Retain - Input interception[*2*4]	- Hi-Z - Input interception[*2*4]	- Retain - Input interception[*2*4]	- Hi-Z - Input interception[*2*4]	- Hi-Z - Input enabled[*1] (However, it does not function.)
PG1/TRG/ADTG/BZ/OUT0/X0A	OSC input	OSC input	Hi-Z	Hi-Z	Hi-Z	Hi-Z	—
	I/O port[*4]/ peripheral function I/O	I/O port[*4]/ peripheral function I/O	- Retain[*11] - Input interception[*2*4*9]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*4*9]	- Retain[*11] - Input interception[*2*4*9]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*4*9]	- Hi-Z - Input enabled[*1] (However, it does not function.)
PG2/PPG/OUT1/X1A	OSC output	OSC output	Hi-Z	Hi-Z	Hi-Z	Hi-Z	—
	I/O port[*4]/ peripheral function I/O	I/O port[*4]/ peripheral function I/O	- Retain - Input interception[*2*4]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*4]	- Retain - Input interception[*2*4]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*4]	- Hi-Z - Input enabled[*1] (However, it does not function.)
PF2/$\overline{\text{RST}}$	Reset input[*3]	Reset input[*3]	Reset input[*3]	Reset input[*3]	Reset input[*3]	Reset input[*3]	Reset input[*3]
P00/INT00/AN00	I/O port/ peripheral function I/O/ analog input	I/O port/ peripheral function I/O/ analog input	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- - Hi-Z - Input enabled[*2] (However, it does not function.)

Table D-1 Pin States in Each Mode (2 / 4)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		On reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P01/INT01/ AN01/BZ	I/O port/ peripheral function I/O/ analog input	I/O port/ peripheral function I/O/ analog input	- Retain[*11] - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Retain[*11] - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - Input enabled[*2] (However, it does not function.)
P02/INT02/ AN02/UCK	I/O port/ peripheral function I/O/ analog input	I/O port/ peripheral function I/O/ analog input	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	
P03/INT03/ AN03/UO/ SDA	I/O port/ peripheral function I/O/ analog input	I/O port/ peripheral function I/O/ analog input	- Retain - Input interception[*2*10a] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*10a] (However, an external interrupt can be input when the external interrupt is enabled.)	- Retain - Input interception[*2*10a] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2*10a] (However, an external interrupt can be input when the external interrupt is enabled.)	
P04/INT04/ AN04/UI/ SCL							
P05/INT05/ AN05/TO0	I/O port/ peripheral function I/O/ analog input	I/O port/ peripheral function I/O/ analog input	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	
P06/INT06/ AN06/TO1							
P07/INT07/ AN07/EC0							
P12/EC0/UI/ SCL/DBG	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[*2*7*9*10b]	- Hi-Z - Input interception[*2*7*9*10b]	- Retain - Input interception[*2*7*9*10b]	- Hi-Z - Input interception[*2*7*9*10b]	- Hi-Z[*6] - Input enabled[*1] (However, it does not function.)

Table D-1 Pin States in Each Mode (3 / 4)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		On reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P60/ OPAMP_P*12	I/O port/ analog input	I/O port/ analog input	- Retain - Input interception*2*8	- Hi-Z - Input interception*2*8	- Retain - Input interception*2*8	- Hi-Z - Input interception*2*8	- Hi-Z - Input enabled*2 (However, it does not function.)
P61/ OPAMP_N*12							
P62/ OPAMP_O*12	I/O port/ analog output	I/O port/ analog output	- Retain*5 - Input interception*2	- Hi-Z*5 - Input interception*2	- Retain*5 - Input interception*2	- Hi-Z*5 - Input interception*2	- Hi-Z - Input enabled*1 (However, it does not function.)
P63/ CMP2_P/ AN12	I/O port/ analog input	I/O port/ analog input	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Hi-Z - Input enabled*2 (However, it does not function.)
P64/ CMP2_N/ AN13							
P65/ CMP3_O/ UO/SDA	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception*2*9*10b	- Hi-Z - Input interception*2*9*10b	- Retain - Input interception*2*9*10b	- Hi-Z - Input interception*2*9*10b	- Hi-Z - Input enabled*1 (However, it does not function.)
P66/ CMP3_P/ AN14	I/O port/ analog input	I/O port/ analog input	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Hi-Z - Input enabled*2 (However, it does not function.)
P67/ CMP3_N/ AN15							
P70/ CMP0_O/ OUT0/TRG	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception*2*9	- Hi-Z - Input interception*2*9	- Retain - Input interception*2*9	- Hi-Z - Input interception*2*9	- Hi-Z - Input enabled*1 (However, it does not function.)
P71/ CMP0_P/ AN08	I/O port/ peripheral function I/O/ analog input	I/O port/ analog input	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Hi-Z - Input enabled*2 (However, it does not function.)
P72/ CMP0_N/ AN09							
P73/ CMP1_O/ OUT1/PPG	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Hi-Z - Input enabled*1 (However, it does not function.)
P74/ CMP1_P/ AN10	I/O port/ analog input	I/O port/ analog input	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Retain - Input interception*2	- Hi-Z - Input interception*2	- Hi-Z - Input enabled*2 (However, it does not function.)
P75/ CMP1_N/ AN11							

Table D-1 Pin States in Each Mode (4 / 4)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		On reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P76/ CMP2_O/ UCK	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception*2*9	- Hi-Z - Input interception*2*9	- Retain - Input interception*2*9	- Hi-Z - Input interception*2*9	- Hi-Z - Input enabled*1 (However, it does not function.)

SPL: Pin state setting bit in the 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, a 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, its pin state is the same as that of other ports.

*2: "Input interception" means direct input gate operation from the pin is disabled.

*3: The pin state when PF2/$\overline{\text{RST}}$ is configured as reset pin

*4: The pin state when these pins are configured as GPIOs

*5: In stop mode or watch mode, analog output depends on OPAMP settings.

