F²MC-8FX

8-BIT MICROCONTROLLER

MB95330H Series

HARDWARE MANUAL

FUJITSU

F²MC-8FX

8-BIT MICROCONTROLLER

MB95330H Series
HARDWARE MANUAL

For the information for microcontroller supports, see the following web site.

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 MB95330H Series is a line of products developed as general-purpose products in the F^2MC-8FX family of proprietary 8-bit single-chip microcontrollers applicable as application-specific integrated circuits (ASICs). The MB95330H 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 MB95330H Series of microcontrollers, this manual describes its functions, features, and operations. You should read through the manual.

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

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

■ Trademark

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^2MC-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.

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 MB95330H Series.

1.1 Features of MB95330H Series

In addition to a compact instruction set, the MB95330H is a series of general-purpose single-chip microcontrollers with a variety of peripheral functions.

■ Features of MB95330H Series

● F^2MC-8FX CPU core

Instruction set optimized for controllers

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

● Clock

- Selectable main clock source
  Main OSC clock (Up to 16.25 MHz, maximum machine clock frequency is 8.125 MHz)
  External clock (Up to 32.5 MHz, maximum machine clock frequency is 16.25 MHz)
  Main 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 × 2 channels
- 8/16-bit PPG × 3 channels
- 16-bit PPG × 1 channel (can work independently or together with the multi-pulse generator)
- 16-bit reload timer × 1 channel (can work independently or together with the multi-pulse generator)
- Time-base timer × 1 channel
- Watch prescaler × 1 channel

● UART/SIO

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

● I^2C

- Built-in wake-up function

MB95330H Series

● Multi-pulse generator (MPG)

- 16-bit reload timer × 1 channel
- 16-bit PPG timer × 1 channel
- Waveform sequencer (including a 16-bit timer equipped with a buffer and a compare clear function)

● LIN-UART

- Full duplex double buffer
- Capable of clock-synchronous serial data transfer and clock-asynchronous serial data transfer

● External interrupt

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

● 8/10-bit A/D converter

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

● Low power consumption modes (standby modes)

- Stop mode
- Sleep mode
- Watch mode
- Time-base timer mode

● I/O port

- MB95F332H/F333H/F334H (maximum no. of I/O ports: 28)
  - General-purpose I/O ports (N-ch open drain)　: 3
  - General-purpose I/O ports (CMOS I/O)　　　: 25
- MB95F332K/F333K/F334K (maximum no. of I/O ports: 29)
  - General-purpose I/O ports (N-ch open drain)　: 4
  - General-purpose I/O ports (CMOS I/O)　　　: 25

● On-chip debug

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

● Hardware/software watchdog timer

- Built-in hardware watchdog timer

● Low-voltage detection reset circuit

- Built-in low-voltage detector

● Clock supervisor counter

- Built-in clock supervisor counter function

● Programmable port input voltage level

- CMOS input level / hysteresis input level

● 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

1.2 Product Line-up of MB95330H Series

Table 1.2-1 lists the product line-up of the MB95330H Series.

■ **Product Line-up of MB95330H Series**

Table 1.2-1 Product Line-up of MB95330H Series (1 / 2)

Part number / Parameter	MB95F332H	MB95F333H	MB95F334H	MB95F332K	MB95F333K	MB95F334K
Type	Flash memory product					
Clock supervisor counter	It supervises the main clock oscillation.					
Program ROM capacity	8 Kbyte	12 Kbyte	20 Kbyte	8 Kbyte	12 Kbyte	20 Kbyte
RAM capacity	240 bytes	496 bytes	1008 bytes	240 bytes	496 bytes	1008 bytes
Low-voltage detection reset	No			Yes		
Reset input	Dedicated			Selected by 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 = 16.25 MHz) Interrupt processing time : 0.6 μs (with machine clock = 16.25 MHz)					
General-purpose I/O	I/O ports (Max): 28 CMOS I/O: 25, N-ch open drain: 3			I/O ports (Max): 29 CMOS I/O: 25, N-ch open drain: 4		
Time-base timer	Interrupt cycle: 0.256 ms to 8.3 s (when external clock = 4 MHz)					
Hardware/ software watchdog timer	Reset generation cycle Main oscillation clock at 10 MHz: 105 ms (Min) The sub-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.					
LIN-UART	A wide range of communication speeds can be selected by a dedicated reload timer. Clock-synchronous serial data transfer and clock-asynchronous serial data transfer is enabled. The LIN function can be used as a LIN master or a LIN slave.					
8/10-bit A/D converter	8 channels					
	8-bit resolution and 10-bit resolution can be chosen.					
8/16-bit composite timer	2 channels					
	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	10 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 modes.					
On-chip debug	1-wire serial control It supports serial writing. (asynchronous mode)					

Table 1.2-1 Product Line-up of MB95330H Series (2 / 2)

Part number / Parameter	MB95F332H	MB95F333H	MB95F334H	MB95F332K	MB95F333K	MB95F334K
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.						
I^2C	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.						
8/16-bit PPG	3 channels Each channel of PPG can be used as two 8-bit PPG channels or a single 16-bit PPG channel. The counter operating clock can be selected from eight clock sources.						
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. It can work independently or together with the multi-pulse generator.						
16-bit reload timer	Two clock modes and two counter operating modes are available to use. It can output square waveform. Count clock: it can be selected from internal clocks (seven types) and external clocks. Two counter operating modes: reload mode and one-shot mode It can work independently or together with the multi-pulse generator.						
Multi-pulse generator	16-bit PPG timer: 1 channel 16-bit reload timer operations: toggle output, one-shot output Event counter: 1 channel Waveform sequencer (including a 16-bit timer equipped with a buffer and a compare clear function)						
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 write/erase 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 LCC-32P-M19						

MB95330H Series

1.3 Differences among Products and Notes on Product Selection

---

The following describes differences among the products of the MB95330H 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 MB95330H Series.

- Package

  For details of information on each package, see Section 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 MB95330H 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.

1.4 Block Diagram of MB95330H Series

Figure 1.4-1 is the block diagram of the MB95330H Series.

■ Block Diagrams of MB95330H Series

Figure 1.4-1 Block Diagram of MB95330H Series
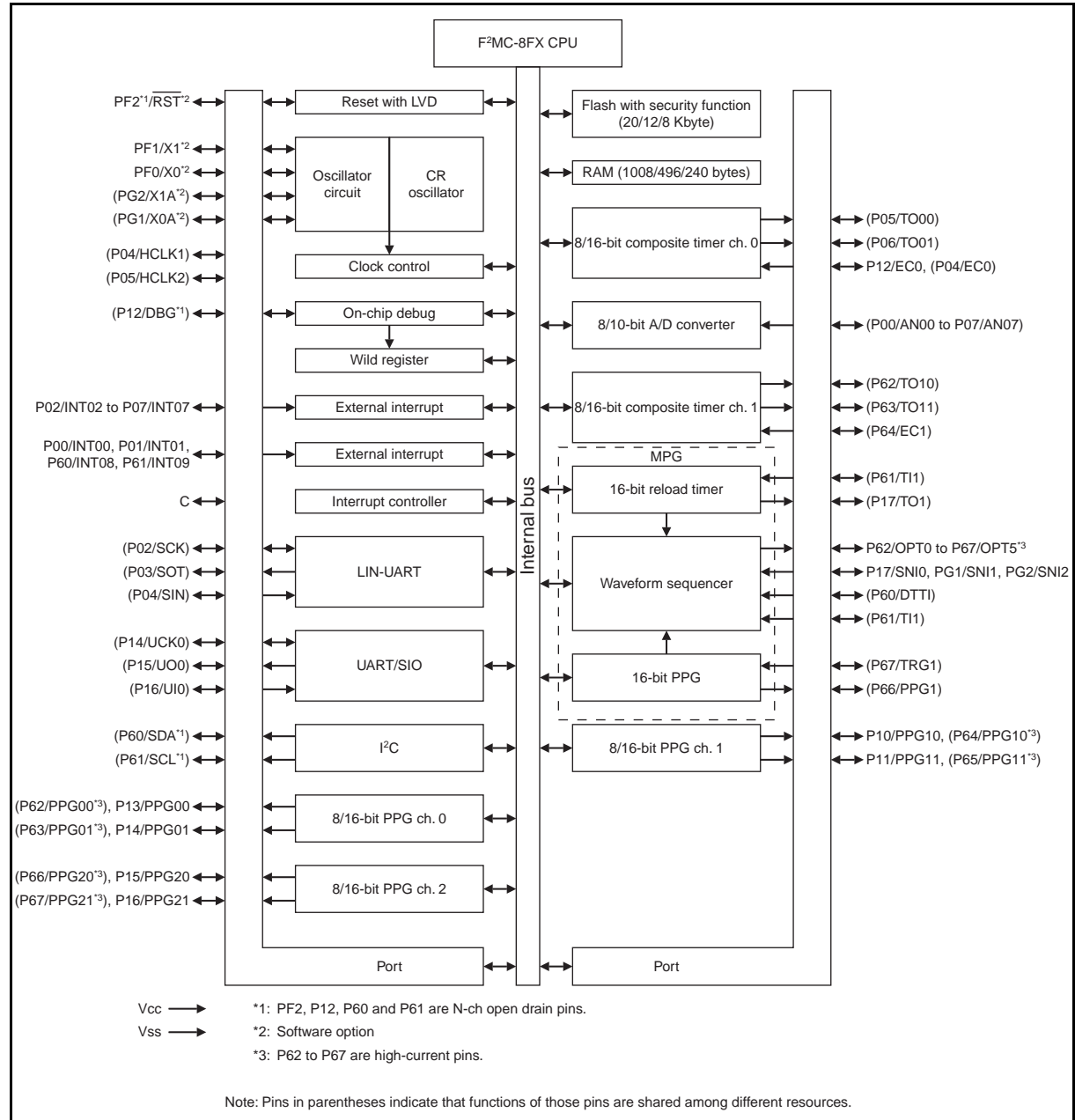
MB95330H Series

1.5 Pin Assignment

Figure 1.5-1, Figure 1.5-2 and Figure 1.5-3 show the respective pin assignments in the three packages of the MB95330H Series.

■ **Pin Assignment of FPT-32P-M30**

Figure 1.5-1 Pin Assignment of FPT-32P-M30
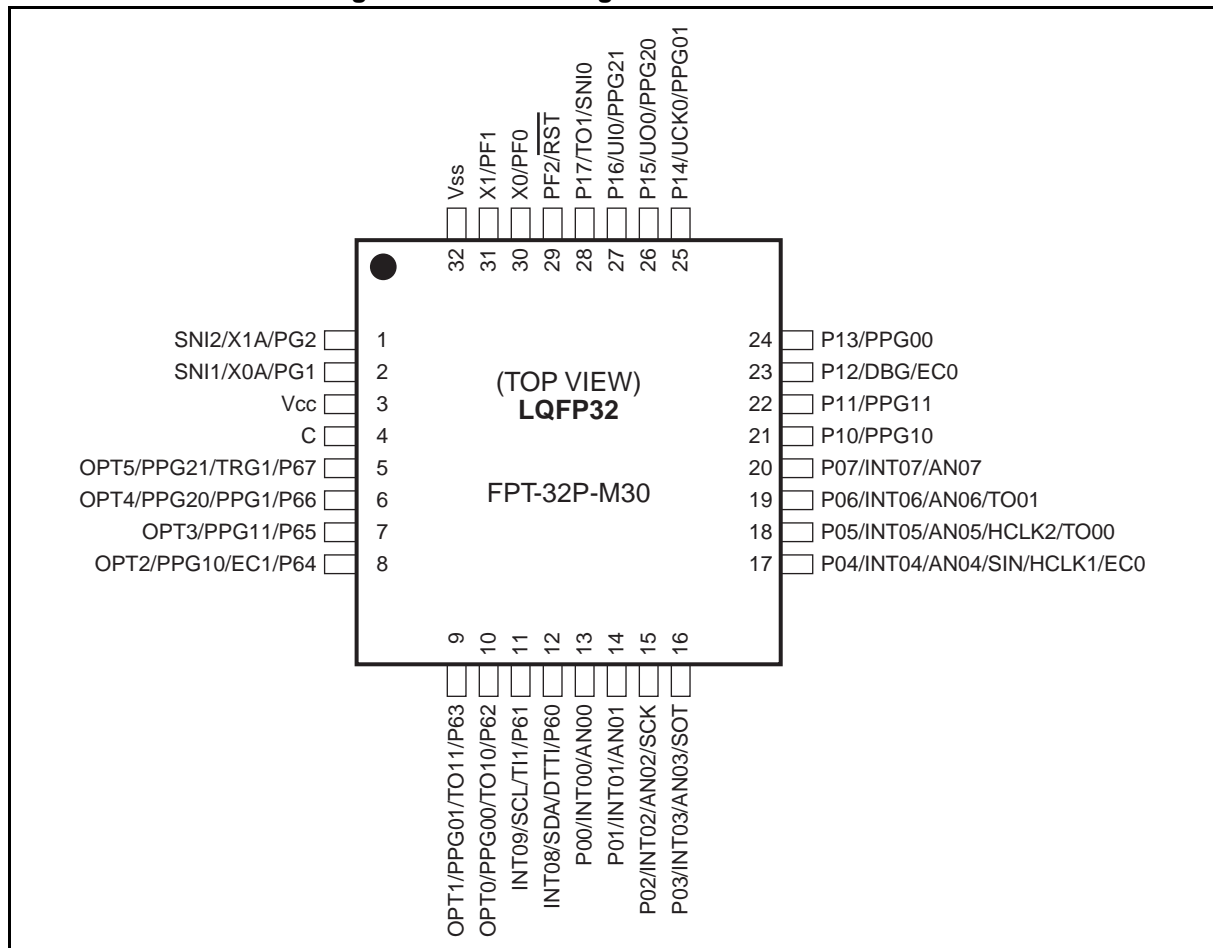
■ **Pin Assignment of DIP-32P-M06**

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

Pin	Signal		Pin	Signal
1	PF2/$\overline{RST}$		32	P17/TO1/SNI0
2	X0/PF0		31	P16/UI0/PPG21
3	X1/PF1		30	P15/UO0/PPG20
4	Vss		29	P14/UCK0/PPG01
5	SNI2/X1A/PG2		28	P13/PPG00
6	SNI1/X0A/PG1		27	P12/DBG/EC0
7	Vcc	(TOP VIEW) **SH-DIP32**	26	P11/PPG11
8	C		25	P10/PPG10
9	OPT5/PPG21/TRG1/P67	DIP-32P-M06	24	P07/INT07/AN07
10	OPT4/PPG20/PPG1/P66		23	P06/INT06/AN06/TO01
11	OPT3/PPG11/P65		22	P05/INT05/AN05/HCLK2/TO00
12	OPT2/PPG10/EC1/P64		21	P04/INT04/AN04/SIN/HCLK1/EC0
13	OPT1/PPG01/TO11/P63		20	P03/INT03/AN03/SOT
14	OPT0/PPG00/TO10/P62		19	P02/INT02/AN02/SCK
15	INT09/SCL/TI1/P61		18	P01/INT01/AN01
16	INT08/SDA/DTTI/P60		17	P00/INT00/AN00

MB95330H Series

■ Pin Assignment of LCC-32P-M19

Figure 1.5-3 Pin Assignment of LCC-32P-M19
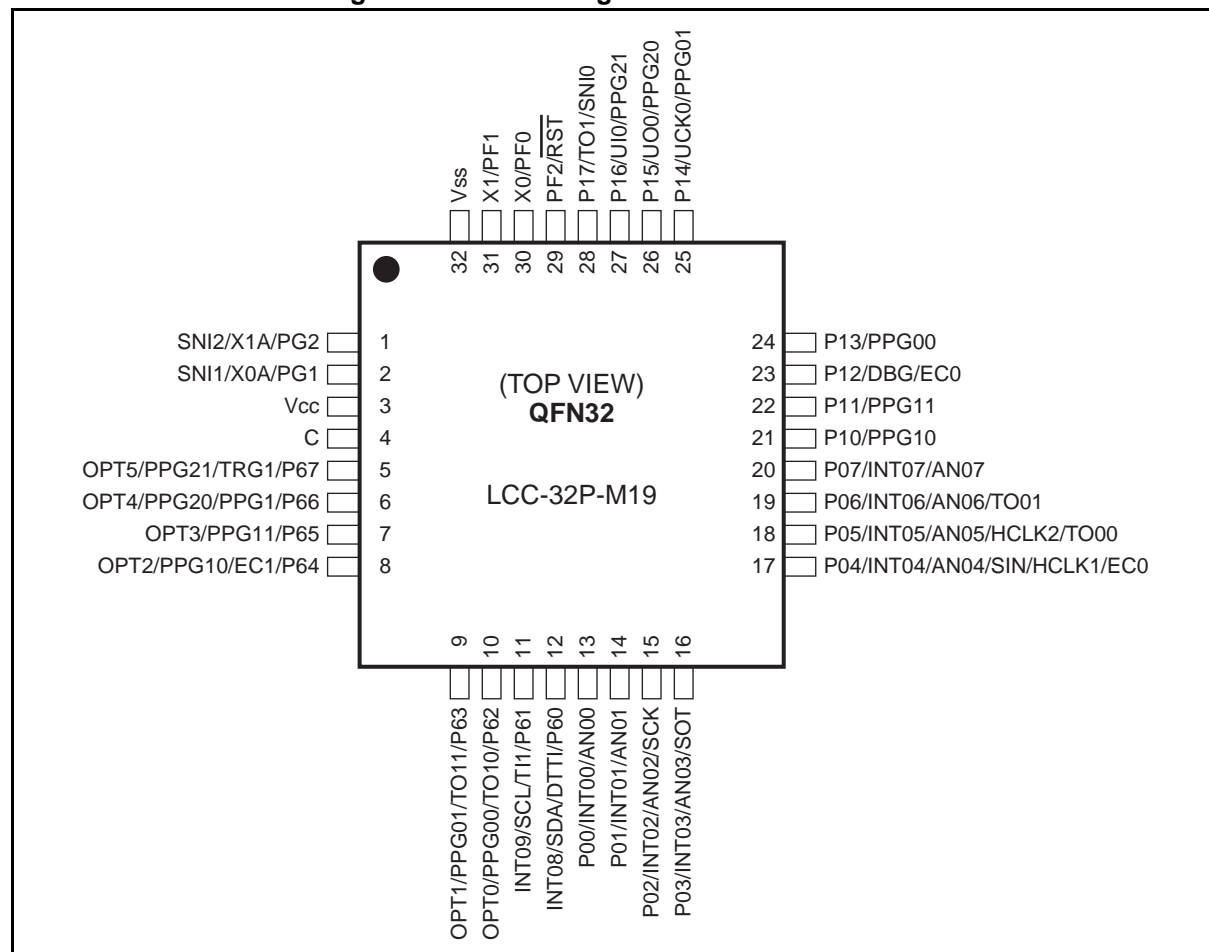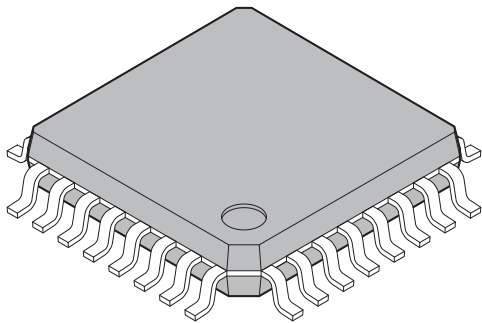
1.6 Package Dimension

The MB95330H Series is available in three types of package.

■ **Package Dimension of FPT-32P-M30**

Figure 1.6-1 Package Dimension of FPT-32P-M30

32-pin plastic LQFP		
	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
(FPT-32P-M30)		
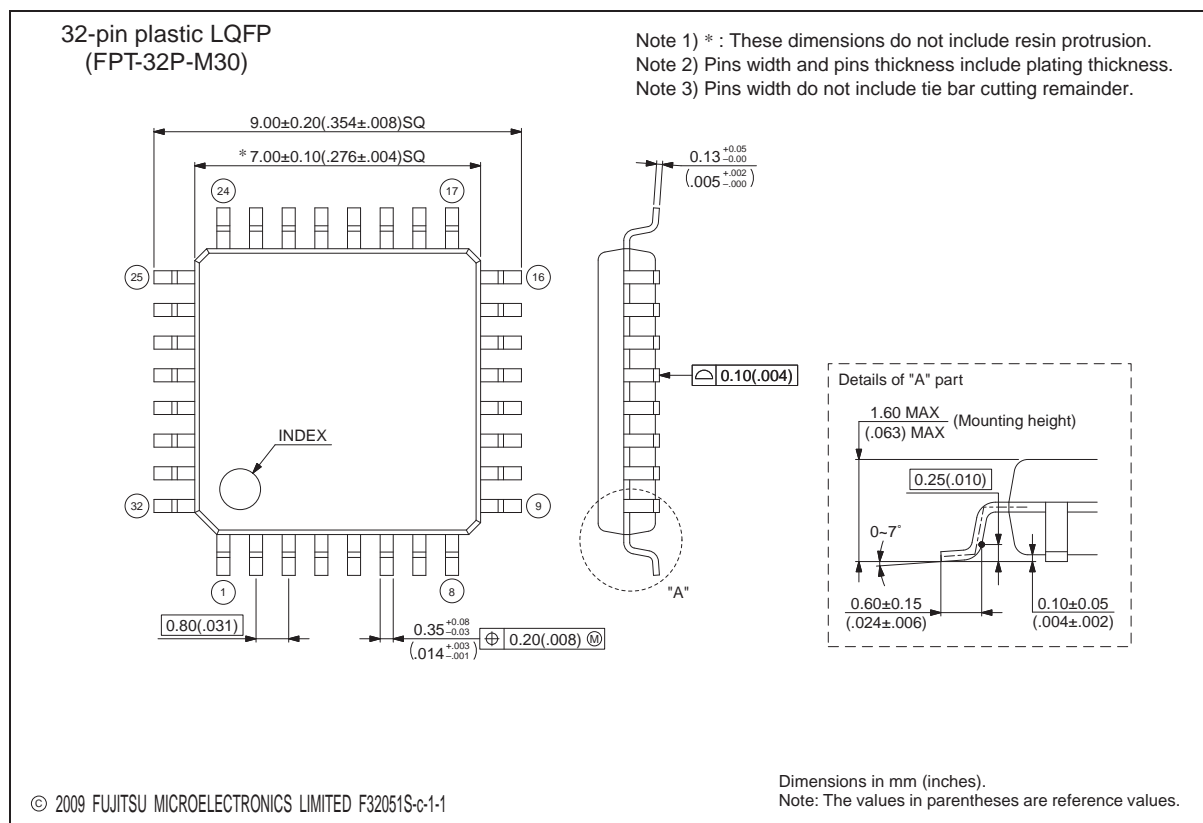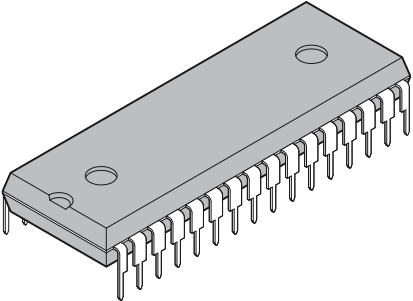
Note 1) ∗ : These dimensions do not include resin protrusion.
Note 2) Pins width and pins thickness include plating thickness.
Note 3) Pins width do not include tie bar cutting remainder.