*6: In OCD mode, the pin state of P12 is retained for maintaining data communications.

*7: In OCD mode, P12 input is enabled for accepting the force break request.

*8: Analog input depends on OPAMP settings.

*9: If the external interrupt is enabled by the external interrupt control register (WICR), then the pin state becomes "Input enabled".

*10a: If P03 and P04 are used as I^{2}C pins and the MCU standby mode wakeup function of the I^{2}C is enabled, then the input buffers for P03 and P04 will be enabled.

*10b: If P12 and P65 are used as I^{2}C pins and the MCU standby mode wakeup function of the I^{2}C is enabled, then the input buffers for P12 and P65 will be enabled.

*11: In stop mode or watch mode, with the SPL bit set to "0", the buzzer output can be used even in main stop mode provided that a frequency output from the watch prescaler (BZCR:BUZ[2:0] = 101_B, 110_B or 111_B) has been selected.

*12: If the OPAMP is enabled, the input and output functions of all OPAMP pins are still operable even in stop mode or watch mode.

MB95430H Series

APPENDIX E Instruction Overview

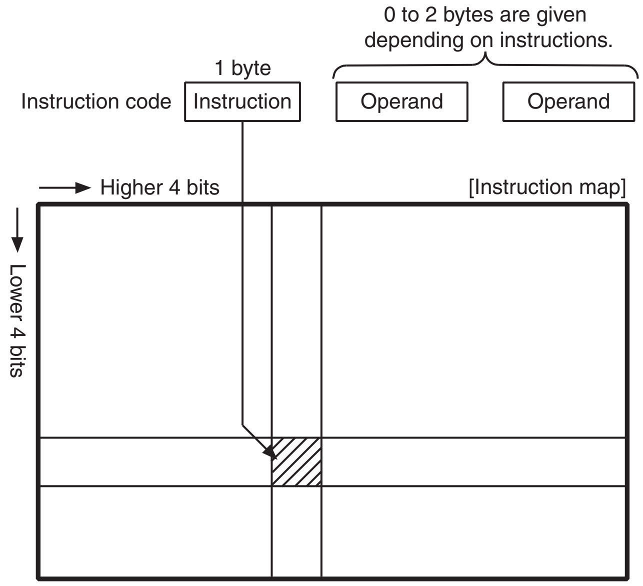
This section explains the instructions used in F^{2}MC-8FX.

■ Instruction Overview of F^{2}MC-8FX

In F^{2}MC-8FX, there are 140 kinds of one byte instructions (as the map, 256 bytes), and the instruction code is composed of the instruction and the operand following it.

Figure E-1 shows the correspondence of the instruction code and the instruction map.

Figure E-1 Instruction Code and Instruction Map



- The instruction is classified into following four types; forwarding system, operation system, branch system and others.
- There are various methods of addressing, and ten kinds of addressing can be selected by the selection and the operand specification of the instruction.
- This provides with the bit operation instruction, and can operate the read-modify-write (RMW) type of instruction.
- There is an instruction that directs special operation.

Code: CM26-00118-1EA

■ Explanation of Display Sign of Instruction

Table E-1 shows the explanation of the sign used by explaining the instruction code of this APPENDIX E.

Table E-1 Explanation of Sign in Instruction Table

Sign	Signification
dir	Direct address (8-bit length)
off	Offset (8-bit length)
ext	Extended address (16-bit length)
#vct	Vector table number (3-bit length)
#d8	Immediate data (8-bit length)
#d16	Immediate data (16-bit length)
dir:b	Bit direct address (8-bit length: 3-bit length)
rel	Branch relative address (8-bit length)
@	Register indirect (Example: @A, @IX, @EP)
A	Accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
AH	Upper 8-bit of accumulator (8-bit length)
AL	Lower 8-bit of accumulator (8-bit length)
T	Temporary accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
TH	Upper 8-bit of temporary accumulator (8-bit length)
TL	Lower 8-bit of temporary accumulator (8-bit length)
IX	Index register (16-bit length)
EP	Extra pointer (16-bit length)
PC	Program counter (16-bit length)
SP	Stack pointer (16-bit length)
PS	Program status (16-bit length)
dr	Either of accumulator or index register (16-bit length)
CCR	Condition code register (8-bit length)
RP	Register bank pointer (5-bit length)
DP	Direct bank pointer (3-bit length)
Ri	General-purpose register (8-bit length, i = 0 to 7)
x	This shows that x is immediate data. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
(x)	This shows that contents of x are objects of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
((x))	This shows that the address that contents of x show is an object of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)

■ Explanation of Item in Instruction Table

Table E-2 Explanation of Item in Instruction Table

Item	Description
MNEMONIC	It shows the assembly description of the instruction.
~	It shows the number of cycles of the instruction. One instruction cycle is a machine cycle. Note: The number of cycles of the instruction can be delayed by 1 cycle by the immediately preceding instruction. Moreover, the number of cycles of the instruction might be extended in the access to the I/O area.
#	It shows the number of bytes for the instruction.
Operation	It shows the operations for the instruction.
TL, TH, AH	They show the change (auto forwarding from A to T) in the content when each TL, TH, and AH instruction is executed. The sign in the column indicates the followings respectively. • -: No change • dH: upper 8 bits of the data described in operation. • AL and AH: the contents become those of the immediately preceding instruction's AL and AH. • 00: Become 00
N, Z, V, C	They show the instruction into which the corresponding flag is changed respectively. The sign in the column shows the followings respectively. • -: No change • +: Change • R: Become "0" • S: Become "1"
OP CODE	It shows the code of the instruction. When a pertinent instruction occupies two or more codes, it follows the following description rules. [Example] 48 to 4F: This shows 48, 49....4F.

E.1 Addressing

F²MC-8FX has the following ten types of addressings:

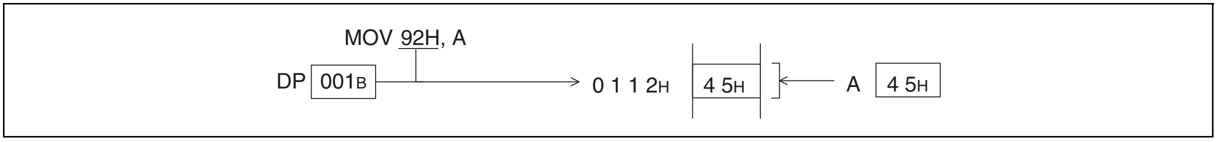
- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

■ Explanation of Addressing

● Direct addressing

This is used when accessing the direct area of "0000_H" to "$047F_H$" with addressing indicated "dir" in instruction table. In this addressing, when the operand address is "00_H" to "$7F_H$", it is accessed into "0000_H" to "$007F_H$". Moreover, when the operand address is "80_H" to "FF_H", the access can be mapped in "0080_H" to "$047F_H$" by setting of direct bank pointer DP. Figure E.1-1 shows an example.

Figure E.1-1 Example of Direct Addressing

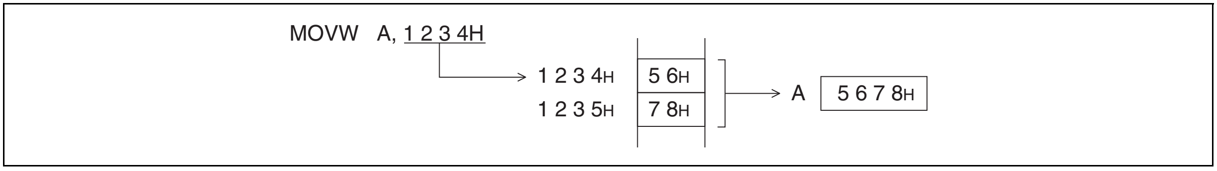


● Extended addressing

This is used when the area of the entire 64 Kbyte 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

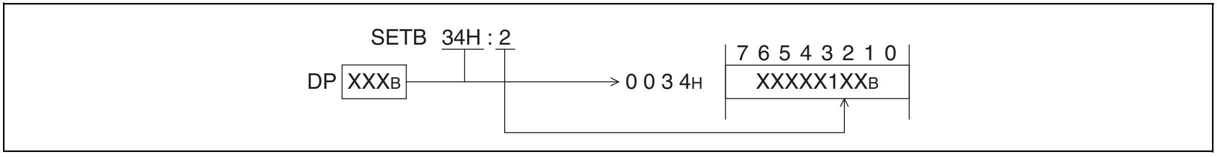


● Bit direct addressing

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.

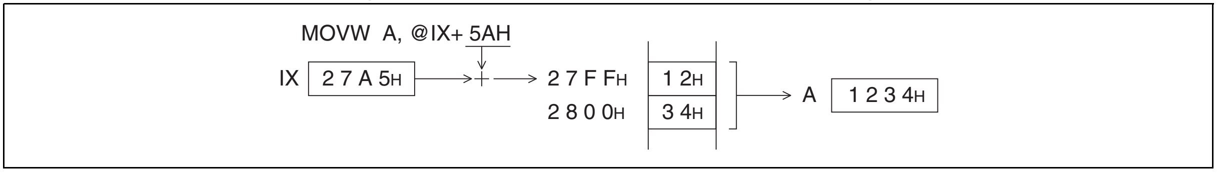
Figure E.1-3 Example of Bit Direct Addressing



● Index addressing

This is used when the area of the entire 64 Kbyte 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.

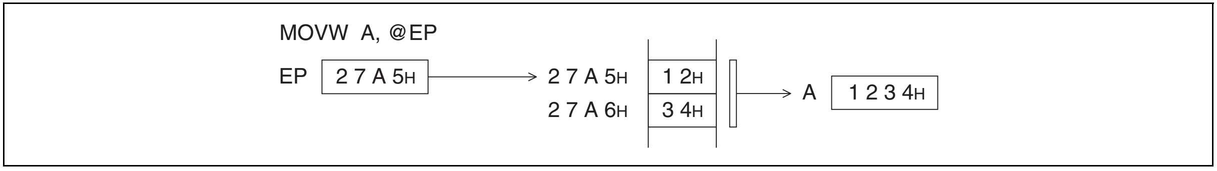
Figure E.1-4 Example of Index Addressing



● Pointer addressing

This is used when the area of the entire 64 Kbyte 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.

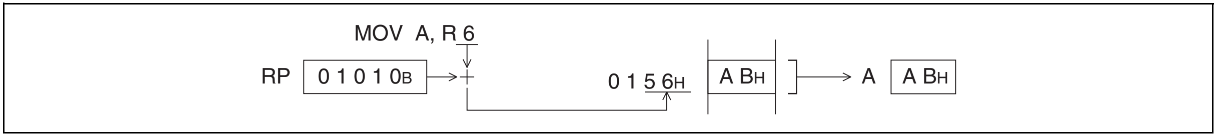
Figure E.1-5 Example of Pointer Addressing



● 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.

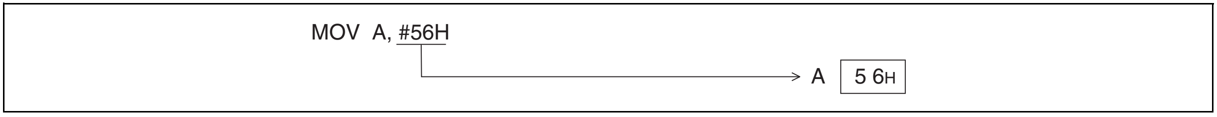
Figure E.1-6 Example of General-purpose Register Addressing



● 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

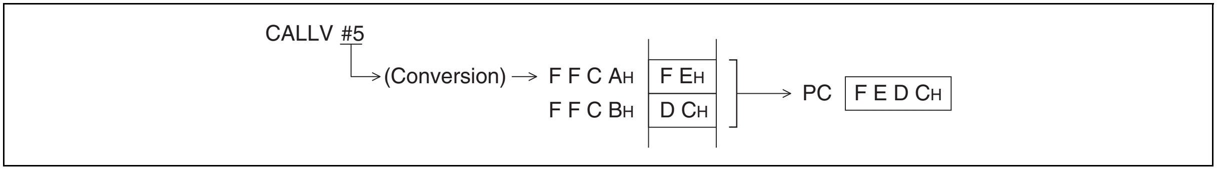
This is used when branching to the subroutine address registered in the table with the addressing shown "#vct" in the instruction table. In this addressing, information on "#vct" is contained in the operation code, and the address of the table is created using the combinations shown in Table E.1-1.