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

© 2009 FUJITSU MICROELECTRONICS LIMITED F32051S-c-1-1

Please check the latest package dimension at the following URL.

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

MB95330H Series

■ Package Dimension of DIP-32P-M06

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

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

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

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

$*28.00^{+0.20}_{-0.30}$
$\left(1.102^{+.008}_{-.012}\right)$

INDEX

$*8.89\pm0.25$
$(.350\pm.010)$

$1.02^{+0.30}_{-0.20}$
$\left(.040^{+.012}_{-.008}\right)$

$4.70^{+0.70}_{-0.20}$
$\left(.185^{+.028}_{-.008}\right)$

0.51(.020)
MIN.

$3.30^{+0.20}_{-0.30}$
$\left(.130^{+.008}_{-.012}\right)$

1.27(.050)
MAX.

1.778(.070)

$0.48^{+0.08}_{-0.12}$
$\left(.019^{+.003}_{-.005}\right)$

⊕ 0.25(.010) Ⓜ

$0.27^{+0.03}_{-0.07}$
$\left(.011^{+.001}_{-.003}\right)$

10.16(.400)

0~15°

© 2003-2008 FUJITSU MICROELECTRONICS LIMITED D32018S-c-1-2

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 LCC-32P-M19**

Figure 1.6-3 Package Dimension of LCC-32P-M19

32-pin plastic QFN	Lead pitch	0.50 mm
	Package width × package length	5.00 mm × 5.00 mm
	Sealing method	Plastic mold
	Mounting height	0.80 mm MAX
	Weight	0.06 g
(LCC-32P-M19)		

32-pin plastic QFN
 (LCC-32P-M19)

5.00±0.10
(.197±.004)

5.00±0.10
(.197±.004)

INDEX AREA

3.50±0.10
(.138±.004)

3.50±0.10
(.138±.004)

0.25 +0.05 −0.07
(.010 +.002 −.003)

(3-R0.20)
((3-R.008))

0.40±0.05
(.016±.002)

1PIN CORNER
(C0.30(C.012))

0.50(.020)
(TYP)

0.75±0.05
(.030±.002)

0.02 +0.03 −0.02
(.001 +.001 −.001)

(0.20(.008))

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

© 2009 FUJITSU MICROELECTRONICS LIMITED C32071S-c-1-1

Please check the latest package dimension at the following URL.

http://edevice.fujitsu.com/package/en-search/
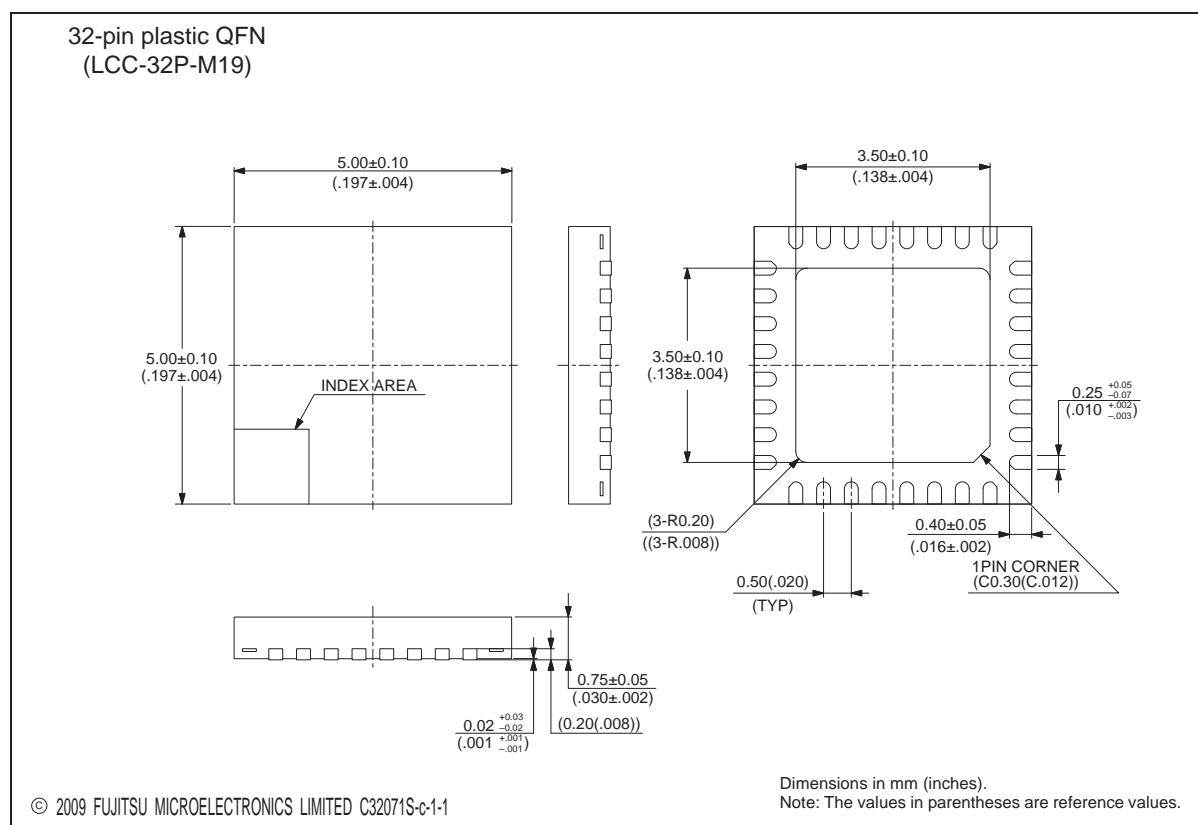
MB95330H Series

1.7 Pin Descriptions

Table 1.7-1 shows pin descriptions of the MB95330H Series. The alphabets in "I/O circuit type" column of the above table correspond to those in "Type" column of Table 1.8-1.

■ **Pin Descriptions**

Table 1.7-1 Pin Descriptions (1 / 4)

Pin no.		Pin name	I/O circuit type[4]	Function
LQFP32[1] & QFN32[2]	SH-DIP32[3]			
1	5	PG2	C	General-purpose I/O port
		X1A		Subclock I/O oscillation pin
		SNI2		Trigger input pin for the position detection function of the MPG waveform sequencer
2	6	PG1	C	General-purpose I/O port
		X0A		Subclock input oscillation pin
		SNI1		Trigger input pin for the position detection function of the MPG waveform sequencer
3	7	V_{CC}	—	Power supply pin
4	8	C	—	Capacitor connection pin
5	9	P67	D	General-purpose I/O port High-current pin
		PPG21		8/16-bit PPG ch. 2 output pin
		TRG1		16-bit PPG ch. 1 trigger input pin
		OPT5		MPG waveform sequencer output pin
6	10	P66	D	General-purpose I/O port High-current pin
		PPG20		8/16-bit PPG ch. 2 output pin
		PPG1		16-bit PPG ch. 1 output pin
		OPT4		MPG waveform sequencer output pin
7	11	P65	D	General-purpose I/O port High-current pin
		PPG11		8/16-bit PPG ch. 1 output pin
		OPT3		MPG waveform sequencer output pin

Table 1.7-1 Pin Descriptions (2 / 4)

Pin no.		Pin name	I/O circuit type[4]	Function
LQFP32[1] & QFN32[2]	SH-DIP32[3]			
8	12	P64	D	General-purpose I/O port High-current pin
		EC1		8/16-bit composite timer ch. 1 clock input pin
		PPG10		8/16-bit PPG ch. 1 output pin
		OPT2		MPG waveform sequencer output pin
9	13	P63	D	General-purpose I/O port High-current pin
		TO11		8/16-bit composite timer ch. 1 output pin
		PPG01		8/16-bit PPG ch. 0 output pin
		OPT1		MPG waveform sequencer output pin
10	14	P62	D	General-purpose I/O port High-current pin
		TO10		8/16-bit composite timer ch. 1 output pin
		PPG00		8/16-bit PPG ch. 0 output pin
		OPT0		MPG waveform sequencer output pin
11	15	P61	I	General-purpose I/O port
		INT09		External interrupt input pin
		SCL		I^2C clock I/O pin
		TI1		16-bit reload timer ch. 1 input pin
12	16	P60	I	General-purpose I/O port
		INT08		External interrupt input pin
		SDA		I^2C data I/O pin
		DTTI		MPG waveform sequencer input pin
13	17	P00	E	General-purpose I/O port
		INT00		External interrupt input pin
		AN00		A/D converter analog input pin
14	18	P01	E	General-purpose I/O port
		INT01		External interrupt input pin
		AN01		A/D converter analog input pin

MB95330H Series

Table 1.7-1 Pin Descriptions (3 / 4)

Pin no.		Pin name	I/O circuit type[4]	Function
LQFP32[1] & QFN32[2]	SH-DIP32[3]			
15	19	P02	E	General-purpose I/O port
		INT02		External interrupt input pin
		AN02		A/D converter analog input pin
		SCK		LIN-UART clock I/O pin
16	20	P03	E	General-purpose I/O port
		INT03		External interrupt input pin
		AN03		A/D converter analog input pin
		SOT		LIN-UART data output pin
17	21	P04	F	General-purpose I/O port
		INT04		External interrupt input pin
		AN04		A/D converter analog input pin
		SIN		LIN-UART data input pin
		HCLK1		External clock input pin
		EC0		8/16-bit composite timer ch. 0 clock input pin
18	22	P05	E	General-purpose I/O port
		INT05		External interrupt input pin
		AN05		A/D converter analog input pin
		HCLK2		External clock input pin
		TO00		8/16-bit composite timer ch. 0 output pin
19	23	P06	E	General-purpose I/O port
		INT06		External interrupt input pin
		AN06		A/D converter analog input pin
		TO01		8/16-bit composite timer ch. 0 output pin
20	24	P07	E	General-purpose I/O port
		INT07		External interrupt input pin
		AN07		A/D converter analog input pin
21	25	P10	G	General-purpose I/O port
		PPG10		8/16-bit PPG ch. 1 output pin
22	26	P11	G	General-purpose I/O port
		PPG11		8/16-bit PPG ch. 1 output pin

Table 1.7-1 Pin Descriptions (4 / 4)

Pin no.		Pin name	I/O circuit type[4]	Function
LQFP32[1] & QFN32[2]	SH-DIP32[3]			
23	27	P12	H	General-purpose I/O port
		DBG		DBG input pin
		EC0		8/16-bit composite timer ch. 0 clock input pin
24	28	P13	G	General-purpose I/O port
		PPG00		8/16-bit PPG ch. 0 output pin
25	29	P14	G	General-purpose I/O port
		UCK0		UART/SIO ch. 0 clock I/O pin
		PPG01		8/16-bit PPG ch. 0 output pin
26	30	P15	G	General-purpose I/O port
		UO0		UART/SIO ch. 0 data output pin
		PPG20		8/16-bit PPG ch. 2 output pin
27	31	P16	J	General-purpose I/O port
		UI0		UART/SIO ch. 0 data input pin
		PPG21		8/16-bit PPG ch. 2 output pin
28	32	P17	G	General-purpose I/O port
		TO1		16-bit reload timer ch. 1 output pin
		SNI0		Trigger input pin for the position detection function of the MPG waveform sequencer
29	1	PF2	A	General-purpose I/O port
		$\overline{\text{RST}}$		Reset pin Dedicated reset pin in MB95F332H/F333H/F334H
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: LCC-32P-M19
*3: Package code: DIP-32P-M06
*4: For the I/O circuit types, see Section 1.8 "I/O Circuit Types".

MB95330H Series

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 / 3)
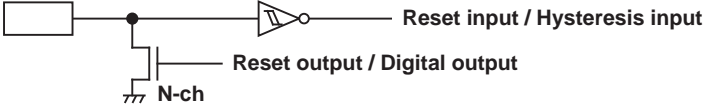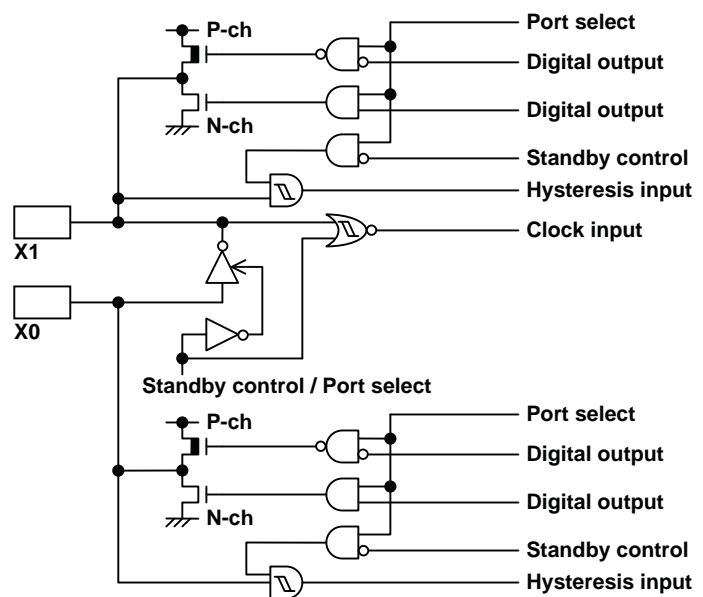
Type	Circuit	Remarks
A	Reset input / Hysteresis input Reset output / Digital output N-ch	• N-ch open drain output • Hysteresis input • Reset output
B	P-ch N-ch Port select Digital output Digital output Standby control Hysteresis input Clock input X1 X0 Standby control / Port select P-ch N-ch Port select 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 / 3)

Type	Circuit	Remarks

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

D		• CMOS output • Hysteresis input
E		• CMOS output • Hysteresis input • Pull-up control available

MB95330H Series

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

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

CHAPTER 2
NOTES ON DEVICE HANDLING

This chapter provides notes on using the MB95330H Series.

2.1   Notes on Device Handling

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 MB95330H 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.

- $\overline{RST}$ pin

  Connect the $\overline{RST}$ 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 $\overline{RST}$ pin and the V_{CC} or V_{SS} pin when designing the layout of the printed circuit board.

  The $\overline{RST}$/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 SYSC register, and the reset input function or the general purpose I/O function can be selected by the RSTEN bit in the SYSC register.

- C pin

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

Figure 2.1-1 DBG/$\overline{RST}$/C Pin Connection
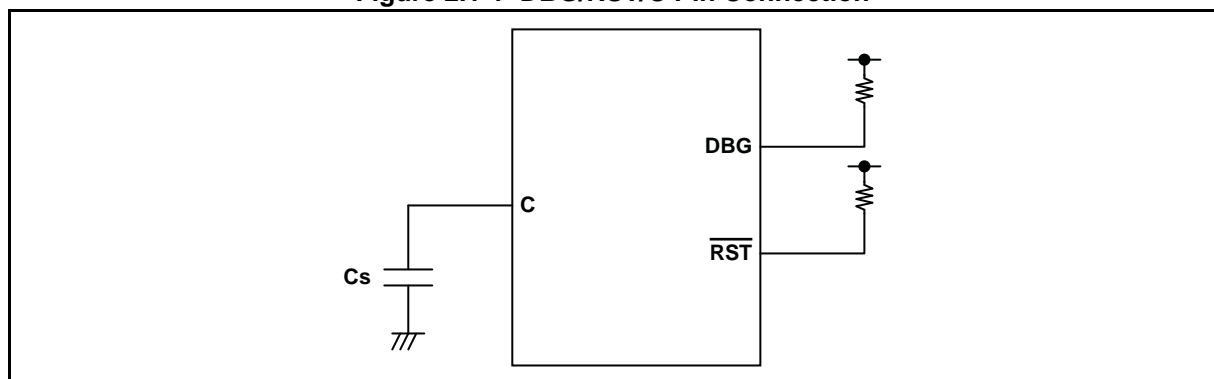
- Note on serial communication

  Since the device may receive wrong data generated by noise, minimize noise when designing the board layout for the sake of serial communication. In addition, consider adding a check bit (e.g. parity) to serial data to ensure the proper execution of serial communication.

FUJITSU SEMICONDUCTOR LIMITED

CHAPTER 3
MEMORY SPACE

This chapter describes the memory space.

3.1 Memory Space

The memory space of the MB95330H Series is 20 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 the product.
- The RAM area from 0090_H to $00FF_H$ can be accessed at high-speed by using the direct addressing instruction.
- In MB95F334H/F334K, the area from 0100_H to $047F_H$ is an extended direct addressing area. It can be accessed at high-speed by the direct addressing instruction with a direct bank pointer set.
- In MB95F333H/F333K, the area from 0100_H to $027F_H$ is an extended direct addressing area. It can be accessed at high-speed by the direct addressing instruction with a direct bank pointer set.
- In MB95F332H/F332K, the area from 0100_H to $017F_H$ is an extended direct addressing area. It can be accessed at high-speed by the direct addressing instruction with a direct bank pointer set.
- In MB95F333H/F333K/F334H/F334K, the area from 0100_H to $01FF_H$ can be used as a general-purpose register area.
- In MB95F332H/F332K, the area from 0100_H to $017F_H$ can be used as a general-purpose register area.

● Program area

- ROM is incorporated in the program area as the internal program area.
- The internal ROM size varies according to the 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.

MB95330H Series

■ Memory Map

Figure 3.1-1 Memory Map
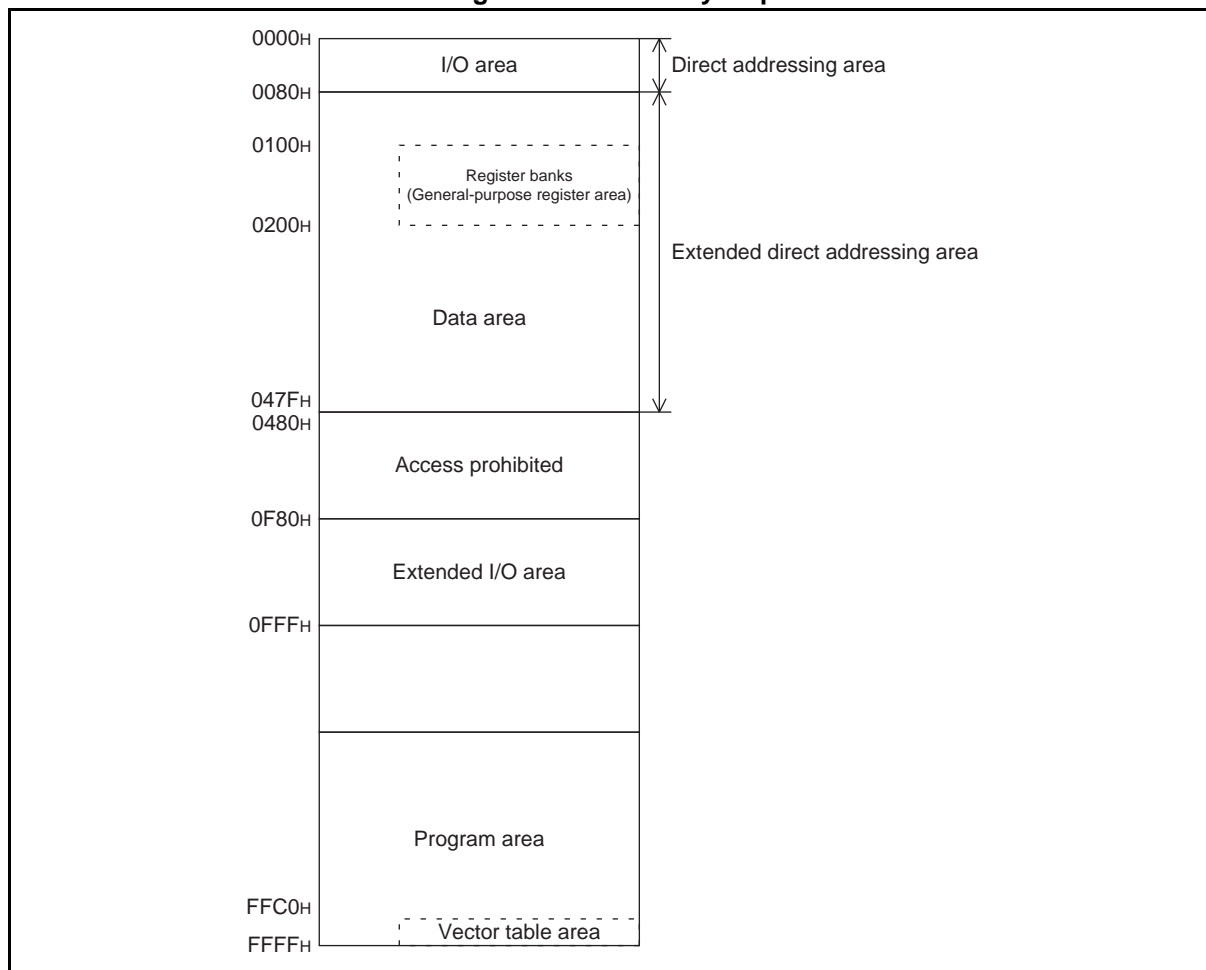
3.1.1 Areas for Specific Applications

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

■ General-purpose Register Area
(Addresses: 0100_H to $01FF_H$ in MB95F333H/F333K/F334H/F334K)
(Addresses: 0100_H to $017F_H$ in MB95F332H/F332K)

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

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

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

- The area from $FFBC_H$ to $FFBF_H$ is used to store data of the non-volatile register. For details, see CHAPTER 30 "NON-VOLATILE REGISTER (NVR) FUNCTION".