Table E.1-1 Vector Table Address Corresponding to "#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.

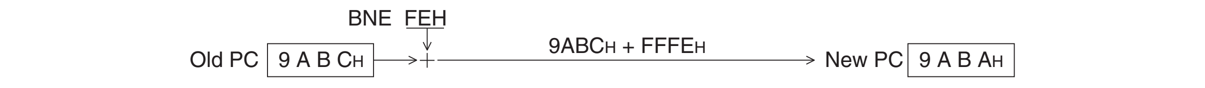
Figure E.1-8 Example of Vector Addressing



● 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

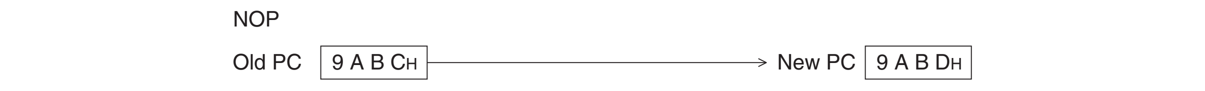


In this example, by jumping to the address where the operation code of BNE is stored, it results in an infinite loop.

● 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 addressings.

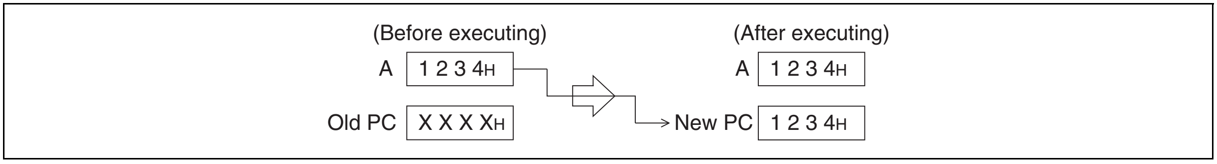
■ Special Instruction

● JMP @A

This instruction is to branch the content of A (accumulator) to PC (program counter) as an address. N pieces of the jump destination is arranged on the table, and one of the contents is selected and transferred to A. N branch processing can be done by executing this instruction.

Figure E.2-1 shows a summary of the instruction.

Figure E.2-1 JMP @A

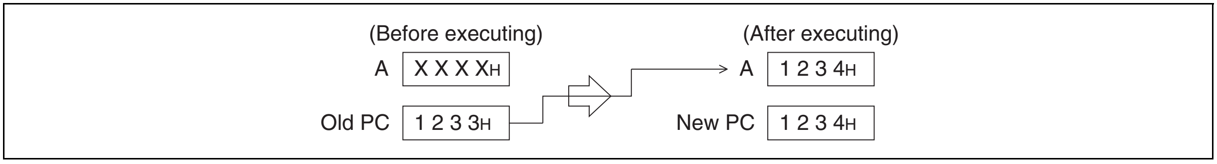


● MOVW A, PC

This instruction works as the opposite of "JMP @A". That is, it stores the content of PC to A. When you have executed this instruction in the main routine and set it to call a specific subroutine, you can make sure that the content of A is the specified value in the subroutine. Also, you can identify that the branch is not from the part that cannot be expected, and use it for the reckless driving judgment.

Figure E.2-2 shows a summary of the instruction.

Figure E.2-2 MOVW A, PC



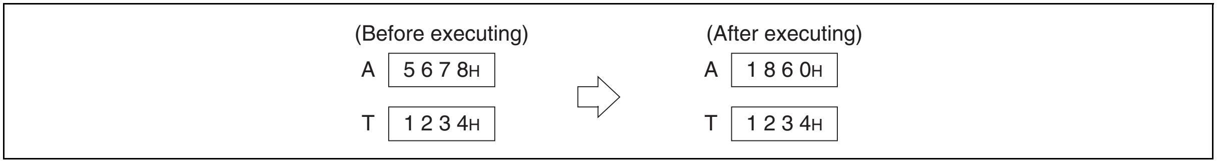
When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-2, the value "1234_H" stored in A corresponds to the address where the following operation code of "MOVW A, PC" is stored.

● MULU A

This instruction performs an unsigned multiplication of AL (lower 8-bit of the accumulator) and TL (lower 8-bit of the temporary accumulator), and stores the 16-bit result in A. The contents of T (temporary accumulator) do not change. The contents of AH (higher 8-bit of the accumulator) and TH (higher 8-bit of the temporary accumulator) before execution of the instruction are not used for the operation. The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a multiplication.

Figure E.2-3 shows a summary of the instruction.