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

- This area is used as the vector table for vector call instructions (CALLV), interrupts, and resets.
- The top of the ROM area is allocated to the vector table area. The start address of a service routine is set to an address in the vector table in the form of data.

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

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

MB95330H Series

3.2 Memory Maps

This section shows the memory maps of the MB95330H Series.

■ Memory Maps

Figure 3.2-1 Memory Maps of Different Products
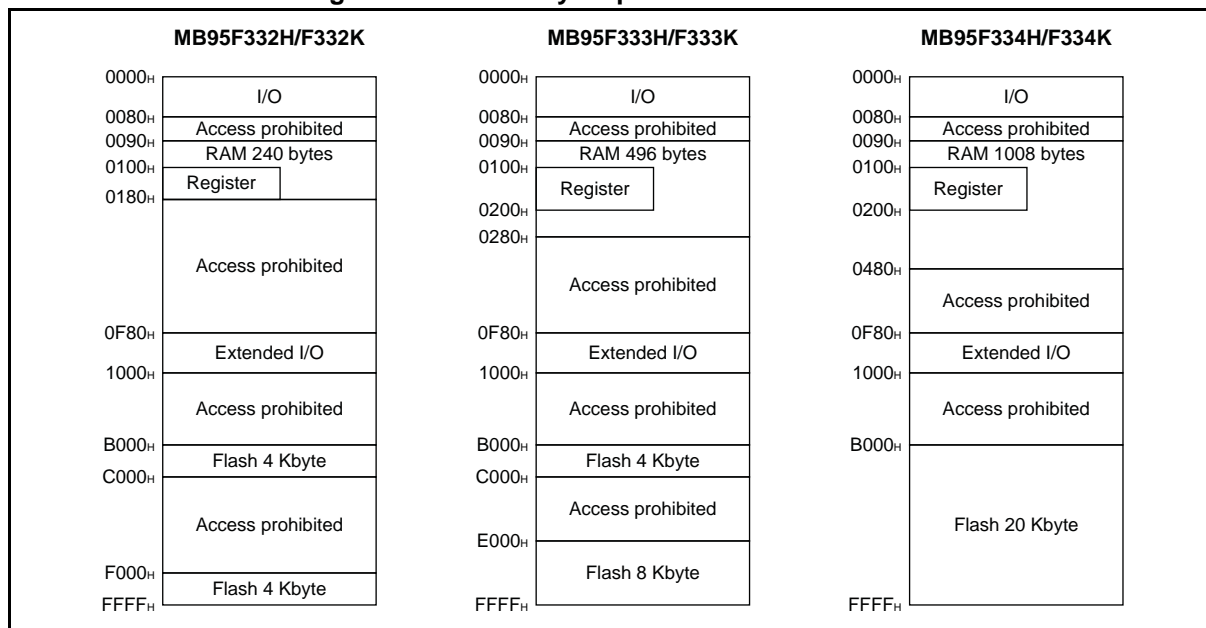
Parameter / Part number	Flash memory	RAM
MB95F332H/F332K	8 Kbyte	240 bytes
MB95F333H/F333K	12 Kbyte	496 bytes
MB95F334H/F334K	20 Kbyte	1008 bytes

CHAPTER 4
MEMORY ACCESS MODE

This chapter describes the memory access mode.

4.1   Memory Access Mode

4.1 Memory Access Mode

The MB95330H 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

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FFFD_H$								

Data	Operation
00_H	Selects single-chip mode.
Other than 00_H	Reserved. Do not set mode data to any value other than 00_H.
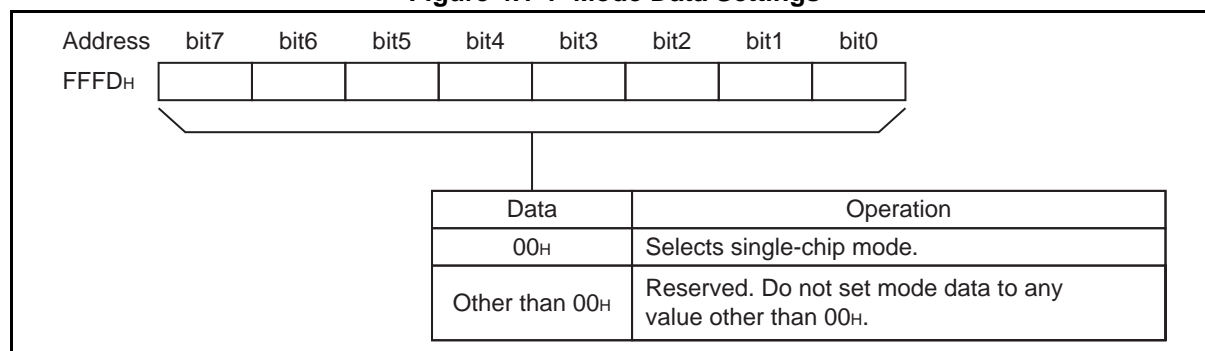
After a reset is released, the CPU fetches mode data first.

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

CHAPTER 5
CPU

This chapter describes the functions and operations of the CPU.

5.1 Dedicated Registers

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

■ Configuration of Dedicated Registers

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

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

Figure 5.1-1 Configuration of Dedicated Registers
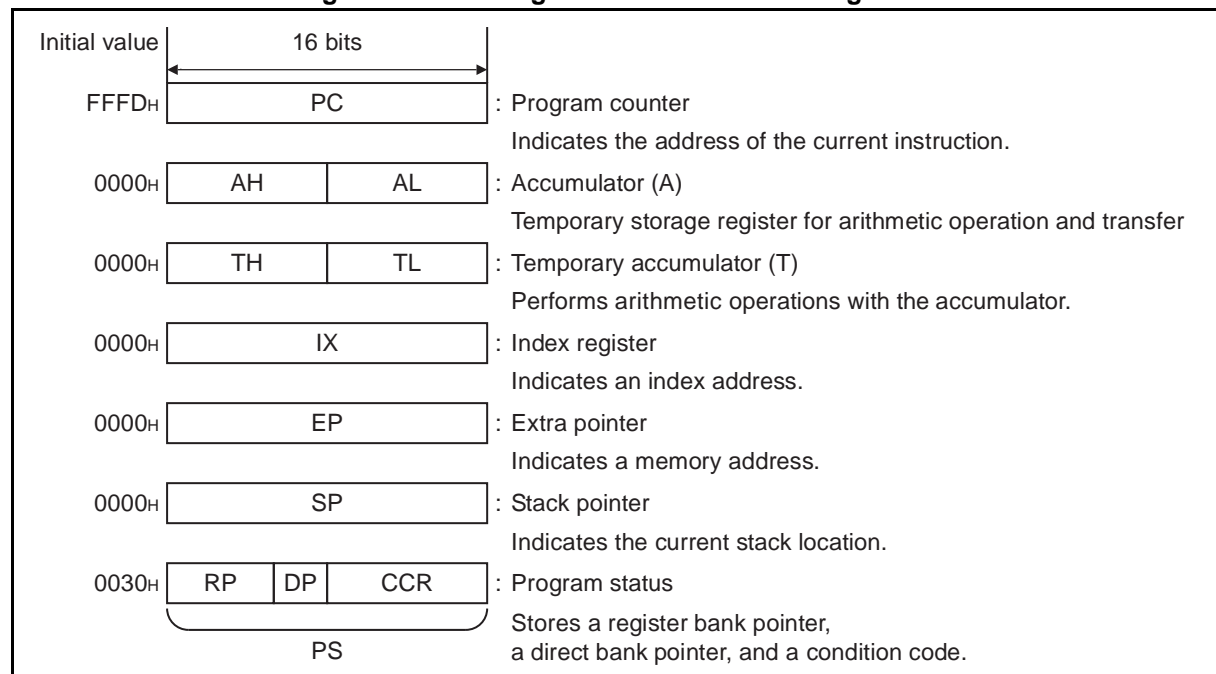
■ Functions of Dedicated Registers

● Program counter (PC)

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

● Accumulator (A)

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

MB95330H Series

● 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 "F^{2}MC-8FX Programming Manual" for details on using the dedicated registers.

5.1.1 Register Bank Pointer (RP)

The register bank pointer (RP) in bit15 to bit11 of the program status (PS) register contains the address of the general-purpose register bank that is currently in use and is translated into a real address when general-purpose register addressing is used.

■ Configuration of Register Bank Pointer (RP)

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

Figure 5.1-2 Configuration of Register Bank Pointer
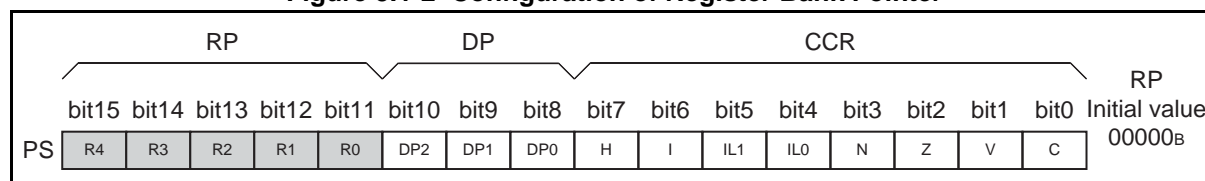
The register bank pointer contains the address of the register bank currently in use. The content of the register bank pointer is translated into a real address according to the rule shown in Figure 5.1-3.

Figure 5.1-3 Rule for Translation into Real Addresses in General-purpose Register Area
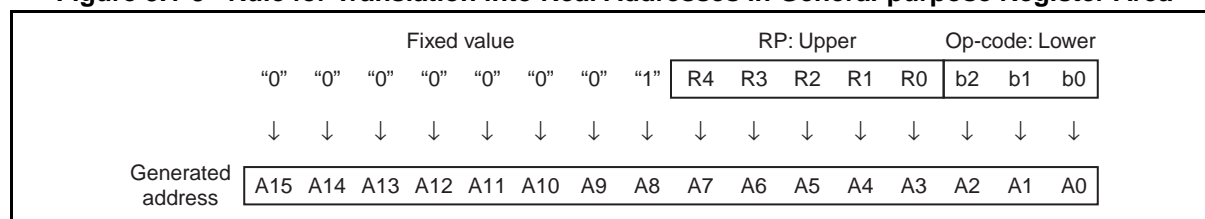
The register bank pointer specifies the register bank used as general-purpose registers in the RAM area. There are a total of 32 register banks. The current register bank is specified by setting a value between 0 and 31 in the upper five bits of the register bank pointer. Each register bank has eight 8-bit general-purpose registers which are selected by the lower three bits of the op-code.

The register bank pointer allows the space from "0100_H" to "$01FF_H$"(max) to be used as a general-purpose register area. However, certain products have restrictions on the size of the area available for the general-purpose register area. The initial value of the register bank pointer after a reset is "0000_H".

■ Mirror Address for Register Bank and Direct Bank Pointer

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

MB95330H Series

5.1.2   Direct Bank Pointer (DP)

The direct bank pointer (DP) in bit10 to bit8 of the program status (PS) register specifies the area to be accessed by direct addressing.

■ Configuration of Direct Bank Pointer (DP)

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

Figure 5.1-4 Configuration of Direct Bank Pointer
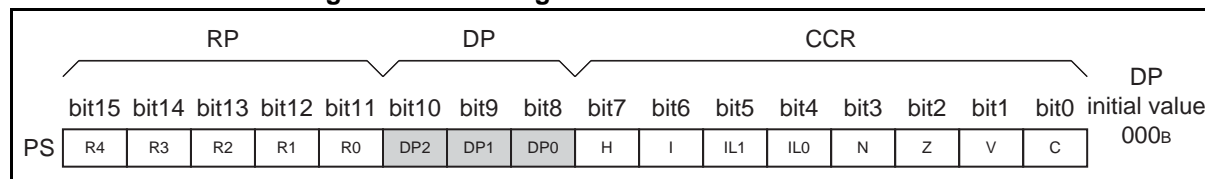
The area of "0000_H to $007F_H$" and that of "0080_H to $047F_H$" can be accessed by direct addressing. Access to 0000_H to $007F_H$ is specified by an operand regardless of the value in the direct bank pointer. Access to 0080_H to $047F_H$ is specified by the value of the direct bank pointer and the 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 $00FF_H$
001_B		0100_H to $017F_H$[*1]
010_B		0180_H to $01FF_H$
011_B		0200_H to $027F_H$[*2]
100_B		0280_H to $02FF_H$
101_B		0300_H to $037F_H$
110_B		0380_H to $03FF_H$
111_B		0400_H to $047F_H$[*3]

*1: The available access area is up to "$017F_H$" in MB95F332H/F332K.
*2: The available access area is up to "$027F_H$" in MB95F333H/F333K.
*3: The available access area is up to "0470_H" in MB95F334H/F334K.

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

MB95330H Series

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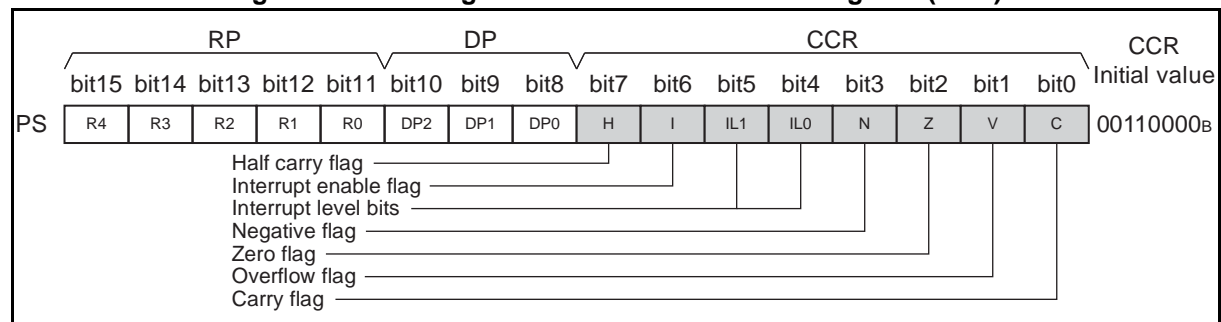
The condition code register is a part of the program status (PS) register and therefore cannot be accessed independently.

■ 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
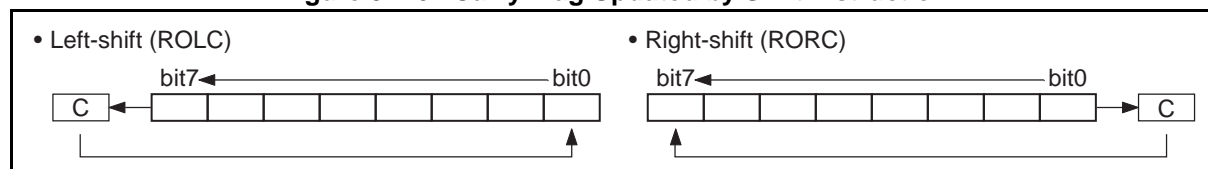
■ **Interrupt Acceptance Control Bits**

● 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 (IRQ00 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 Section 8.1 "Interrupts".

MB95330H Series

5.2 General-purpose Register

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

■ Configuration of General-purpose Register

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

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

Figure 5.2-1 Configuration of Register Banks
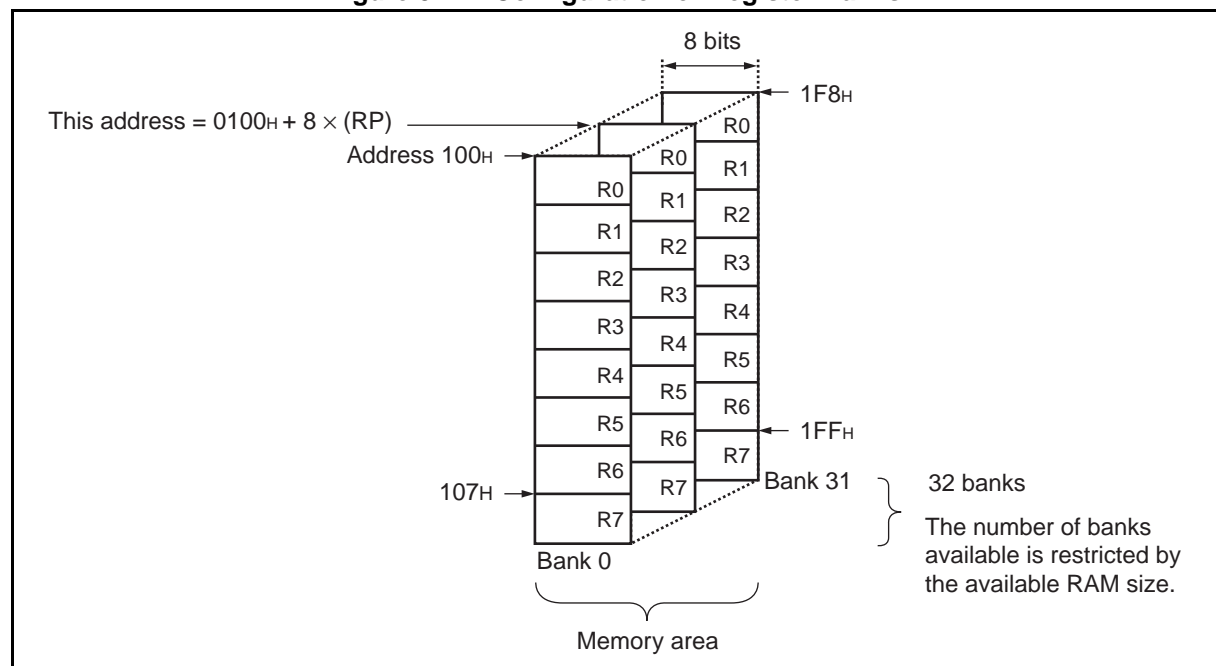
For information on the general-purpose register area available in each model, see Section 3.1.1 "Areas for Specific Applications".

■ **Features of General-purpose Registers**

The general-purpose register has the following features.

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

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

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

Notes:

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

- Read the interrupt level bits and save their values before writing a value to the RP.
- Directly write a new value to the RP mirror address "0078_H" to update the RP.
- As for the product whose RAM size is 240 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.

MB95330H Series

5.3 Placement of 16-bit Data in Memory

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

■ Placement of 16-bit Data in Memory

● State of 16-bit data stored in RAM

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

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

Figure 5.3-1 Placement of 16-bit Data in Memory
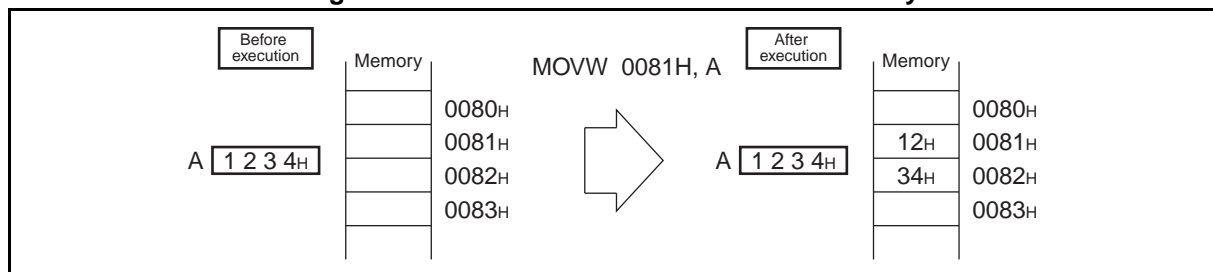
● Storage state of 16-bit data specified by an operand

Even when the operand in an instruction specifies 16-bit data, the upper byte is stored at the address closer to the op-code (instruction) and the lower byte is stored at the address next to the one at which the upper byte is stored.

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

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

Figure 5.3-2 Placement of 16-bit Data in Instruction
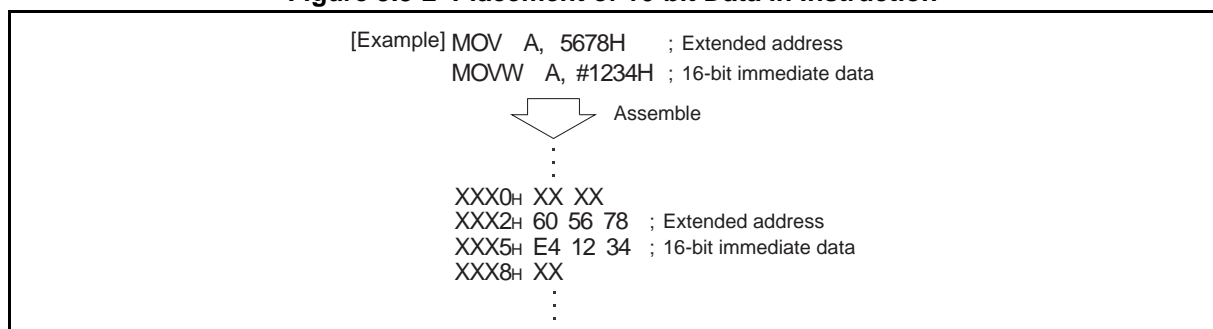
● Storage state of 16-bit data in the stack

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

CHAPTER 6
CLOCK CONTROLLER

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

6.1 Overview of Clock Controller

The F^{2}MC-8FX family has a built-in clock controller that optimizes its power consumption. It 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 internal CR oscillator and frequency divider circuits.

■ Overview of Clock Controller

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

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

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

This device has four source clocks: a main clock that is the main oscillation clock divided by two, a subclock that is the sub-oscillation clock divided by two, a main CR clock that is the trimmed accurate clock, and a sub-CR clock that is not trimmed by the CR clock divided by two.

MB95330H Series

■ Block Diagram of Clock Controller

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

Figure 6.1-1 Block Diagram of Clock Controller

(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)
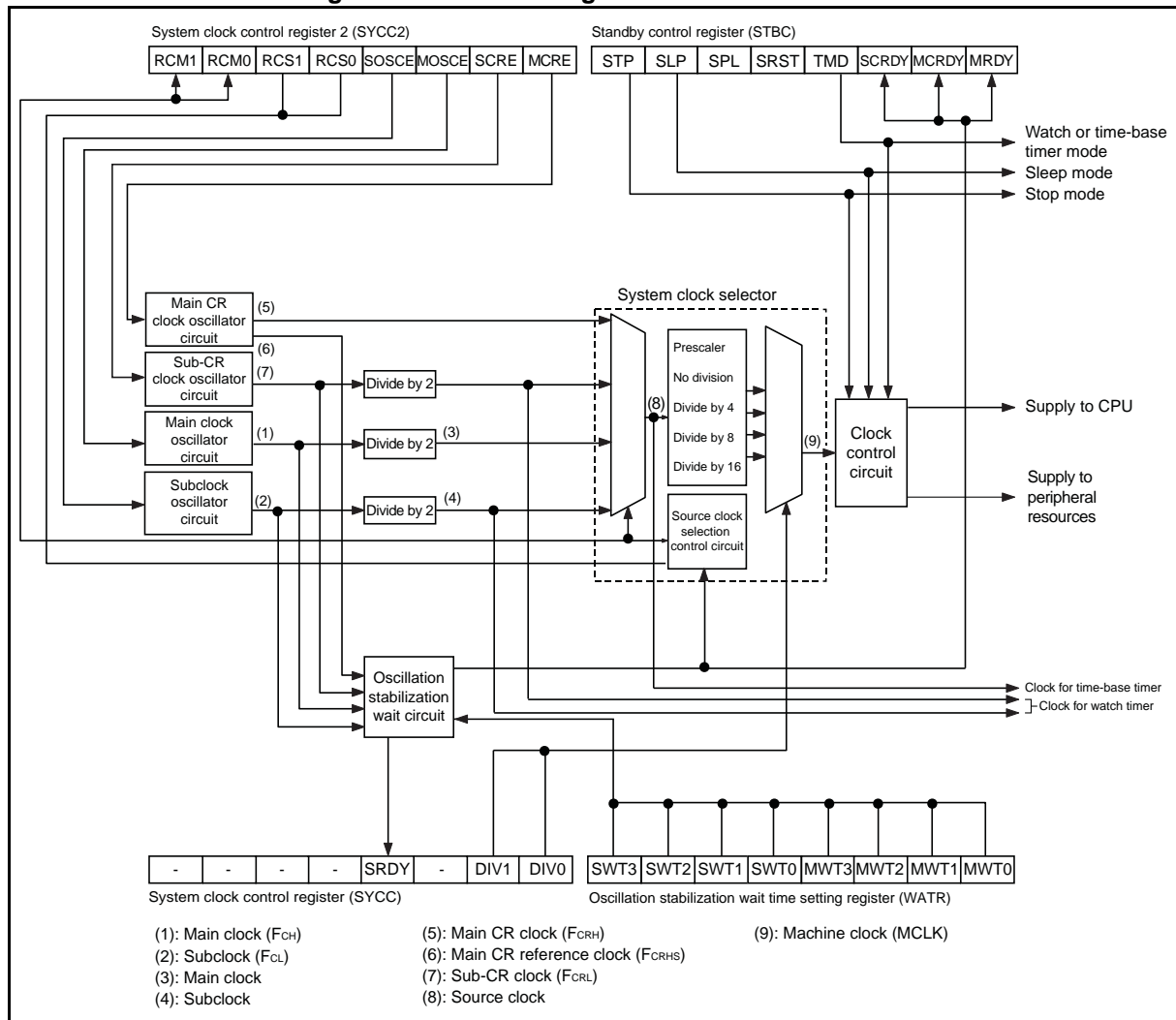
The clock controller consists of the following blocks:

● 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.

MB95330H Series

■ Clock Modes

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

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

Table 6.1-1   Clock Modes and Machine Clock Selection

Clock mode	Machine clock
Main clock mode	The machine clock is generated from the main clock (main clock divided by 2).
Main CR clock mode	The machine clock is generated from the main CR clock.
Subclock mode	The machine clock is generated from the subclock (subclock divided by 2).
Sub-CR clock mode	The machine clock is generated from the sub-CR clock.

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

■ Peripheral Function not Affected by Clock Mode

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

Table 6.1-2   Peripheral Function Not Affected by Clock Mode

Peripheral function	Operating clock
Watchdog timer	Main clock (with 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.

MB95330H Series

■ 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.

MB95330H Series

6.2 Oscillation Stabilization Wait Time

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

■ Oscillation Stabilization Wait Time

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

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

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

Figure 6.2-1 Behavior of Oscillator Immediately after Starting Oscillation
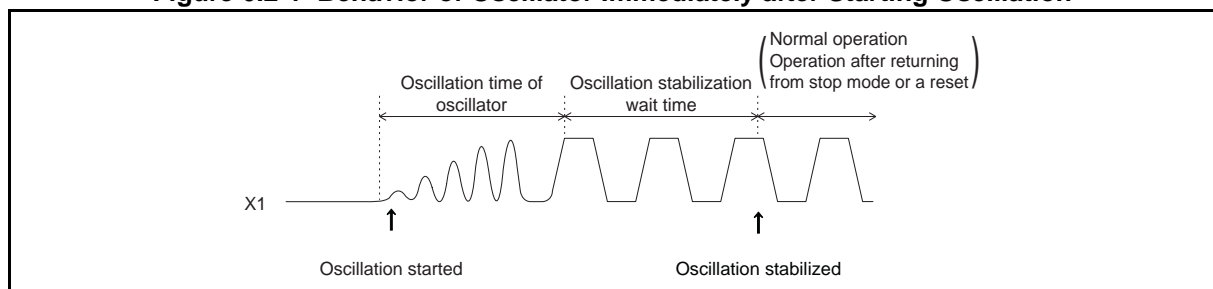
Oscillation stabilization wait time of main clock, subclock, main CR clock, sub-CR clock is counted by using a dedicated counter. The count value can be set in the oscillation stabilization wait time setting register (WATR). Set it in keeping with the oscillator characteristics.

When a power-on reset occurs, the oscillation stabilization wait time is fixed at the initial value.

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

Table 6.2-1 Oscillation Stabilization Wait Time

Clock	Reset source	Oscillation stabilization wait time
Main clock	Power-on reset	Initial value: $(2^{14}-2)/F_{CH}$. F_{CH} is the main clock frequency
	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	$2^8/F_{CRH}$*
Sub-CR clock	$2^5/F_{CRL}$

*: $F_{CRH} = 1$ 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 Section 6.7 "Clock Modes" and Section 6.8 "Operations in Low-power Consumption Mode (Standby Mode)".

MB95330H Series

6.3 System Clock Control Register (SYCC)

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

■ Configuration of System Clock Control Register (SYCC)

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

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0007H	-	-	-	-	SRDY	-	DIV1	DIV0	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/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
░ : Initial value

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

Bit name		Function
bit7 to bit4, bit2	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit3	SRDY: Subclock oscillation stabilization bit	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.
bit1, bit0	DIV1, DIV0: Machine clock divide ratio select bits	• 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 below

DIV1	DIV0	Machine clock divide ratio
0	0	Source clock (No division)
0	1	Source clock/4
1	0	Source clock/8
1	1	Source clock/16

MB95330H Series

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
0005_H	SWT3	SWT2	SWT1	SWT0	MWT3	MWT2	MWT1	MWT0	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

MWT3	MWT2	MWT1	MWT0	Number of cycles	Main Oscillation Clock F_{CH} = 4 MHz	
1	1	1	1	$2^{14} - 2$	$(2^{14} - 2)/F_{CH}$	About 4.10 ms
1	1	1	0	$2^{13} - 2$	$(2^{13} - 2)/F_{CH}$	About 2.05 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)

Bit name		Function
bit7 to bit4	SWT3, SWT2, SWT1, SWT0: Subclock oscillation stabilization wait time select bits	These bits set the subclock oscillation stabilization wait time.

SWT3, SWT2, SWT1, SWT0	Number of cycles	Subclock F_{CL} = 32.768 kHz	
1111_B	$2^{15}-2$	$(2^{15}-2)/F_{CL}$	About 1.0 s
1110_B	$2^{14}-2$	$(2^{14}-2)/F_{CL}$	About 0.5 s
1101_B	$2^{13}-2$	$(2^{13}-2)/F_{CL}$	About 0.25 s
1100_B	$2^{12}-2$	$(2^{12}-2)/F_{CL}$	About 0.125 s
1011_B	$2^{11}-2$	$(2^{11}-2)/F_{CL}$	About 62.44 ms
1010_B	$2^{10}-2$	$(2^{10}-2)/F_{CL}$	About 31.19 ms
1001_B	$2^{9}-2$	$(2^{9}-2)/F_{CL}$	About 15.56 ms
1000_B	$2^{8}-2$	$(2^{8}-2)/F_{CL}$	About 7.75 ms
0111_B	$2^{7}-2$	$(2^{7}-2)/F_{CL}$	About 3.85 ms
0110_B	$2^{6}-2$	$(2^{6}-2)/F_{CL}$	About 1.89 ms
0101_B	$2^{5}-2$	$(2^{5}-2)/F_{CL}$	About 915.5 μs
0100_B	$2^{4}-2$	$(2^{4}-2)/F_{CL}$	About 427.2 μs
0011_B	$2^{3}-2$	$(2^{3}-2)/F_{CL}$	About 183.1 μs
0010_B	$2^{2}-2$	$(2^{2}-2)/F_{CL}$	About 61.0 μs
0001_B	$2^{1}-2$	$(2^{1}-2)/F_{CL}$	0.0 μs
0000_B	$2^{1}-2$	$(2^{1}-2)/F_{CL}$	0.0 μs

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.

MB95330H Series

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

Bit name		Function
bit3 to bit0	MWT3, MWT2, MWT1, MWT0: Main clock oscillation stabilization wait time select bits	These bits set the main clock oscillation stabilization wait time.

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

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.

■ 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:

MB95F332H/F333H/F334H

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
0008H	STP	SLP	SPL	SRST	TMD	SCRDY	MCRDY	MRDY	00000XXXB
	R0,W	R0,W	R/W	R0,W	R0,W	R/WX	R/WX	R/WX	

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

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

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

TMD	Watch bit		
	Read	Write	
0	"0" is always read.	Has no effect on operation.	
1	-	Main clock mode/ Main CR clock mode — Causes transition to time-base timer mode	Subclock mode/Sub-CR clock 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".)
-	: Undefined bit
X	: Indeterminate
	: Initial value

MB95330H Series

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

Bit name		Function
bit7	STP: Stop bit	This bit sets the transition to stop mode. **Writing "0":** this bit is meaningless. **Writing "1":** causes the device to transit to stop mode. When this bit is read, it always returns "0". Note: After an interrupt request is issued, writing "1" to this bit is ignored. For details, see Section 6.8.1 "Notes on Using Standby Mode".
bit6	SLP: Sleep bit	This bit sets the transition to sleep mode. **Writing "0":** this bit is meaningless. **Writing "1":** causes the device to transit to sleep mode. When this bit is read, it always returns "0". Note: After an interrupt request is issued, writing "1" to this bit is ignored. For details, see Section 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 Section 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.

Notes:

- Set the standby mode after making sure that the transition to clock mode has been completed by comparing the values of the clock mode monitor bits (SYCC2:RCM1,RCM0) and clock mode setting bits (SYCC2:RCS1,RCS0) in the system clock control register 2.

- If two or more of the following bits, stop bit (STP), sleep bit (SLP), software reset bit (SRST) and watch bit (TMD), are set to "1" together, the order of priority for such bits is as follows:

  (1) Software reset bit (SRST)

  (2) Stop bit (STP)

  (3) Watch bit (TMD)

  (4) Sleep bit (SLP)

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

MB95330H Series

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.

MB95330H Series

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 operates with the subclock.

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 operates with the subclock.

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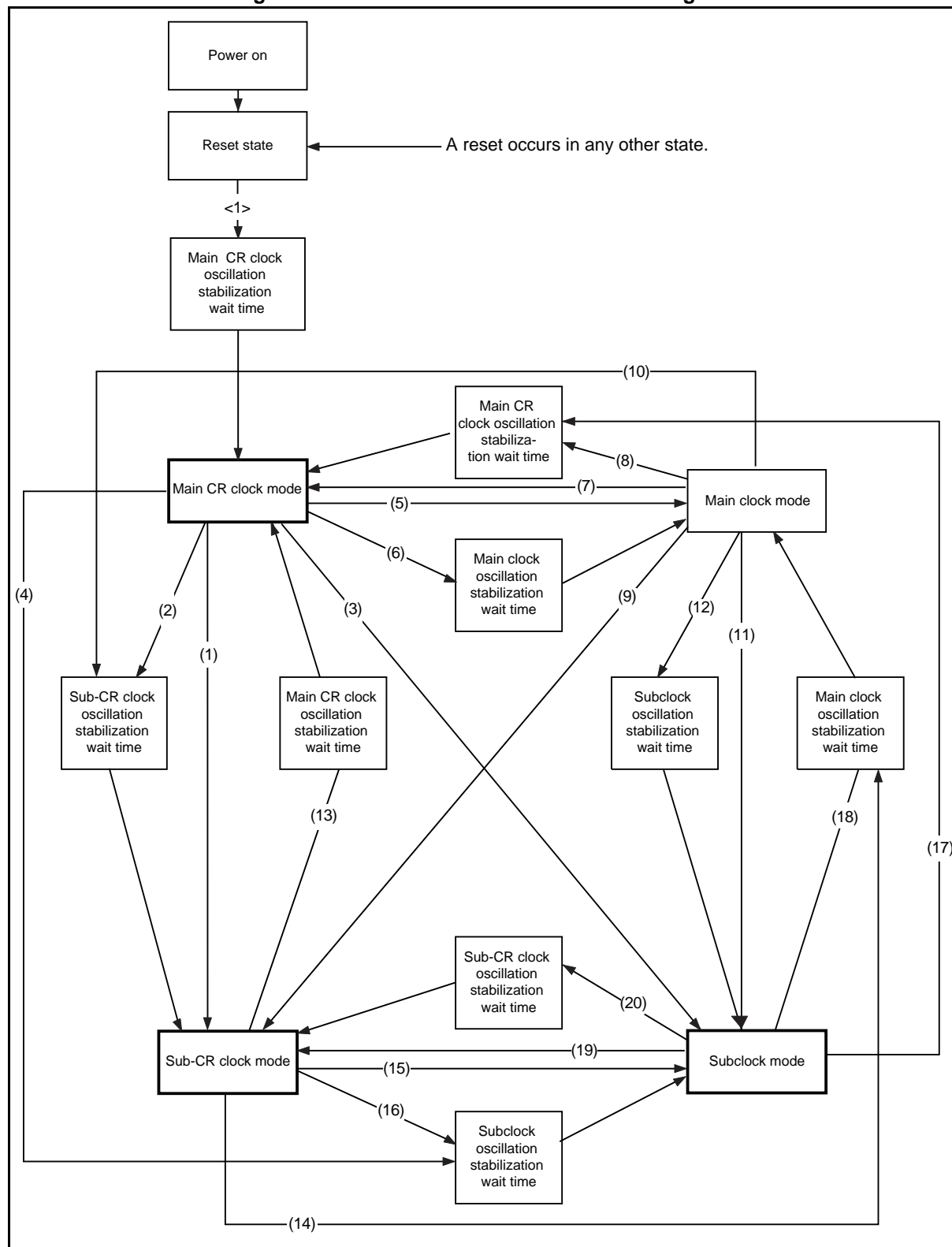
Table 6.7-1 Clock Mode State Transition Table (1 / 2)

	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) (2)	Main CR clock	Sub-CR clock	The device transits to sub-CR clock mode when the system clock 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", the device transits to sub-CR clock mode immediately after the system clock select bits (SYCC2:RCS1, RCS0) are set to "00_B".
(3) (4)		Subclock	When the system clock 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", the device transits to subclock mode immediately after the system clock select bits (SYCC2:RCS1, RCS0) are set to "01_B".
(5) (6)		Main clock	When the system clock 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", the device transits to main clock mode immediately after the system clock select bits (SYCC2:RCS1, RCS0) are set to "11_B".
(7) (8)	Main clock	Main CR clock	When the system clock 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", the device transits to main CR clock mode immediately after the system clock select bits (SYCC2:RCS1, RCS0) are set to "10_B".
(9) (10)		Sub-CR clock	Same as (1) and (2)
(11) (12)		Subclock	Same as (3) and (4)

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

	Current State	Next State	Description
(13)	Sub-CR clock	Main CR clock	When the system clock 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 system clock 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 MB95330H 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 setting bits (SYCC2:RCS1, RCS0) in the system clock control register.

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

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

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

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

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

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.

MB95330H Series

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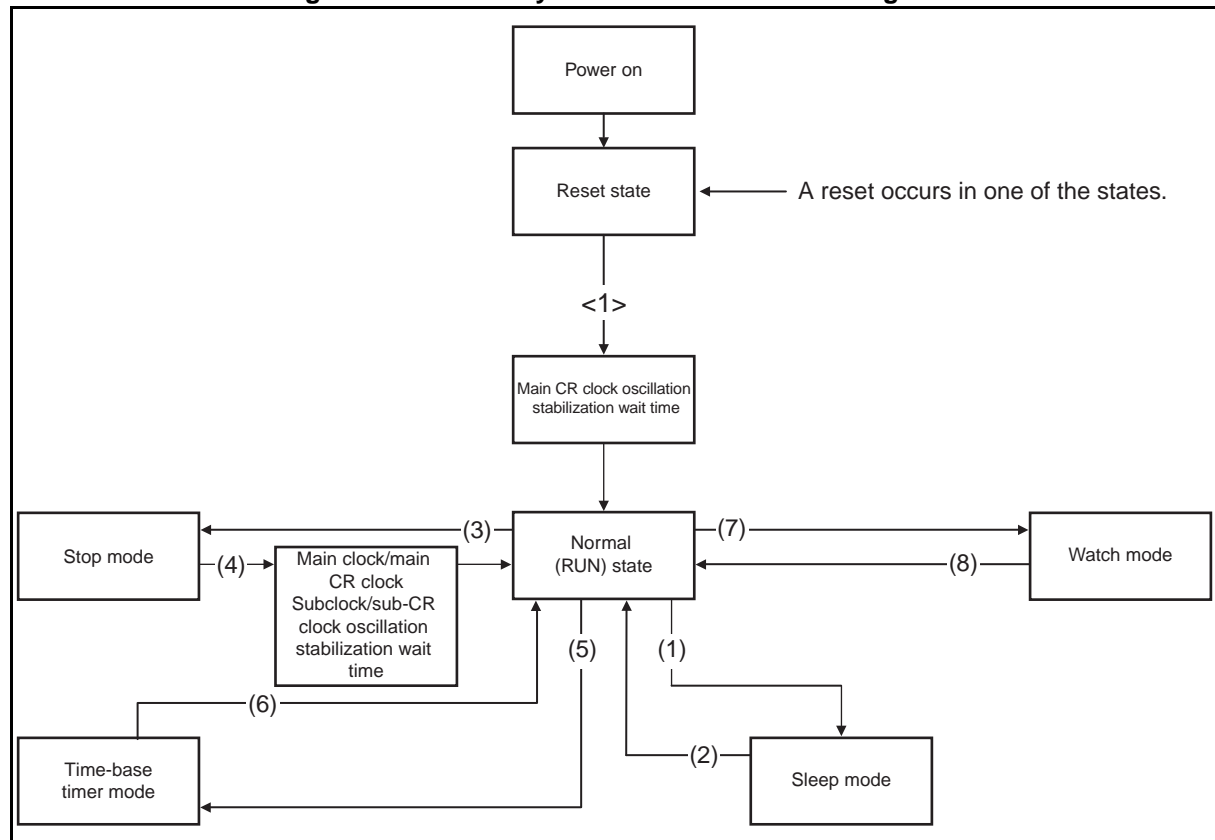
Table 6.8-1 State Transition Table (Transitions to and from Standby Modes)

	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 waits 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 (NVR) FUNCTION".

● 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.

MB95330H Series

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 (NVR) FUNCTION".

● 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 (NVR) FUNCTION".

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 (NVR) FUNCTION".

● Transition to time-base timer mode

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

● Transition to watch mode

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

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

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

● Release from watch mode

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

Note:

If the device is released from watch mode by an interrupt, a peripheral function having transited to 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.