Figure E.2-3 MULU A

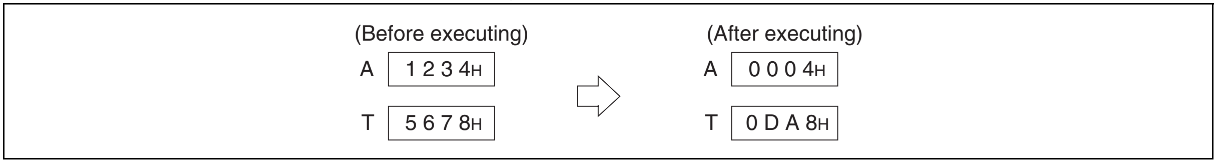


● DIVU A

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

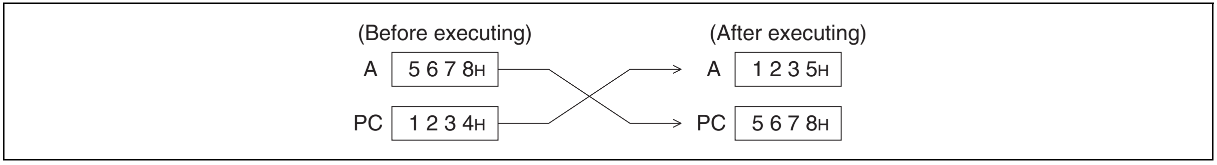


● XCHW A, PC

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.

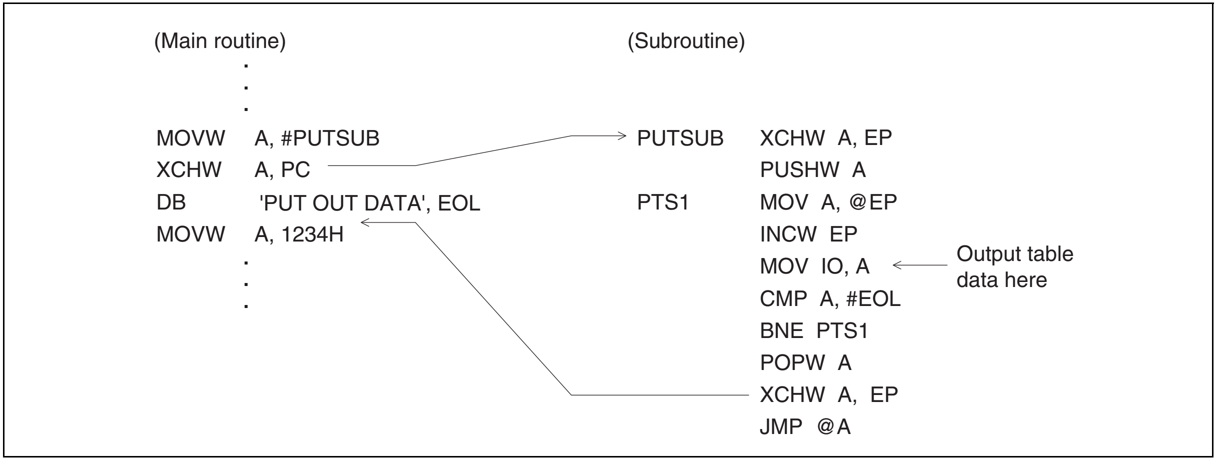
Figure E.2-5 XCHW A, PC



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"



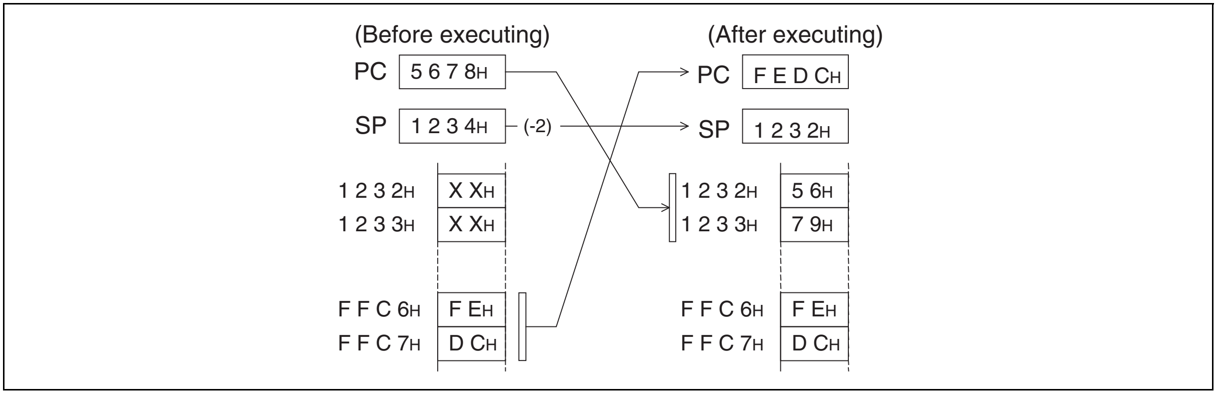
```
(Main routine)                          (Subroutine)
      .
      .
      .
MOVW   A, #PUTSUB                       PUTSUB   XCHW  A, EP
XCHW   A, PC                                     PUSHW  A
DB     'PUT OUT DATA', EOL              PTS1     MOV  A, @EP
MOVW   A, 1234H                                  INCW  EP
      .                                          MOV  IO, A   <---- Output table
      .                                                             data here
      .                                          CMP  A, #EOL
                                                 BNE  PTS1
                                                 POPW  A
                                                 XCHW  A,  EP
                                                 JMP  @A
```

● 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 that are read differently than usual by a bit manipulation instruction.

■ Read-modify-write Operation

By using these bit manipulation instructions, you can set only the specified bit in a register or RAM location to "1" (SETB) or clear to "0" (CLRB). However, as the CPU operates data in 8-bit units, the actual operation (read-modify-write operation) involves a sequence of steps: 8-bit data is read, the specified bit is changed, and the data is written back to the location at the original address.

Table E.3-1 shows bus operation for bit manipulation instructions.

Table E.3-1 Bus Operation for Bit Manipulation Instructions

CODE	MNEMONIC	~	Cycle	Address bus	Data bus	RD	WR	RMW
A0 to A7	CLRB dir:b	4	1	N+2	Next instruction	1	0	1
			2	dir address	Data	1	0	1
A8 to AF	SETB dir:b		3	dir address	Data	0	1	0
			4	N+3	Instruction after next	1	0	0

■ Read Destination on the Execution of Bit Manipulation Instructions

For some I/O ports and the interrupt request flag bits, the read destination differs between a normal read operation and a read-modify-write operation.

● I/O ports (during a bit manipulation)

From some I/O ports, an I/O pin value is read during a normal read operation, while a port data register value is read during a bit manipulation. This prevents the other port data register bits from being changed accidentally, regardless of the I/O directions and states of the pins.

● Interrupt request flag bits (during a bit manipulation)

An interrupt request flag bit functions as a flag bit indicating whether an interrupt request exists during a normal read operation, however, "1" is always read from this bit during a bit manipulation. This prevents the flag from being cleared accidentally by writing the value "0" to the interrupt request flag bit when manipulating another bit.

E.4 F^{2}MC-8FX Instructions

Table E.4-1 to Table E.4-4 show the instructions used by the F^{2}MC-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 Instruction (1/2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	ADDC A, Ri	2	1	(A) ← (A) + (Ri) + C	-	-	-	+	+	+	+	28 to 2F
2	ADDC A, #d8	2	2	(A) ← (A) + d8 + C	-	-	-	+	+	+	+	24
3	ADDC A, dir	3	2	(A) ← (A) + (dir) + C	-	-	-	+	+	+	+	25
4	ADDC A, @IX + off	3	2	(A) ← (A) + ((IX) + off) + C	-	-	-	+	+	+	+	26
5	ADDC A, @EP	2	1	(A) ← (A) + ((EP)) + C	-	-	-	+	+	+	+	27
6	ADDCW A	1	1	(A) ← (A) + (T) + C	-	-	dH	+	+	+	+	23
7	ADDC A	1	1	(AL) ← (AL) + (TL) + C	-	-	-	+	+	+	+	22
8	SUBC A, Ri	2	1	(A) ← (A) - (Ri) - C	-	-	-	+	+	+	+	38 to 3F
9	SUBC A, #d8	2	2	(A) ← (A) - d8 - C	-	-	-	+	+	+	+	34
10	SUBC A, dir	3	2	(A) ← (A) - (dir) - C	-	-	-	+	+	+	+	35
11	SUBC A, @IX + off	3	2	(A) ← (A) - ((IX) + off) - C	-	-	-	+	+	+	+	36
12	SUBC A, @EP	2	1	(A) ← (A) - ((EP)) - C	-	-	-	+	+	+	+	37
13	SUBCW A	1	1	(A) ← (T) - (A) - C	-	-	dH	+	+	+	+	33
14	SUBC A	1	1	(AL) ← (TL) - (AL) - C	-	-	-	+	+	+	+	32
15	INC Ri	3	1	(Ri) ← (Ri) + 1	-	-	-	+	+	+	-	C8 to CF
16	INCW EP	1	1	(EP) ← (EP) + 1	-	-	-	-	-	-	-	C3
17	INCW IX	1	1	(IX) ← (IX) + 1	-	-	-	-	-	-	-	C2
18	INCW A	1	1	(A) ← (A) + 1	-	-	dH	+	+	-	-	C0
19	DEC Ri	3	1	(Ri) ← (Ri) - 1	-	-	-	+	+	+	-	D8 to DF
20	DECW EP	1	1	(EP) ← (EP) - 1	-	-	-	-	-	-	-	D3
21	DECW IX	1	1	(IX) ← (IX) - 1	-	-	-	-	-	-	-	D2
22	DECW A	1	1	(A) ← (A) - 1	-	-	dH	+	+	-	-	D0
23	MULU A	8	1	(A) ← (AL) × (TL)	-	-	dH	-	-	-	-	01
24	DIVU A	17	1	(A) ← (T) / (A) , MOD→ (T)	dL	dH	dH	-	+	-	-	11
25	ANDW A	1	1	(A) ← (A) ∧ (T)	-	-	dH	+	+	R	-	63
26	ORW A	1	1	(A) ← (A) ∨ (T)	-	-	dH	+	+	R	-	73
27	XORW A	1	1	(A) ← (A) ∀ (T)	-	-	dH	+	+	R	-	53
28	CMP A	1	1	(TL) - (AL)	-	-	-	+	+	+	+	12
29	CMPW A	1	1	(T) - (A)	-	-	-	+	+	+	+	13
30	RORC A	1	1	→ C→ A ┐	-	-	-	+	+	-	+	03
31	ROLC A	1	1	┌ C← A ←	-	-	-	+	+	-	+	02
32	CMP A, #d8	2	2	(A) - d8	-	-	-	+	+	+	+	14
33	CMP A, dir	3	2	(A) - (dir)	-	-	-	+	+	+	+	15
34	CMP A, @EP	2	1	(A) - ((EP))	-	-	-	+	+	+	+	17
35	CMP A, @IX + off	3	2	(A) - ((IX) + off)	-	-	-	+	+	+	+	16
36	CMP A, Ri	2	1	(A) - (Ri)	-	-	-	+	+	+	+	18 to 1F
37	DAA	1	1	decimal adjust for addition	-	-	-	+	+	+	+	84
38	DAS	1	1	decimal adjust for subtraction	-	-	-	+	+	+	+	94
39	XOR A	1	1	(A) ← (AL) ∀ (TL)	-	-	-	+	+	R	-	52
40	XOR A, #d8	2	2	(A) ← (AL) ∀ d8	-	-	-	+	+	R	-	54
41	XOR A, dir	3	2	(A) ← (AL) ∀ (dir)	-	-	-	+	+	R	-	55
42	XOR A, @EP	2	1	(A) ← (AL) ∀ ((EP))	-	-	-	+	+	R	-	57
43	XOR A, @IX + off	3	2	(A) ← (AL) ∀ ((IX) + off)	-	-	-	+	+	R	-	56
44	XOR A, Ri	2	1	(A) ← (AL) ∀ (Ri)	-	-	-	+	+	R	-	58 to 5F
45	AND A	1	1	(A) ← (AL) ∧ (TL)	-	-	-	+	+	R	-	62
46	AND A, #d8	2	2	(A) ← (AL) ∧ d8	-	-	-	+	+	R	-	64
47	AND A, dir	3	2	(A) ← (AL) ∧ (dir)	-	-	-	+	+	R	-	65
48	AND A, @EP	2	1	(A) ← (AL) ∧ ((EP))	-	-	-	+	+	R	-	67
49	AND A, @IX + off	3	2	(A) ← (AL) ∧ ((IX) + off)	-	-	-	+	+	R	-	66