MB95330H 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
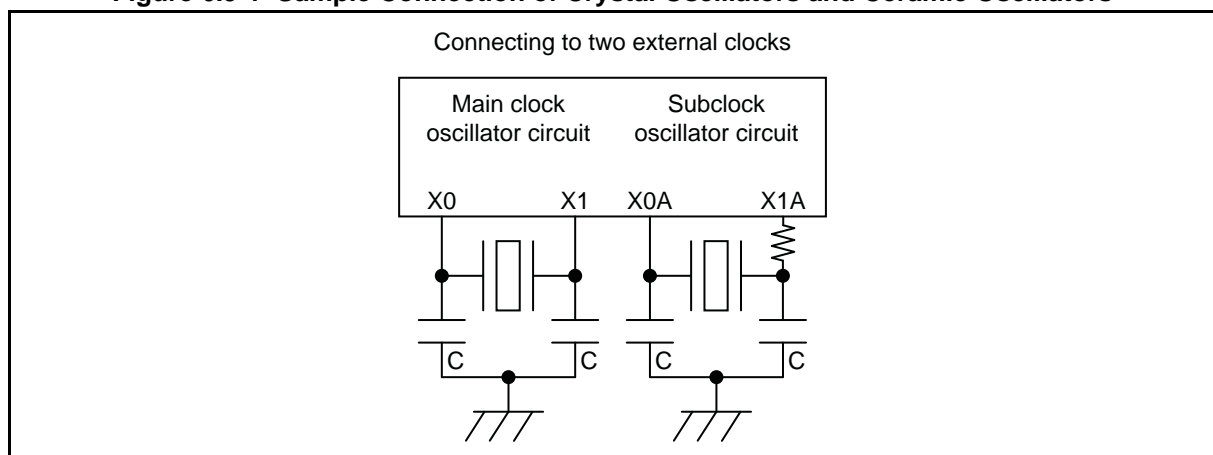
● Using external clock

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 MB95330H 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. In addition, clock signals can be supplied to the external clock input pins HCLK1/HCLK2.

Figure 6.9-2 Sample Connection of External Clocks
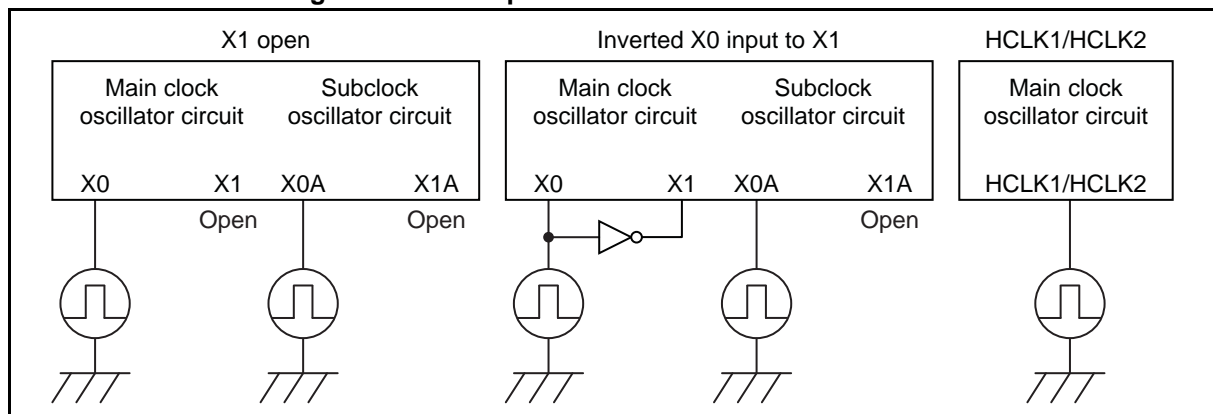
6.10   Overview of Prescaler

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

MB95330H 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
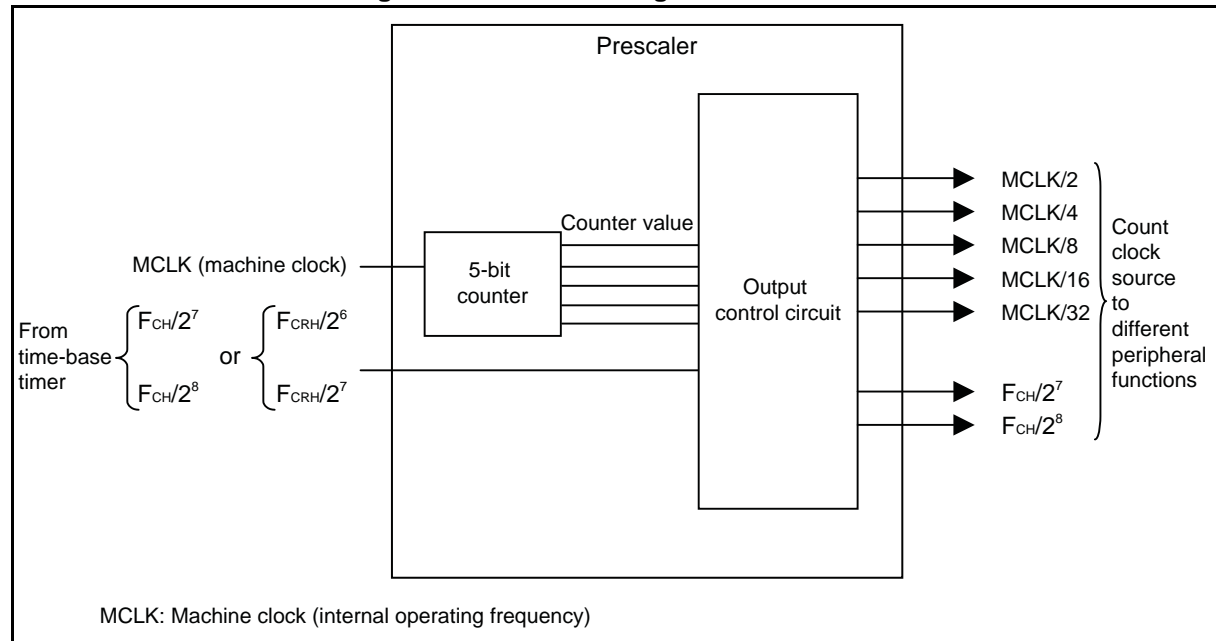
• 5-bit counter

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 lists the count clock sources generated by the prescaler.

Table 6.12-1 Count Clock Sources Generated by Prescaler

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

MB95330H Series

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
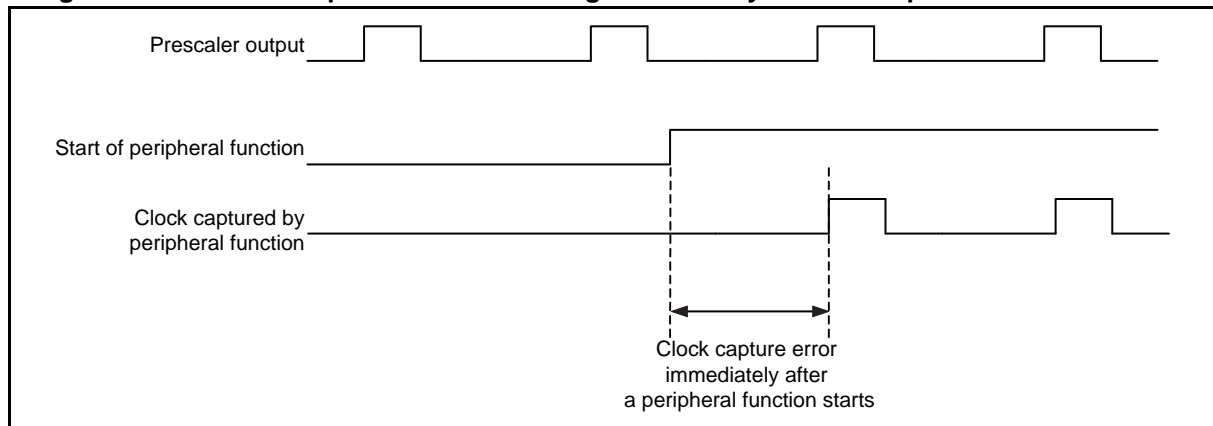
The prescaler count value affects the following peripheral functions:

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

FUJITSU SEMICONDUCTOR LIMITED

CHAPTER 7
RESET

This chapter 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.

MB95330H Series

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

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 Section 1.2 "Product Line-up of MB95330H 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 19 "LOW-VOLTAGE DETECTION RESET CIRCUIT".

■ Reset Time

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

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

■ Reset Output

The reset pin outputs "L" level during a reset provided that the reset input function is enabled. However, during an external reset, the reset pin cannot output "L" level. For details of the settings of the reset input function and reset output function, see CHAPTER 31 "SYSTEM CONFIGURATION CONTROLLER".

■ Overview of Reset Operation

Figure 7.1-1 Reset Operation Flow
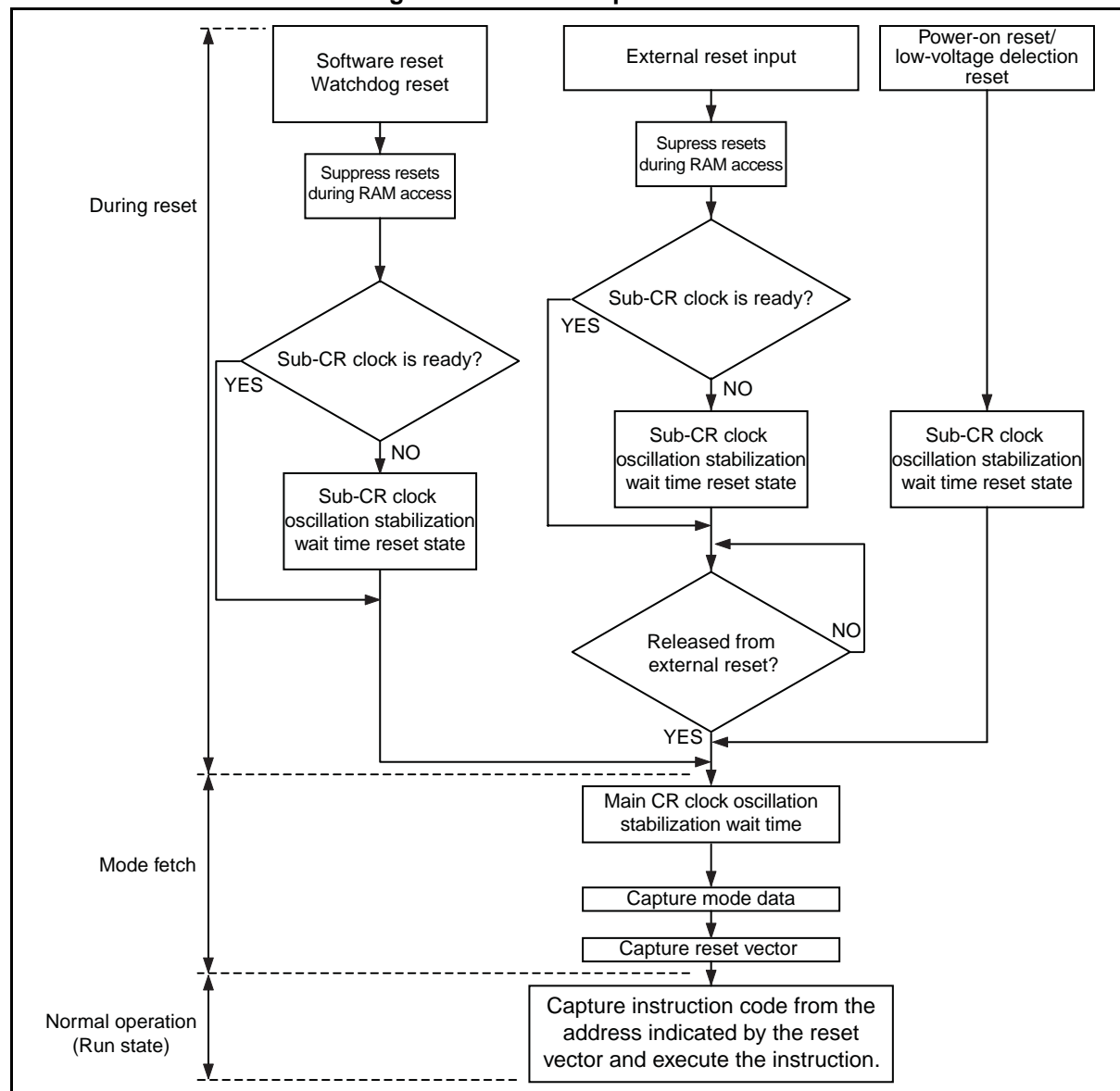
In any reset, the CPU performs mode fetch after the main CR clock oscillation stabilization wait time elapses.

■ 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.

MB95330H Series

■ 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 MB95330H Series".

7.2 Reset Source Register (RSRR)

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

■ Configuration of Reset Source Register (RSRR)

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

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0009H	-	-	-	EXTS	WDTR	PONR	HWR	SWR	XXXXXXXXB
	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	The read value is always "0". Writing a value to it 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.

MB95330H Series

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.

FUJITSU SEMICONDUCTOR LIMITED

CHAPTER 8

INTERRUPTS

This chapter describes the interrupts.

8.1   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, IRQ00 is at the top of the priority order.

MB95330H Series

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

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

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

If the interrupt level of an interrupt request is 3, the CPU ignores that interrupt request.

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

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

LXX[1:0]	Interrupt level	Priority
00	0	Highest
01	1	
10	2	
11	3	Lowest (No interrupt)

XX:00 to 23   Number of an interrupt request

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

MB95330H Series

8.1.2   Interrupt Processing

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

■ Interrupt Processing

The procedure for processing an interrupt is as follows: the generation of an interrupt source in a peripheral resource, the execution of the main program, the setting of the interrupt request flag bit, the evaluation of the interrupt request enable bit, the evaluation of the interrupt level (ILR0 to ILR5 and CCR:IL1, IL0), the checking for interrupt requests of the same interrupt level made simultaneously, and the evaluation of the interrupt enable flag (CCR:I).

Figure 8.1-2 shows the interrupt processing.

Figure 8.1-2 Interrupt Processing
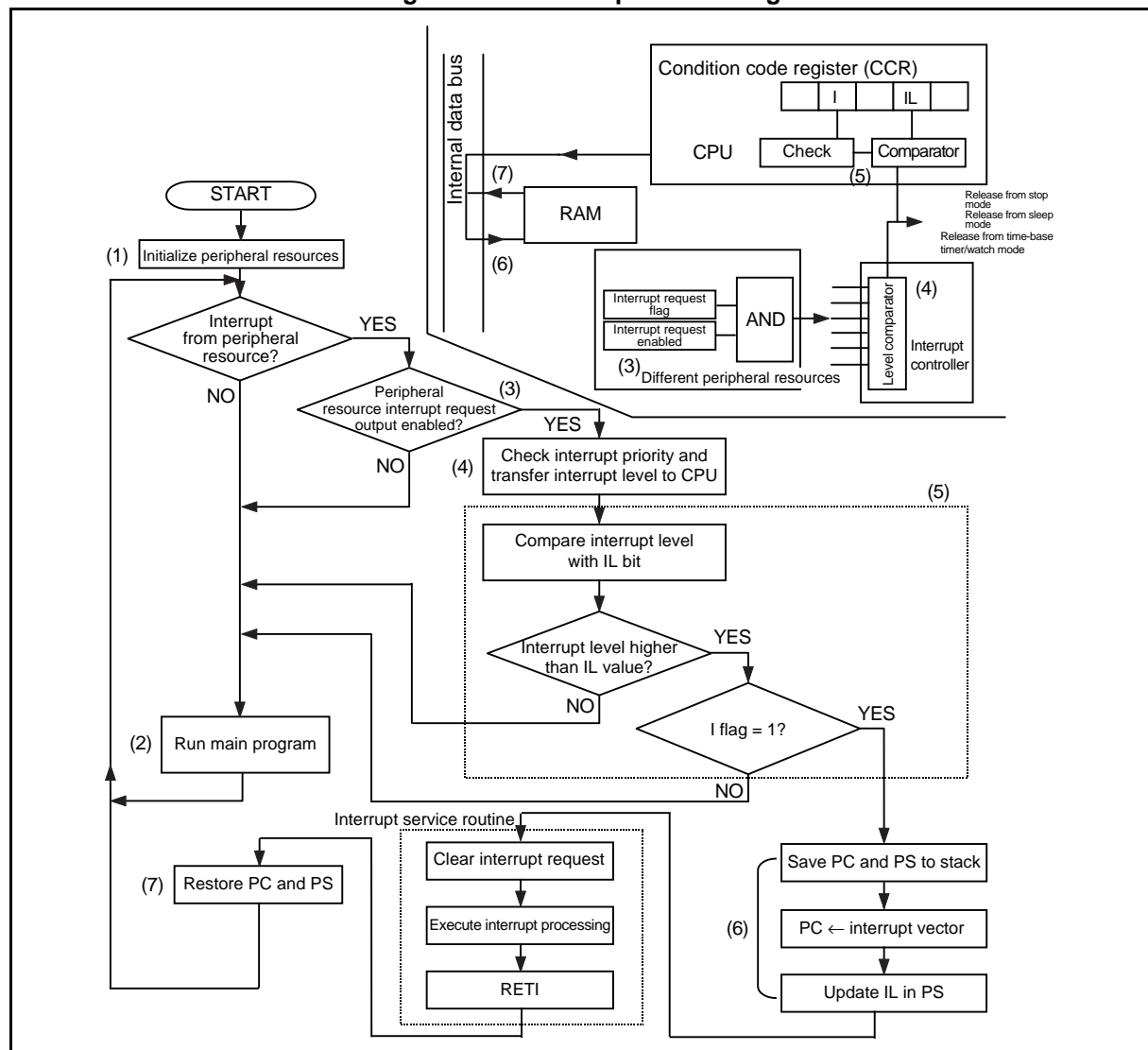
(1) All interrupt requests are disabled immediately after a reset. In the peripheral resource initialization program, initialize those peripheral functions that generate interrupts and set their interrupt levels in their respective interrupt level setting registers (ILR0 to ILR5) before starting operating such peripheral functions. The interrupt level can be set to 0, 1, 2, or 3. Level 0 is given the highest priority, and level 1 the second highest. Assigning level 3 to a peripheral resource disables interrupts from that peripheral resource.

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

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

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

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

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

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

Note:

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

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

MB95330H Series

8.1.3   Nested Interrupts

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

■ Nested Interrupts

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

[Example: Nested interrupts]

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

Figure 8.1-3 Example of Nested Interrupts
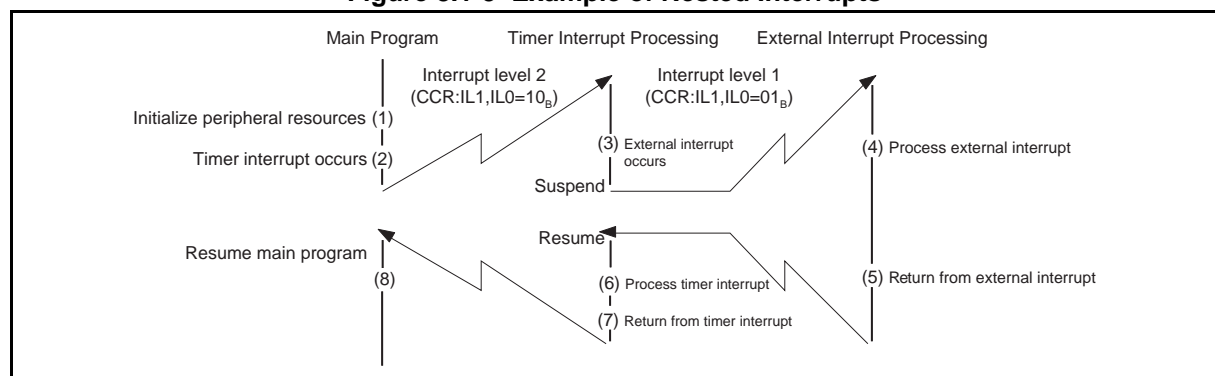
- While the timer interrupt is being processed, the interrupt level bits in the condition code register (CCR: IL1, IL0) hold the same value as that of the interrupt level setting registers (ILR0 to ILR5) corresponding to the timer interrupt (level 2 in this example). If an interrupt request whose interrupt level has priority over the interrupt level of the timer interrupt (level 1 in the example) is made, that interrupt is processed first.

- To temporarily disable nested interrupts processing while the timer interrupt is being processed, disable interrupts by setting the interrupt enable flag in the condition code register (CCR:I) to "0", or set the interrupt level bits (CCR:IL1, IL0) to "00_B".

- After the interrupt processing is completed, if the interrupt return instruction (RETI) is executed, the value of the program counter (PC) and that of the program status (PS) are restored, and the CPU resumes executing the program interrupted. In addition, the values of the condition code register (CCR) return to the ones existing before the interrupt due to the restoration of the value of the program status (PS).

8.1.4 Interrupt Processing Time

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

■ Interrupt Processing Time

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

● Interrupt request sampling wait time

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

● Interrupt handling time

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

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

Figure 8.1-4 Interrupt Processing Time
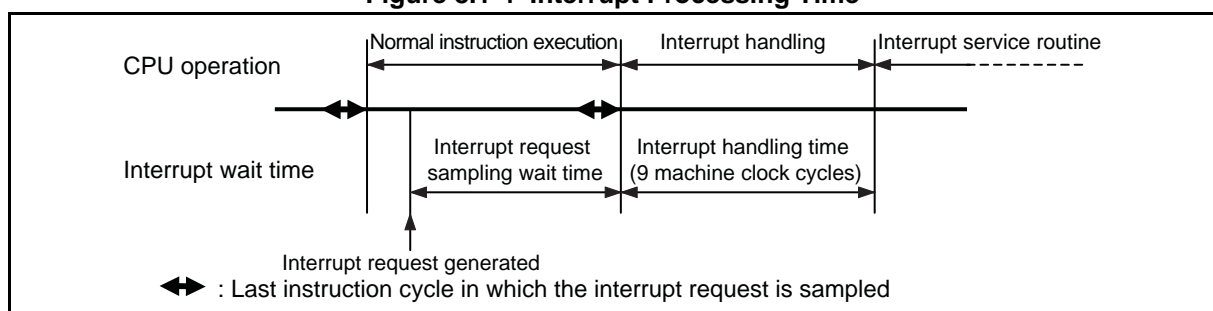
When an interrupt request occurs immediately after the CPU starts executing the DIVU instruction, whose execution cycle is the longest (17 machine clock cycles), the interrupt processing time spans 26 machine clock cycles.