Table E.4-2 Arithmetic Operation Instruction (1/2)

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
50	AND	A, Ri	2	1	(A) ← (AL) ∧ (Ri)	-	-	-	+	+	R	-	68 to 6F
51	OR	A	1	1	(A) ← (AL) ∨ (TL)	-	-	-	+	+	R	-	72
52	OR	A, #d8	2	2	(A) ← (AL) ∨ d8	-	-	-	+	+	R	-	74
53	OR	A, dir	3	2	(A) ← (AL) ∨ (dir)	-	-	-	+	+	R	-	75
54	OR	A, @EP	2	1	(A) ← (AL) ∨ ((EP))	-	-	-	+	+	R	-	77
55	OR	A, @IX + off	3	2	(A) ← (AL) ∨ ((IX) + off)	-	-	-	+	+	R	-	76
56	OR	A, Ri	2	1	(A) ← (AL) ∨ (Ri)	-	-	-	+	+	R	-	78 to 7F
57	CMP	dir, #d8	4	3	(dir) - d8	-	-	-	+	+	+	+	95
58	CMP	@EP, #d8	3	2	((EP)) - d8	-	-	-	+	+	+	+	97
59	CMP	@IX + off, #d8	4	3	((IX) + off) - d8	-	-	-	+	+	+	+	96
60	CMP	Ri, #d8	3	2	(Ri) - d8	-	-	-	+	+	+	+	98 to 9F
61	INCW	SP	1	1	(SP) ← (SP) + 1	-	-	-	-	-	-	-	C1
62	DECW	SP	1	1	(SP) ← (SP) - 1	-	-	-	-	-	-	-	D1

■ Branch Instructions

Table E.4-3 Branch Instructions

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	BZ/BEQ	rel(at branch)	4	2	if Z = 1 then PC← PC + rel	-	-	-	-	-	-	-	FD
	BZ/BEQ	rel(at no branch)	2										
2	BNZ/BNE	rel(at branch)	4	2	if Z = 0 then PC← PC + rel	-	-	-	-	-	-	-	FC
	BNZ/BNE	rel(at no branch)	2										
3	BC/BLO	rel(at branch)	4	2	if C = 1 then PC← PC + rel	-	-	-	-	-	-	-	F9
	BC/BLO	rel(at no branch)	2										
4	BNC/BHS	rel(at branch)	4	2	if C = 0 then PC← PC + rel	-	-	-	-	-	-	-	F8
	BNC/BHS	rel(at no branch)	2										
5	BN	rel(at branch)	4	2	if N = 1 then PC← PC + rel	-	-	-	-	-	-	-	FB
	BN	rel(at no branch)	2										
6	BP	rel(at branch)	4	2	if N = 0 then PC← PC + rel	-	-	-	-	-	-	-	FA
	BP	rel(at no branch)	2										
7	BLT	rel(at branch)	4	2	if V ∀ N = 1 then PC← PC + rel	-	-	-	-	-	-	-	FF
	BLT	rel(at no branch)	2										
8	BGE	rel(at branch)	4	2	if V ∀ N = 0 then PC← PC + rel	-	-	-	-	-	-	-	FE
	BGE	rel(at no branch)	2										
9	BBC	dir : b, rel	5	3	if (dir : b) = 0 then PC← PC + rel	-	-	-	-	+	-	-	B0 to B7
10	BBS	dir : b, rel	5	3	if (dir : b) = 1 then PC← PC + rel	-	-	-	-	+	-	-	B8 to BF
11	JMP	@A	3	1	(PC) ← (A)	-	-	-	-	-	-	-	E0
12	JMP	ext	4	3	(PC) ← ext	-	-	-	-	-	-	-	21
13	CALLV	#vct	7	1	vector call	-	-	-	-	-	-	-	E8 to EF
14	CALL	ext	6	3	subroutine call	-	-	-	-	-	-	-	31
15	XCHW	A, PC	3	1	(PC) ← (A) , (A) ← (PC) + 1	-	-	dH	-	-	-	-	F4
16	RET		6	1	return from subroutine	-	-	-	-	-	-	-	20
17	RETI		8	1	return from interrupt	-	-	-		restore			30

■ Other Instructions

Table E.4-4 Other Instructions

No.	MNEMONIC		~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	PUSHW	A	4	1	((SP))← (A), (SP)← (SP) - 2	-	-	-	-	-	-	-	40
2	POPW	A	3	1	(A)← ((SP)), (SP)← (SP) + 2	-	-	dH	-	-	-	-	50
3	PUSHW	IX	4	1	((SP))← (IX), (SP)← (SP) - 2	-	-	-	-	-	-	-	41
4	POPW	IX	3	1	(IX)← ((SP)), (SP)← (SP) + 2	-	-	-	-	-	-	-	51
5	NOP		1	1	No operation	-	-	-	-	-	-	-	00
6	CLRC		1	1	(C)← 0	-	-	-	-	-	-	R	81
7	SETC		1	1	(C)← 1	-	-	-	-	-	-	S	91
8	CLRI		1	1	(I)← 0	-	-	-	-	-	-	-	80
9	SETI		1	1	(I)← 1	-	-	-	-	-	-	-	90

E.5 Instruction Map

Table E.5-1 shows the instruction map of F^2MC-8FX.