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

MB95330H Series

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
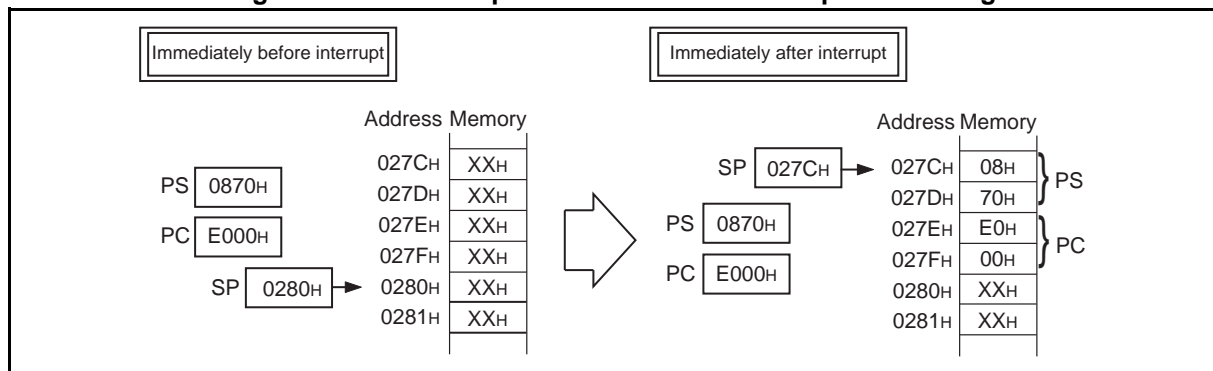
■ Stack Operation after Returning from an Interrupt

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 maximum RAM address and make the data area start from the lowest 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
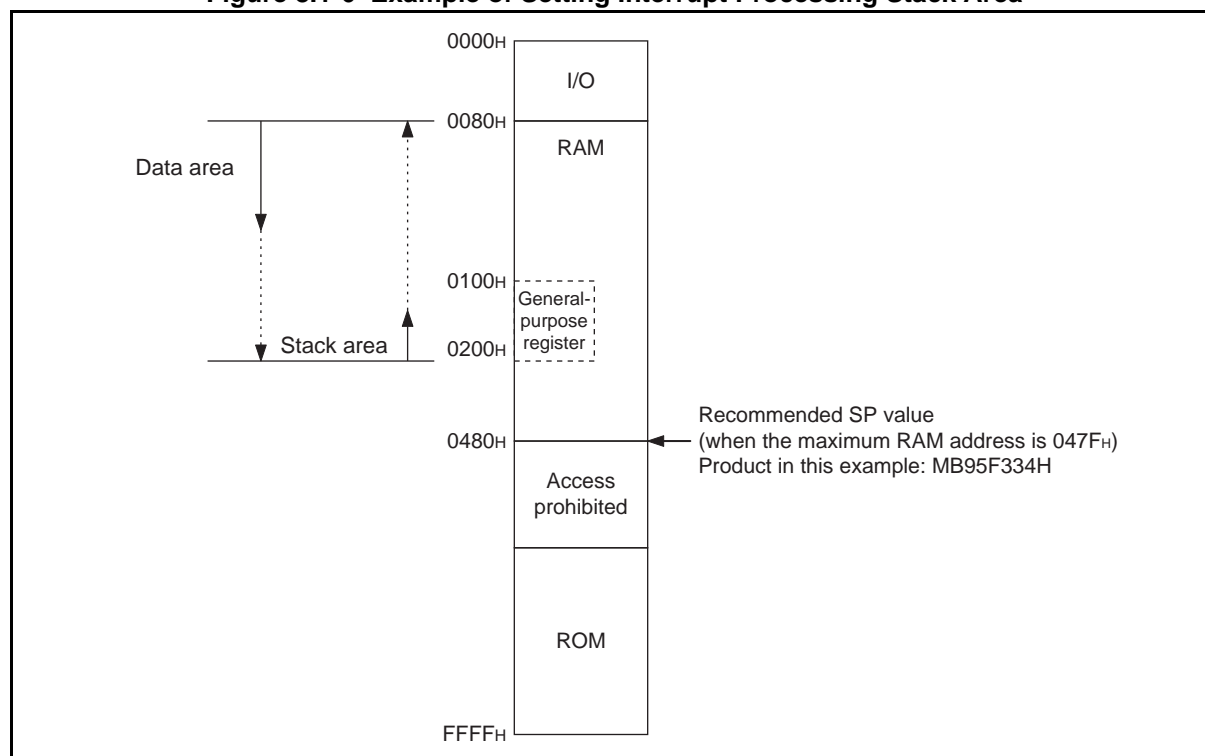
Note:

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 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 1 pull-up register	PUL1	R, RM/W	00000000_B
Port G pull-up register	PULG	R/W	00000000_B
A/D input disable register (Lower)	AIDRL	R/W	00000000_B
Input level select register	ILSR	R/W	00000000_B

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 (RMW) type of instruction.)

MB95330H Series

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	P01 general-purpose I/O	INT01 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN01 analog input				
P02/INT02/ AN02/SCK	P02 general-purpose I/O	INT02 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN02 analog input				
		LIN-UART clock I/O				
P03/INT03/ AN03/SOT	P03 general-purpose I/O	INT03 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN03 analog input				
		LIN-UART data output				
P04/INT04/ AN04/SIN/ HCLK1[*1]/ EC0	P04 general-purpose I/O	INT04 external interrupt input	Hysteresis/ CMOS/Analog	CMOS	-	❍
		AN04 analog input				
		LIN-UART data input				
		External clock input				
		8/16-bit composite timer ch. 0 clock input				
P05/INT05/ AN05/ HCLK2[*2]/ TO00	P05 general-purpose I/O	INT05 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN05 analog input				
		External clock input				
		8/16-bit composite timer ch. 0 output				
P06/INT06/ AN06/TO01	P06 general-purpose I/O	INT06 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN06 analog input				
		8/16-bit composite timer ch. 0 output				
P07/INT07/ AN07	P07 general-purpose I/O	INT07 external interrupt input	Hysteresis/ Analog	CMOS	-	❍
		AN07 analog input				

OD: Open drain, PU: Pull-up

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

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

MB95330H Series

■ Block Diagrams of Port 0

Figure 9.2-1 Block Diagram of P00 and P01
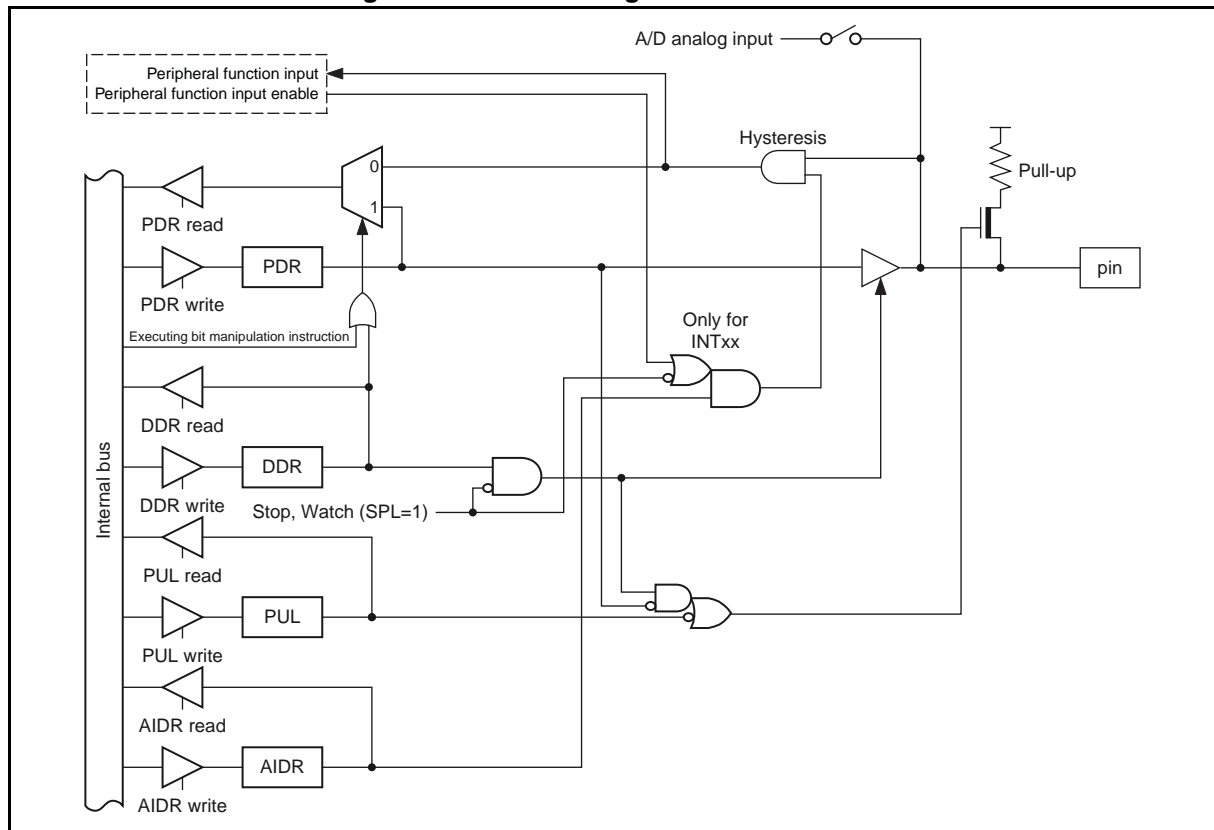
Figure 9.2-2 Block Diagram of P02, P03 and P05

Figure 9.2-3 Block Diagram of P04

Figure 9.2-4 Block Diagram of P06

Figure 9.2-5 Block Diagram of P07

9.2.1　Port 0 Registers

This section describes the registers of port 0.

■ Port 0 Register Functions

Table 9.2-2 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 lists the correspondence between port 0 pins and each register bit.

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

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

MB95330H Series

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 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 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 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 bit4 in ILSR to "1" changes only P04 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P04 should become the hysteresis input level.

- For pins other than P04, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.

- When changing the input level of P04, ensure that the peripheral function (LIN-UART/External interrupt/8/16-bit composite timer ch. 0 clock input/External clock input) has been stopped.

MB95330H Series

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.

9.3   Port 1

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

■ Port 1 Configuration

Port 1 is made up of the following elements.

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

■ Port 1 Pins

Port 1 has eight I/O pins.

Table 9.3-1 lists the port 1 pins.

Table 9.3-1 Port 1 Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P10/PPG10	P10 general-purpose I/O	8/16-bit PPG ch. 1 output	Hysteresis	CMOS	-	❍
P11/PPG11	P11 general-purpose I/O	8/16-bit PPG ch. 1 output	Hysteresis	CMOS	-	❍
P12/ DBG/EC0	P12 general-purpose I/O	DBG input pin	Hysteresis	CMOS	❍	-
		8/16-bit composite timer ch. 0 clock input				
P13/PPG00	P13 general-purpose I/O	8/16-bit PPG ch. 0 output	Hysteresis	CMOS	-	❍
P14/UCK0/ PPG01	P14 general-purpose I/O	UART/SIO ch. 0 clock I/O	Hysteresis	CMOS	-	❍
		8/16-bit PPG ch. 0 output				
P15/UO0/ PPG20	P15 general-purpose I/O	UART/SIO ch. 0 data output	Hysteresis	CMOS	-	❍
		8/16-bit PPG ch. 2 output				
P16/UI0/ PPG21	P16 general-purpose I/O	UART/SIO ch. 0 data input	Hysteresis/ CMOS	CMOS	-	❍
		8/16-bit PPG ch. 2 output				
P17/TO1/ SNI0	P17 general-purpose I/O	16-bit reload timer ch. 1 output	Hysteresis	CMOS	-	❍
		Trigger input for the position detection function of the MPG waveform sequencer				

OD: Open drain, PU: Pull-up

MB95330H Series

■ Block Diagrams of Port 1

Figure 9.3-1 Block Diagram of P10, P11, P13 and P15
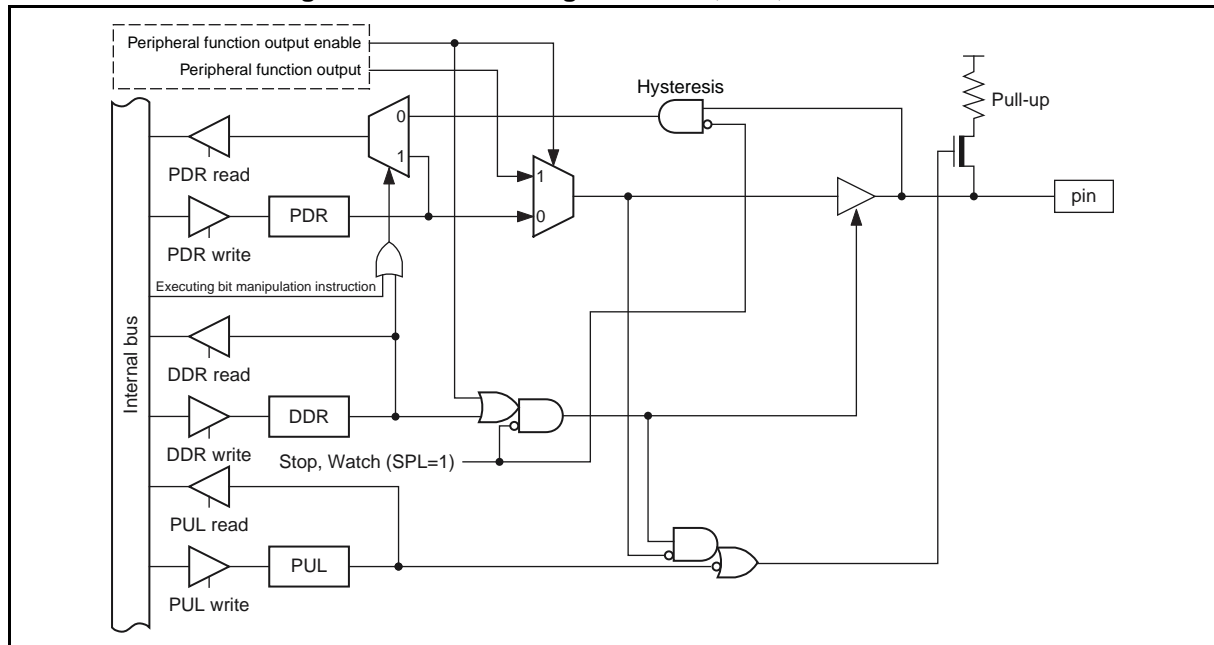
Figure 9.3-2 Block Diagram of P12
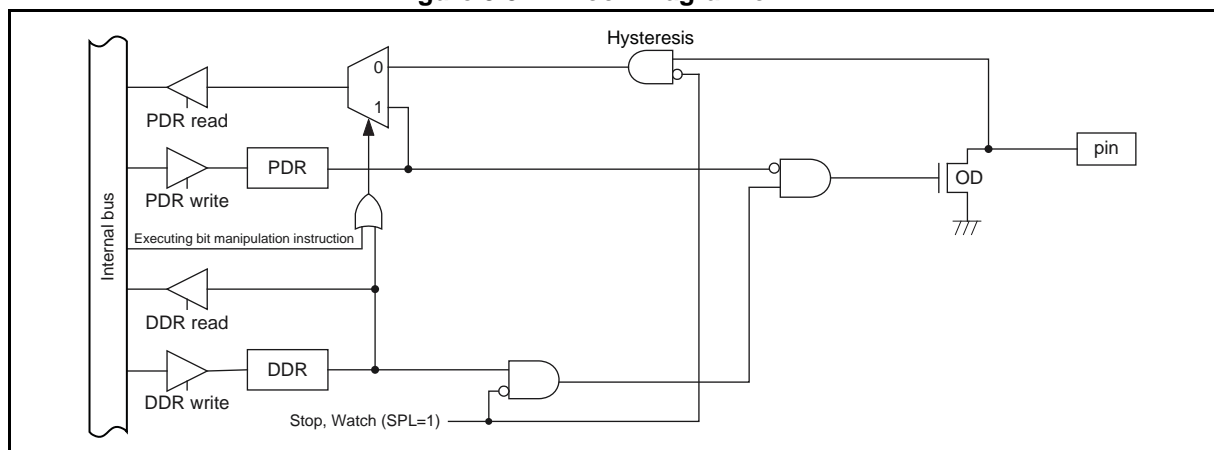
Figure 9.3-3 Block Diagram of P14
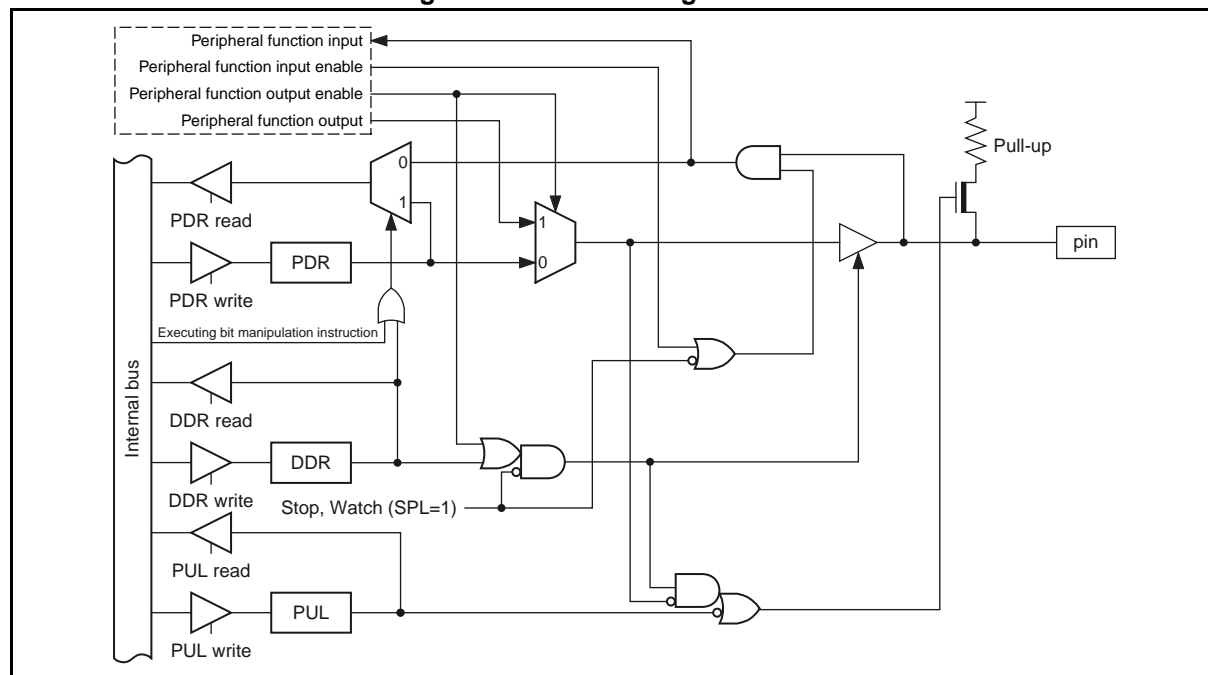
Figure 9.3-4 Block Diagram of P16

MB95330H Series
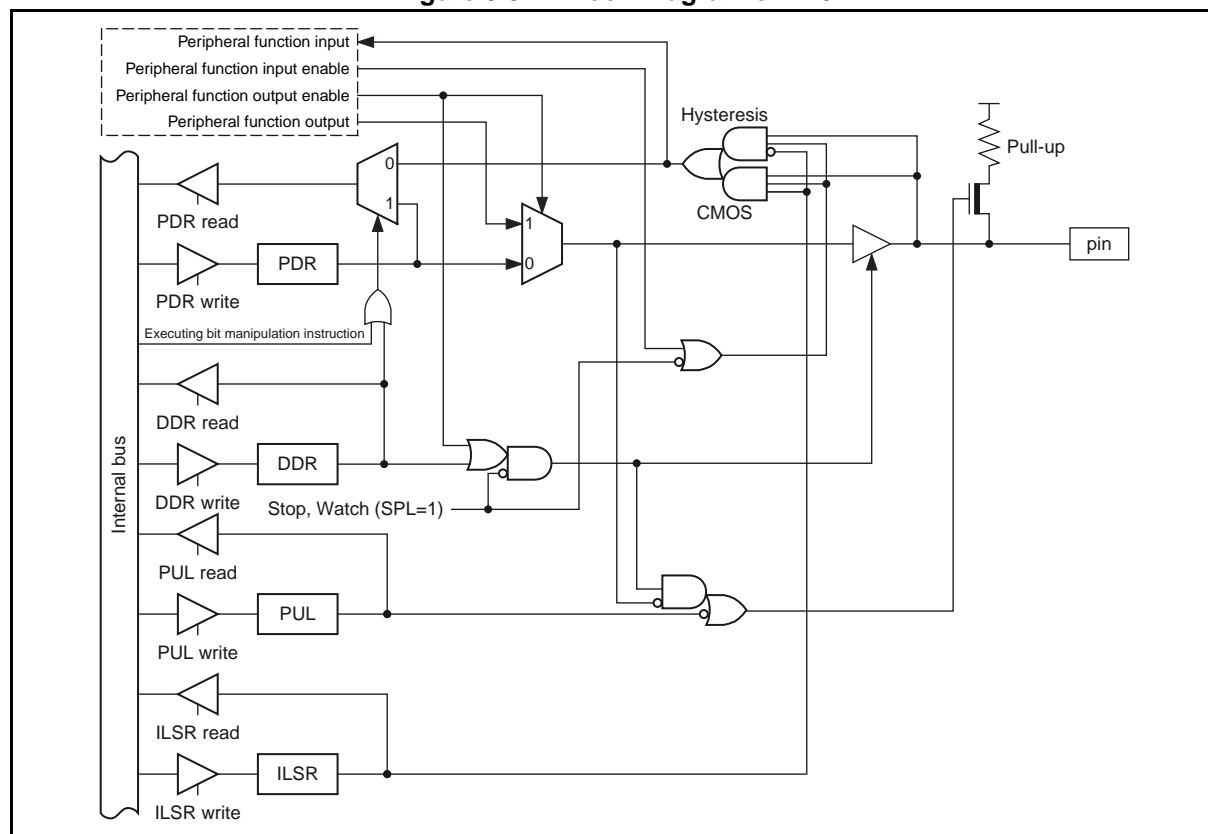
Figure 9.3-5 Block Diagram of P17
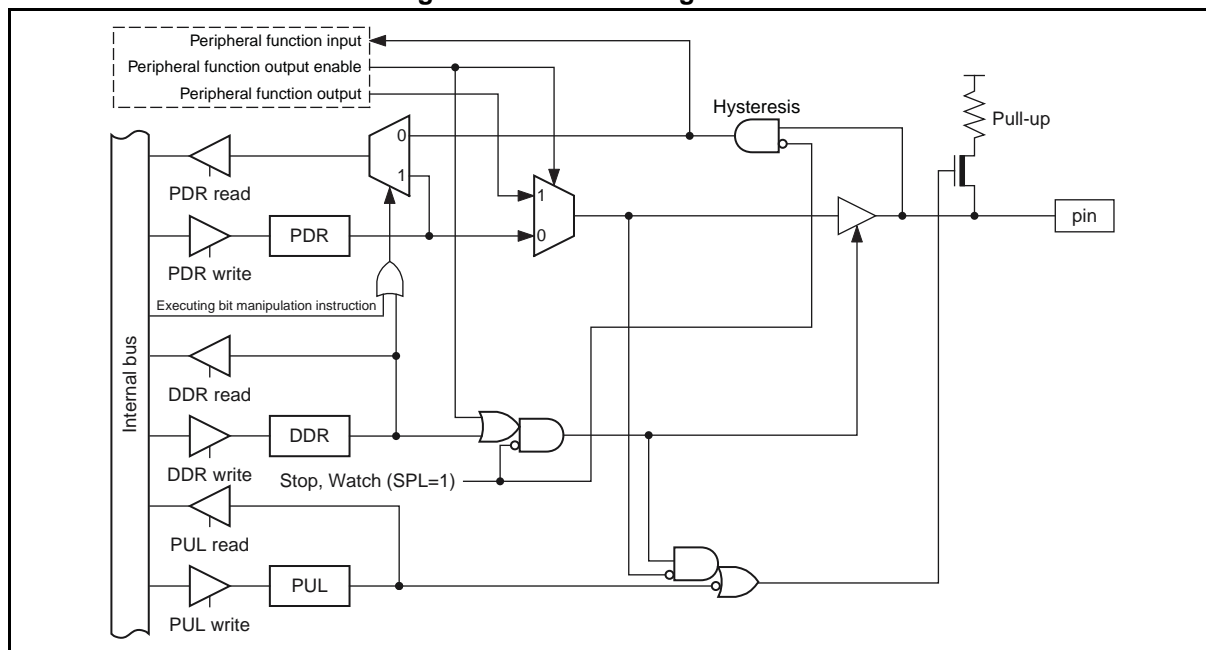
9.3.1   Port 1 Registers