■ Instruction Map

Table E.5-1 Instruction Map of F^2MC-8FX

L \ H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A, ext	MOVW A, PS	CLRI	SETI	CLRB dir : 0	BBC dir : 0, rel	INCW A	DECW A	JMP @A	MOVW A, PC
1	MULU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV ext, A	MOVW PS, A	CLRC	SETC	CLRB dir : 1	BBC dir : 1, rel	INCW SP	DECW SP	MOVW SP, A	MOVW A, SP
2	ROLC A	CMP A	ADDC A	SUBC A	XCH A, T	XOR A	AND A	OR A	MOV @A, T	MOV A, @A	CLRB dir : 2	BBC dir : 2, rel	INCW IX	DECW IX	MOVW IX, A	MOVW A, IX
3	RORC A	CMPW A	ADDCW A	SUBCW A	XCHW A, T	XORW A	ANDW A	ORW A	MOVW @A, T	MOVW A, @A	CLRB dir : 3	BBC dir : 3, rel	INCW EP	DECW EP	MOVW EP, A	MOVW A, EP
4	MOV A, #d8	CMP A, #d8	ADDC A, #d8	SUBC A, #d8		XOR A, #d8	AND A, #d8	OR A, #d8	DAA	DAS	CLRB dir : 4	BBC dir : 4, rel	MOVW A, ext	MOVW ext, A	MOVW A, #d16	XCHW A, PC
5	MOV A, dir	CMP A, dir	ADDC A, dir	SUBC A, dir	MOV dir, A	XOR A, dir	AND A, dir	OR A, dir	MOV dir, #d8	CMP dir, #d8	CLRB dir : 5	BBC dir : 5, rel	MOVW A, dir	MOVW dir, A	MOVW SP, #d16	XCHW A, SP
6	MOV A, @IX+d	CMP A, @IX+d	ADDC A, @IX+d	SUBC A, @IX+d	MOV @IX+d, A	XOR A, @IX+d	AND A, @IX+d	OR A, @IX+d	MOV @IX+d,#d8	CMP @IX+d,#d8	CLRB dir : 6	BBC dir : 6, rel	MOVW A, @IX+d	MOVW @IX+d, A	MOVW IX, #d16	XCHW A, IX
7	MOV A, @EP	CMP A, @EP	ADDC A, @EP	SUBC A, @EP	MOV @EP, A	XOR A, @EP	AND A, @EP	OR A, @EP	MOV @EP, #d8	CMP @EP, #d8	CLRB dir : 7	BBC dir : 7, rel	MOVW A, @EP	MOVW @EP, A	MOVW EP, #d16	XCHW A, EP
8	MOV A, R0	CMP A, R0	ADDC A, R0	SUBC A, R0	MOV R0, A	XOR A, R0	AND A, R0	OR A, R0	MOV R0, #d8	CMP R0, #d8	SETB dir : 0	BBS dir : 0, rel	INC R0	DEC R0	CALLV #0	BNC rel
9	MOV A, R1	CMP A, R1	ADDC A, R1	SUBC A, R1	MOV R1, A	XOR A, R1	AND A, R1	OR A, R1	MOV R1, #d8	CMP R1, #d8	SETB dir : 1	BBS dir : 1, rel	INC R1	DEC R1	CALLV #1	BC rel
A	MOV A, R2	CMP A, R2	ADDC A, R2	SUBC A, R2	MOV R2, A	XOR A, R2	AND A, R2	OR A, R2	MOV R2, #d8	CMP R2, #d8	SETB dir : 2	BBS dir : 2, rel	INC R2	DEC R2	CALLV #2	BP rel
B	MOV A, R3	CMP A, R3	ADDC A, R3	SUBC A, R3	MOV R3, A	XOR A, R3	AND A, R3	OR A, R3	MOV R3, #d8	CMP R3, #d8	SETB dir : 3	BBS dir : 3, rel	INC R3	DEC R3	CALLV #3	BN rel
C	MOV A, R4	CMP A, R4	ADDC A, R4	SUBC A, R4	MOV R4, A	XOR A, R4	AND A, R4	OR A, R4	MOV R4, #d8	CMP R4, #d8	SETB dir : 4	BBS dir : 4, rel	INC R4	DEC R4	CALLV #4	BNZ rel
D	MOV A, R5	CMP A, R5	ADDC A, R5	SUBC A, R5	MOV R5, A	XOR A, R5	AND A, R5	OR A, R5	MOV R5, #d8	CMP R5, #d8	SETB dir : 5	BBS dir : 5, rel	INC R5	DEC R5	CALLV #5	BZ rel
E	MOV A, R6	CMP A, R6	ADDC A, R6	SUBC A, R6	MOV R6, A	XOR A, R6	AND A, R6	OR A, R6	MOV R6, #d8	CMP R6, #d8	SETB dir : 6	BBS dir : 6, rel	INC R6	DEC R6	CALLV #6	BGE rel
F	MOV A, R7	CMP A, R7	ADDC A, R7	SUBC A, R7	MOV R7, A	XOR A, R7	AND A, R7	OR A, R7	MOV R7, #d8	CMP R7, #d8	SETB dir : 7	BBS dir : 7, rel	INC R7	DEC R7	CALLV #7	BLT rel

MB95430H Series
APPENDIX F   Mask Options

The mask option list of the MB95430H Series is shown in Table F-1.

■ Mask Option List

Table F-1 Mask Option List

No.	Part Number	MB95F432H MB95F433H MB95F434H	MB95F432K MB95F433K MB95F434K
	Selectable/Fixed	Fixed	
1	Low voltage detection reset	Without low-voltage detection reset	With low-voltage detection reset
2	Reset	With dedicated reset input	Without dedicated reset input

MB95430H Series

Register Index

MB95430H Series

Pin Function Index

MB95430H Series

Interrupt Vector Index

CM26-101xx-1E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

New 8FX

8-BIT MICROCONTROLLER

MB95430H Series

HARDWARE MANUAL

March 2011 the first edition

Published **FUJITSU SEMICONDUCTOR LIMITED**

Edited Sales Promotion Department