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		
PUL1	0	Pull-up disabled		
	1	Pull-up enabled		
ILSR	0	Hysteresis input level selected		
	1	CMOS input level selected		

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

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

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

Pin name	Correspondence between related register bits and pins							
	P17	P16	P15	P14	P13	P12	P11	P10
PDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR1								
PUL1	bit7	bit6	bit5	bit4	bit3	bit2*	bit1	bit0
ILSR	-	bit3	-	-	-	-	-	-

*: Though P12 has no pull-up function, bit2 in the PUL1 register can still be accessed. The operation of port P12 is not affected by the setting of bit2 in the PUL1 register.

MB95330H Series

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 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 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 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" 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 of P14/UCK0/PPG01 and P16/UI0/PPG21 is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection control register (WICR) of the interrupt pin selection circuit, the input is enabled and will not be 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 of the input level select register

- Setting the bit3 in ILSR to "1" changes only P16 from the hysteresis input level to the CMOS input level. When the same bit is set to "0", the input level of P16 should become the hysteresis input level.

- For pins other than P16, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.

- When changing the input level of P16, ensure that the peripheral function (UART/SIO/ PPG) has been stopped.

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

Table 9.3-4 Pin State of Port 1

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

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

Hi-Z: High impedance

*: "Input enabled" means that the input function is enabled. After a reset, setting the port for internal pull-up or as an output pin is recommended.

MB95330H Series

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)
- 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 (1 / 2)

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P60/INT08/ SDA/DTTI	P60 general-purpose I/O	INT08 external interrupt input	Hysteresis/ CMOS	CMOS	❍	-
		I^2C data I/O				
		MPG waveform sequencer input				
P61/INT09/ SCL/TI1	P61 general-purpose I/O	INT09 external interrupt input	Hysteresis/ CMOS	CMOS	❍	-
		I^2C clock I/O				
		16-bit reload timer ch. 1 input				
P62/TO10/ PPG00/ OPT0	P62 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit composite timer ch. 1 output				
		8/16-bit PPG ch. 0 output				
		MPG waveform sequencer output				
P63/TO11/ PPG01/ OPT1	P63 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit composite timer ch. 1 output				
		8/16-bit PPG ch. 0 output				
		MPG waveform sequencer output				
P64/EC1/ PPG10/ OPT2	P64 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit composite timer ch. 1 clock input				
		8/16-bit PPG ch. 1 output				
		MPG waveform sequencer output				

Table 9.4-1 Port 6 Pins (2 / 2)

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
P65/PPG11/ OPT3	P65 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit PPG ch. 1 output				
		MPG waveform sequencer output				
P66/PPG20/ PPG1/OPT4	P66 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit PPG ch. 2 output				
		16-bit PPG ch. 1 output				
		MPG waveform sequencer output				
P67/PPG21/ TRG1/OPT5	P67 general-purpose I/O	High-current output port	Hysteresis	CMOS	-	-
		8/16-bit PPG ch. 2 output				
		16-bit PPG ch. 1 trigger input				
		MPG waveform sequencer output				

OD: Open drain, PU: Pull-up

MB95330H Series

■ Block Diagrams of Port 6

Figure 9.4-1 Block Diagram of P60 and P61
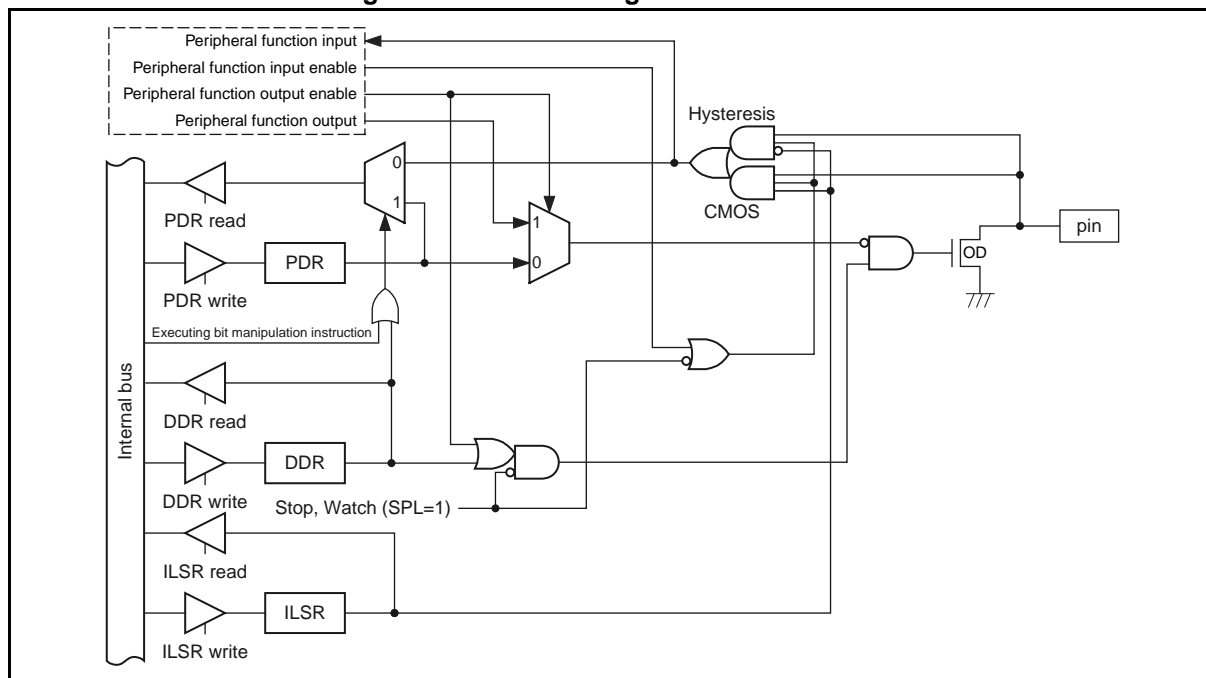
Figure 9.4-2 Block Diagram of P62, P63, P65 and P66
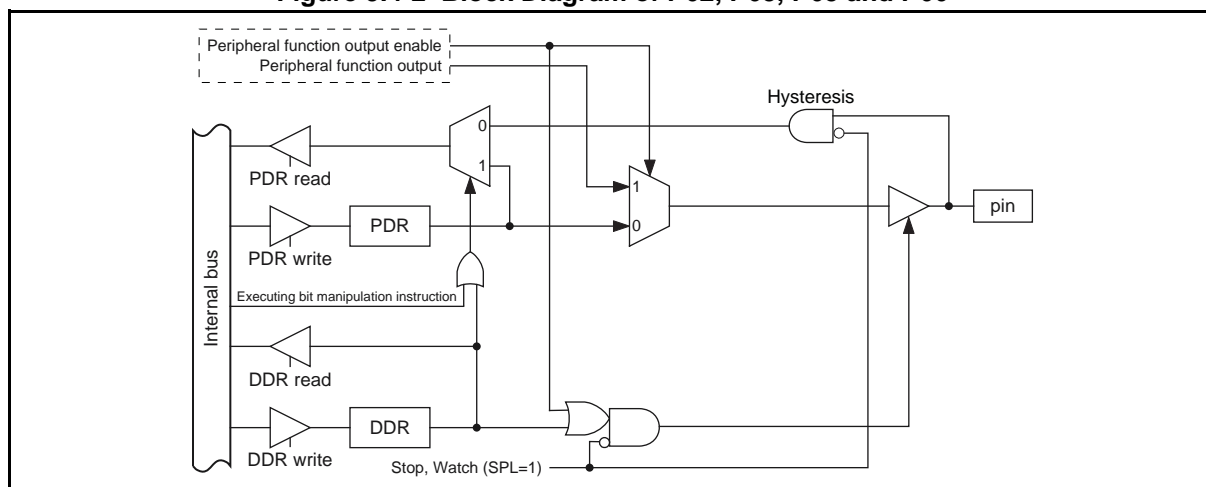
Figure 9.4-3 Block Diagram of P64 and P67
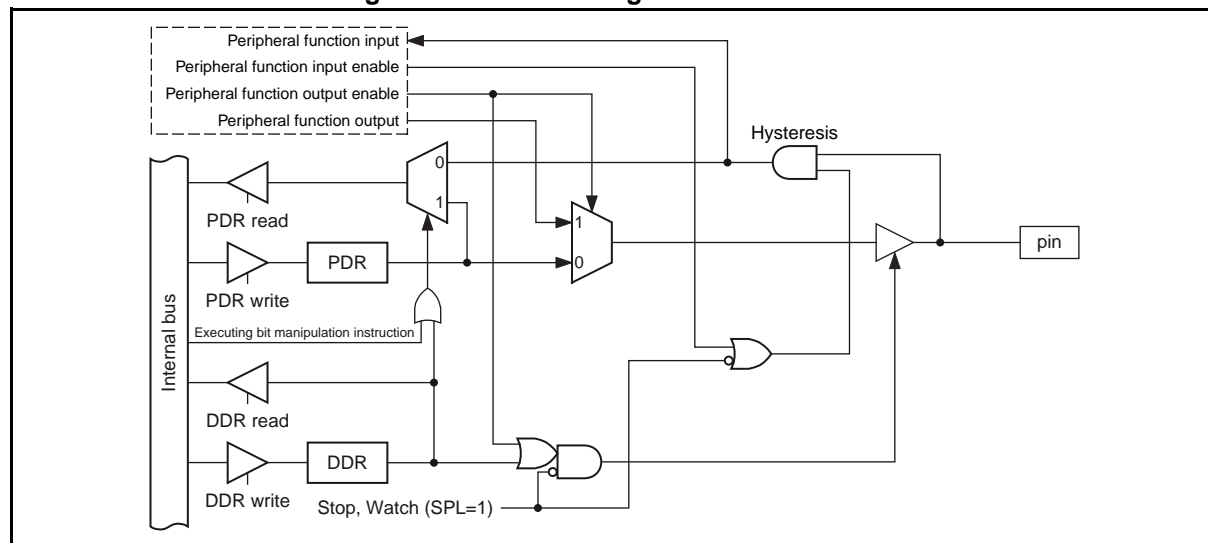
MB95330H Series

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		
ILSR	0	Hysteresis input level selected		
	1	CMOS input level selected		

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

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

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

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

9.4.2 Operations of Port 6

This section describes the operations of port 6.

■ Operations of Port 6

● Operation as an output port

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

● Operation as an input port

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

MB95330H Series

● 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 of P60/INT08/SDA/DTTI and P61/INT09/SCL/TI1, OPT2/PPG10/EC1/P64 and OPT5/PPG21/TRG1/P67 is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection control register (WICR) of the interrupt pin selection circuit, the input is enabled and will not be 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 of the input level select register

- Setting the bit0/1 in ILSR to "1" changes only P60/61 from the hysteresis input level to the CMOS input level. When the bit0/1 in ILSR is set to "0", the input level of P60/61 should become the hysteresis input level.

- For pins other than P60/61, the CMOS input level cannot be selected, but only the hysteresis input level can be selected.

- When changing the input level of P60/61, ensure that the peripheral function (External interrupt/I^2C/MPG/16-bit reload timer) 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 enabled. After a reset, setting the port for internal pull-up or as an output pin is recommended.

9.5 Port F

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

■ Port F Configuration

Port F is made up of the following elements.

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

■ Port F Pins

Port F has three I/O pins.

Table 9.5-1 lists the port F pins.

Table 9.5-1 Port F Pins

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

OD: Open drain, PU: Pull-up

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

[2]: If the external reset is selected (SYSC:RSTEN=1), the port function cannot be used. This pin is a dedicated reset pin in MB95F332H/F333H/F334H.

MB95330H Series

■ Block Diagrams of Port F

Figure 9.5-1 Block Diagram of PF0 and PF1
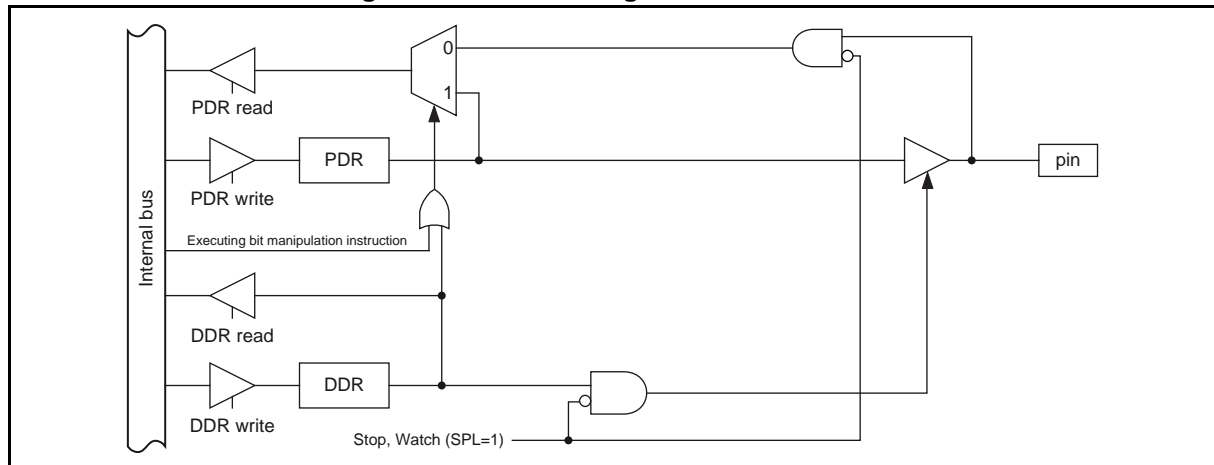
Figure 9.5-2 Block Diagram of PF2
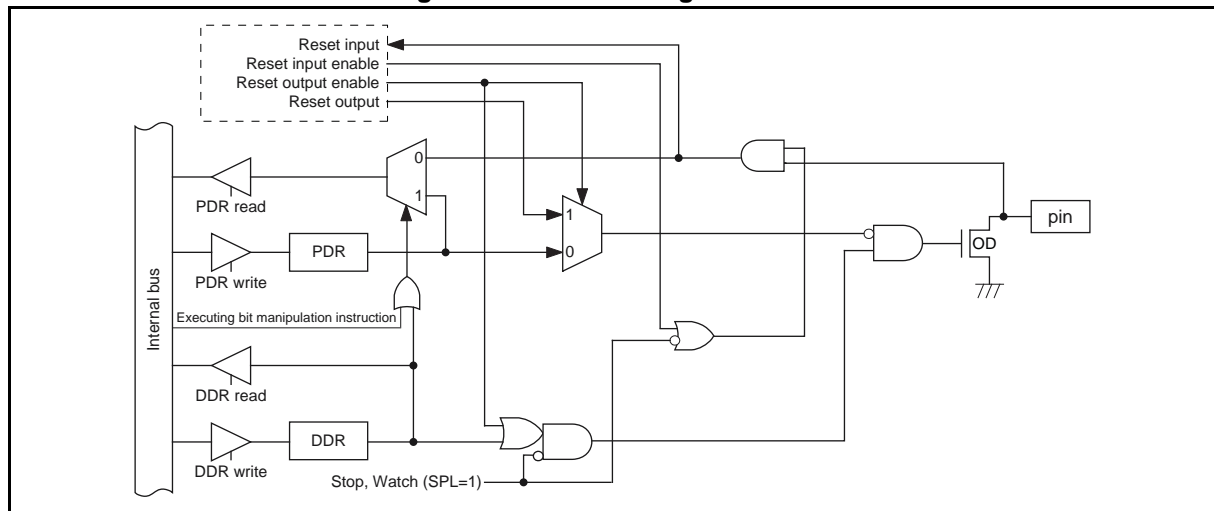
9.5.1   Port F Registers

This section describes the registers of port F.

■ **Port F Register Functions**

Table 9.5-2 lists the port F register functions.

Table 9.5-2 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 the N-ch open drain pin, this should be Hi-Z.

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

Table 9.5-3 Correspondence between Registers and Pins for Port F

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

    *: PF2/$\overline{\text{RST}}$ is a dedicated reset pin in MB95F332H/F333H/F334H.

MB95330H Series

9.5.2 Operations of Port F

This section describes the operations of port F.

■ Operations of Port F

● Operation as an output port

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

● Operation as an input port

- A pin will become an input port if the bit in the DDR register corresponding to that pin is set to "0".
- If data is written to the PDR register, the value is stored in the output latch but is not output to the pin set as an input port.
- Reading the PDR register returns the pin value. However, if the read-modify-write 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 9.5-4 shows the pin states of port F.

Table 9.5-4 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 enabled. After a reset, setting the port for internal pull-up or as an output pin is recommended.

[2]: Only for PF2 at power-on reset.

MB95330H Series

9.6   Port G

Port G is a general-purpose I/O port.

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

■ Port G Configuration

Port G is made up of the following elements.

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

■ Port G Pins

Port G has two I/O pins.

Table 9.6-1 lists the port G pins.

Table 9.6-1 Port G Pins

Pin name	Function	Shared peripheral function	I/O type			
			Input	Output	OD	PU
PG1/X0A[*]/ SNI1	PG1 general-purpose I/O	Subclock oscillation pin	Hysteresis	CMOS	-	❍
		Trigger input for the position detection function of the MPG waveform sequencer				
PG2/X1A[*]/ SNI2	PG2 general-purpose I/O	Subclock oscillation pin	Hysteresis	CMOS	-	❍
		Trigger input for the position detection function of the MPG waveform sequencer				

OD: Open drain, PU: Pull-up

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

■ Block Diagram of Port G

Figure 9.6-1 Block Diagram of PG1 and PG2

MB95330H Series
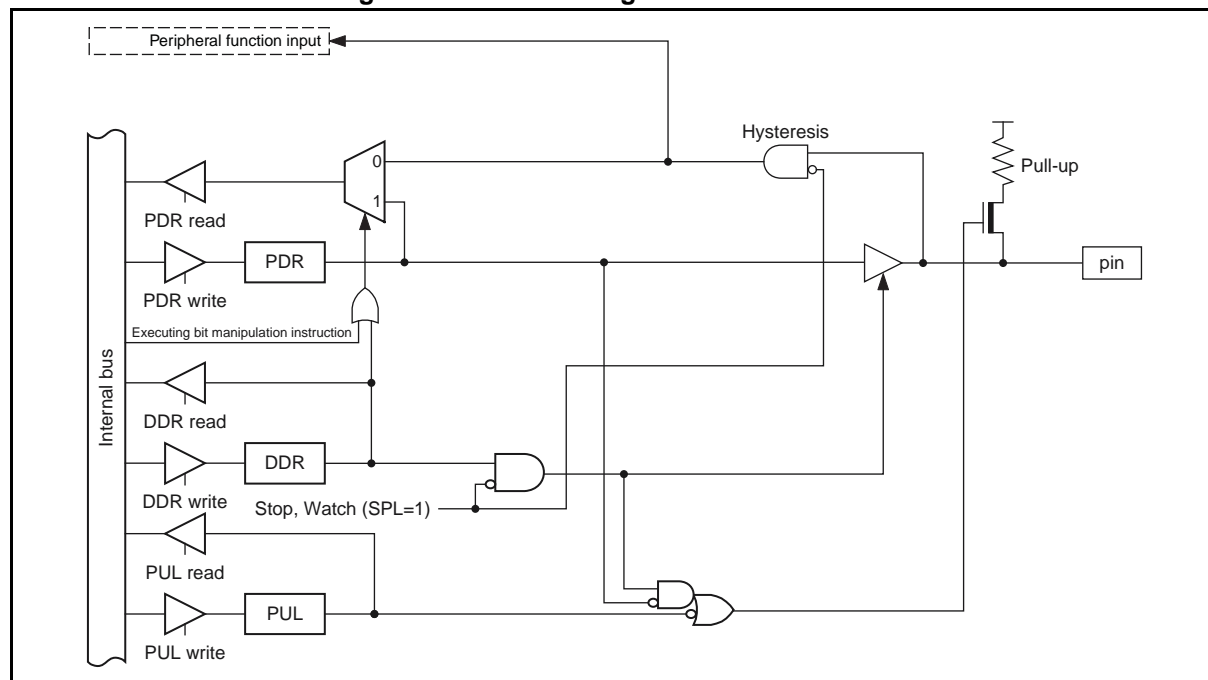
9.6.1   Port G Registers

This section describes the registers of port G.

■ Port G Register Functions

Table 9.6-2 lists the port G register functions.

Table 9.6-2 Port G Register Functions

Register 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 9.6-3 lists the correspondence between port G pins and each register bit.

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

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

9.6.2 Operations of Port G

This section describes the operations of port G.

■ Operations of Port G

● Operation as an output port

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

● Operation as an input port

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

MB95330H Series

● Operation of the pull-up register

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

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

Table 9.6-4 Pin State of Port G

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

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

Hi-Z: High impedance

*: "Input enabled" means that the input function is enabled. After a reset, setting the port for internal pull-up or as an 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 RCM1 bit and RCM0 bit 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 whenever a selected interval time elapses.

- The length of an interval time can be selected from the following 16 values.

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

MB95330H 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
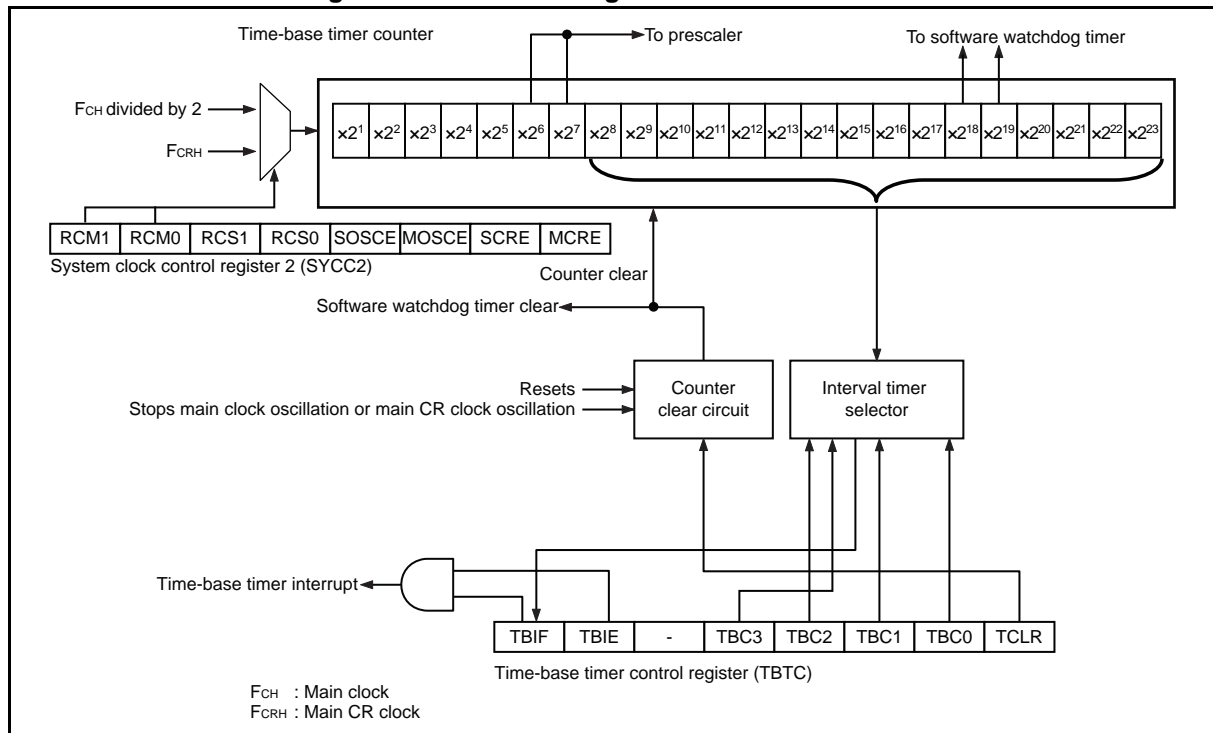
● Time-base timer counter

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 status 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.

MB95330H Series

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
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	

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

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

TBIE	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)

Bit name		Function
bit7	TBIF: Time-base timer interrupt request flag bit	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". **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	TBIE: Time-base timer interrupt request enable bit	This bit enables/disables the output of interrupt requests to the interrupt controller. **Writing "0"**: disables the output of time-base timer interrupt requests. **Writing "1"**: 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".
bit5	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit4 to bit1	TBC3 to TBC0: Interval time select bits	These bits select interval time. (see table below)
bit0	TCLR: Time-base timer initialization bit	This bit clears the time-base timer counter. **Writing "0"**: is ignored and has no effect on the operation. **Writing "1"**: initializes all counter bits to "1". When this bit is read, it always returns "0". Note: When the output of the 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.

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

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". With the TBIF bit set to "1", if the time-base timer interrupt request enable bit is also enabled (TBTC:TBIE = 1), an interrupt request (IRQ19) 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 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 clearing of a counter (TBTC:TCLR = 1) and the underflow of the time-base timer counter occur simultaneously.
- 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

MB95330H Series

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

Table 10.4-2 Register and Vector Table Addresses Related to 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

MB95330H Series

■ 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 to TBC0) to "0011$_B$" ($2^{16} \times 2/F_{CH}$)

- TBTC:TBC3 to 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 to MWT0 : Main clcok 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 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)

MB95330H Series

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:CS1, CS0 = 00_B or 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 used 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 to be 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 (NVR) FUNCTION" 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 kHz to 200 kHz)	$XX1_B$ (SWWDT) or HWWDT[1]	328 ms	2.62 s

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

MB95330H Series

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
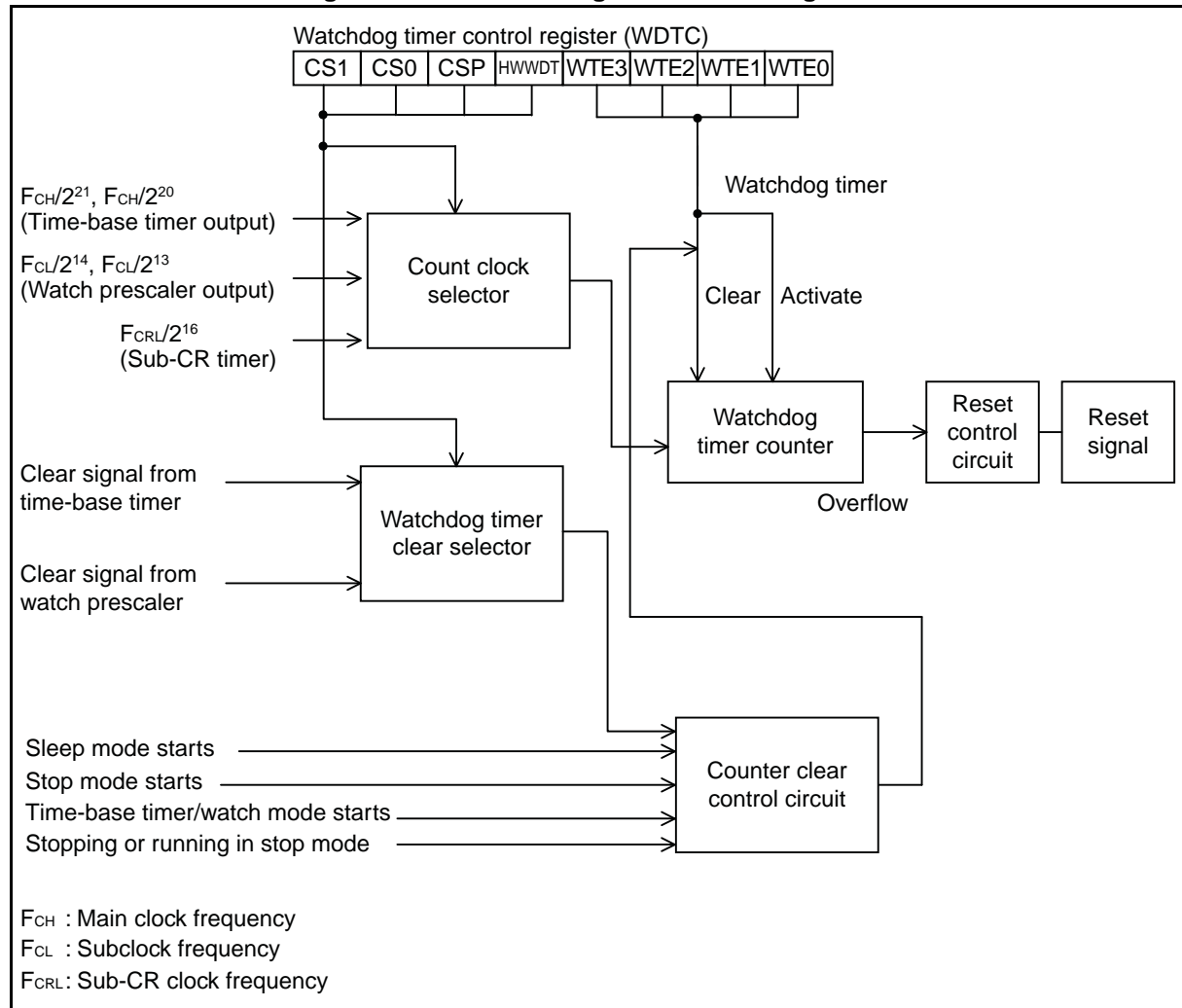
● Count clock selector

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).

MB95330H Series

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,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.
R1/WX	: The read value is "1". Writing a value to it has no effect on operation.

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.
X   : Don't care
▨   : 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)

Bit name		Function			
bit7, bit6	CS1, CS0: Count clock switch bits	These bits select the count clock of the watchdog timer.			
bit5	CSP: Count clock select sub-CR selector bit	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 ($2^{16}/F_{CRL}$)
		• 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 is to be stopped in subclock mode, always select the output of the watch prescaler in subclock mode.			
bit4	HWWDT: Hardware watchdog timer activation bit	The bit is a read-only bit, used to confirm the start/stop of the hardware watchdog timer. **"1"**: The hardware watchdog timer has been activated. **"0"**: The hardware watchdog timer has stopped (The software watchdog timer can be activated).			
bit3 to bit0	WTE3, WTE2, WTE1, WTE0: Watchdog control bits	These bits are used to control the watchdog timer. **Writing "0101$_B$":** activates the watchdog timer (at the first write access after a reset) or clears it (from the second write access after a reset). **Writing other than "0101$_B$":** has no effect on operation. • When these bits are read, they always return "0000$_B$".			

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. 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 (NVR) FUNCTION" for details of the watchdog timer selection ID.

- Start operation after a reset.

- CS1,CS0,CSP bits are read-only bits, fixed at "001_B".

- The timer is cleared by a reset and resumes operation after the reset is released.

● Clearing the watchdog timer

- When the counter of the watchdog timer is not cleared within the interval time, it overflows, allowing the watchdog timer to generate a watchdog reset.

- The counter of the hardware watchdog timer is cleared when "0101_B" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0). The counter of the software watchdog timer is cleared when "0101_B" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0) for the second time and from the second time onward.

- The watchdog timer is cleared at the same time as the timer selected as the count clock (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 relation between the timing of clearing the watchdog timer and the interval time when the time-base timer output $F_{CH}/2^{21}$ (F_{CH}: main clock) is selected as the count clock (main clock = 4 MHz).

Figure 11.4-1 Clearing Timing and Interval Time of Watchdog Timer
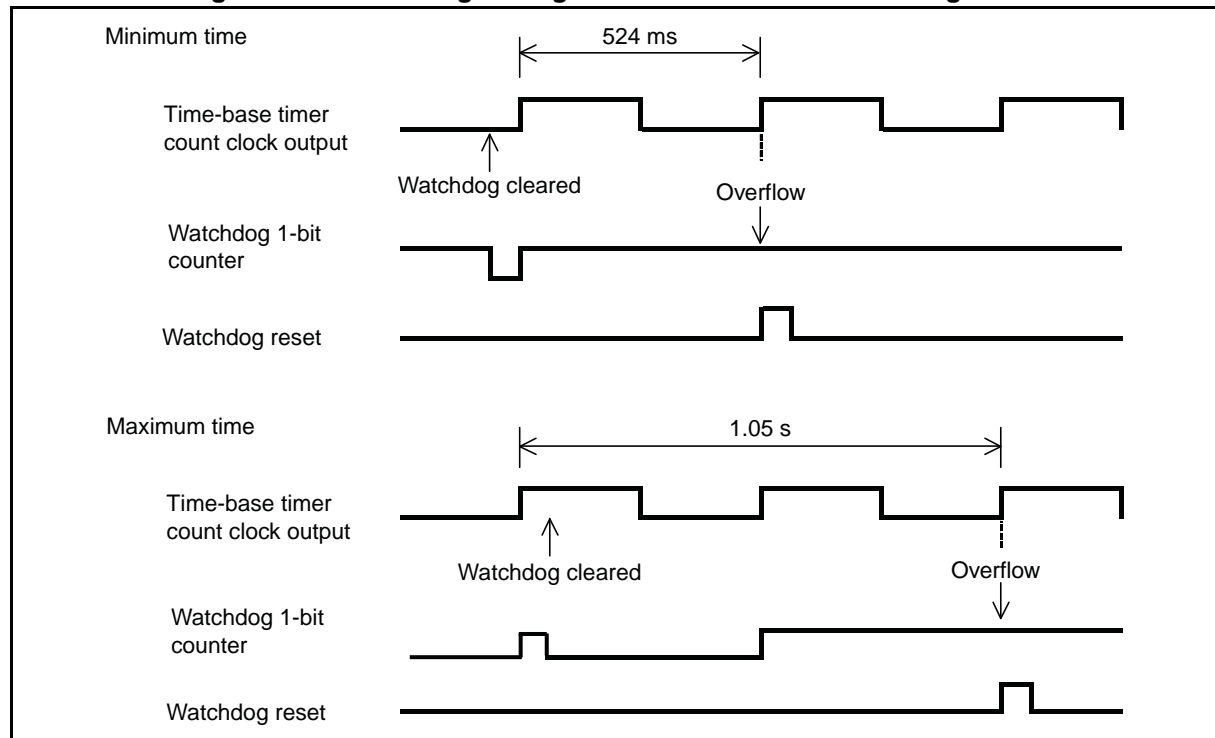
● Operation in subclock mode

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

■ Setting Procedure Example

Below is the procedure for setting the software watchdog timer.

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

Below is the procedure for setting the hardware watchdog timer.

1) Write "A597$_H$" (the hardware watchdog time is enabled except in standby mode) or any other value (the hardware watchdog timer is enabled in every mode) except "A596$_H$" and "A597$_H$" to the addresses FFBE$_H$ and FFBF$_H$ on the Flash memory, which are copied to the watchdog timer selection ID registers WDTH/WDTL (0FEB$_H$/0FEC$_H$). See CHAPTER 30 "NON-VOLATILE REGISTER (NVR) FUNCTION" for details of the watchdog timer selection ID registers.

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

MB95330H Series

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 or the sub-CR clock 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 MB95330H Series for the accuracy of the sub-CR clock frequency.

MB95330H 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

F_{CL} : Subclock
F_{CRL}: Sub-CR clock
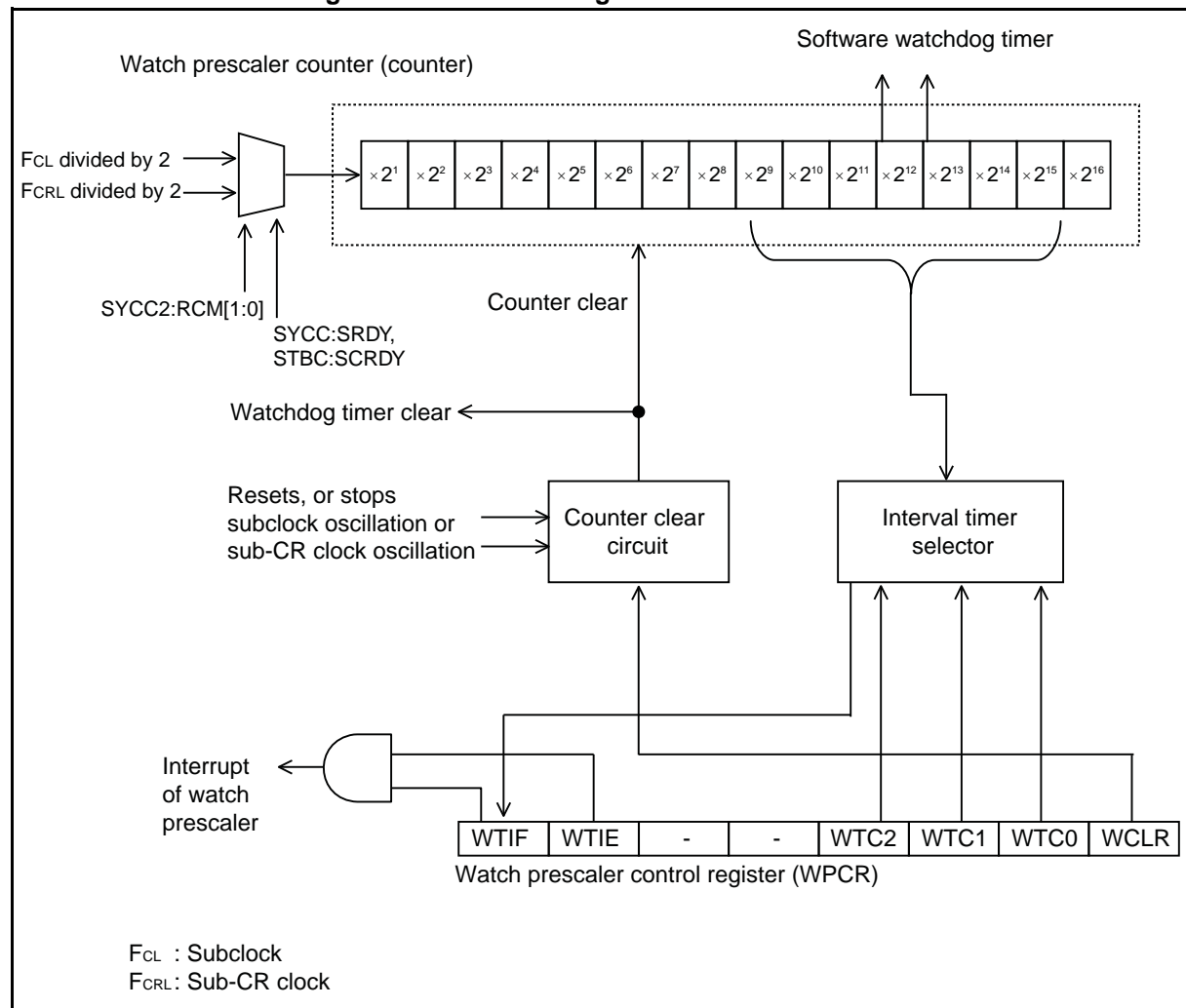
● Watch prescaler counter (counter)

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.

MB95330H Series

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
000B$_H$	WTIF	WTIE	-	-	WTC2	WTC1	WTC0	WCLR	00000000$_B$
	R(RM1),W	R/W	R0/WX	R0/WX	R/X	R/W	R/W	R0,W	

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

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

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

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

R/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)

Bit name		Function
bit7	WTIF: Watch interrupt request flag bit	This bit becomes "1" when the selected interval time of the watch prescaler has elapsed. • An interrupt request is generated when this bit and the interrupt request enable bit (WTIE) are set to "1". **Writing "0"**: sets this bit to "0". **Writing "1"**: is ignored and has no effect on operation. • When read by the read-modify-write (RMW) type of instruction, this bit always returns "1".
bit6	WTIE: Interrupt request enable bit	This bit enables or disables the output of interrupt requests to interrupt controller. **Writing "0"**: disables the interrupt request output of the watch prescaler. **Writing "1"**: enables the interrupt request output of the watch prescaler. An interrupt request is output when this bit and the watch interrupt request flag bit (WTIF) are set to "1".
bit5, bit4	Undefined bits	These bits are undefined. • When this bit is read, it always returns "0". • Writing a value to this bit has no effect on operation.
bit3 to bit1	WTC2 to WTC0: Watch interrupt interval time select bits	These bits select the interval time.

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

Bit name		Function
bit0	WCLR: Watch timer initialization bit	This bit clears the counter for the watch prescaler. **Writing "0"**: is ignored and has no effect on operation. **Writing "1"**: initializes all counter bits to "1". When this bit is read, it always returns "0". Note: When the output of the watch prescaler is selected as the count clock of the software watchdog timer, clearing the watch prescaler with this bit also clears the software watchdog timer.

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

MB95330H Series

■ 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 (WPCR: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 register 2 (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.

MB95330H Series

Figure 12.5-1 Watch Prescaler Operation Example

- When setting interval time select bits in the watch prescaler control register (WPCR:WTC2 to WTC0) to "011$_B$" ($2^{14} \times 2/F_{CL}$)

- WPCR:WTC2 to WTC0 : Interval time select bits in watch prescaler control register
- WPCR:WCLR : Watch timer initialization bit in watch prescaler control register
- WPCR:WTIF : Watch interrupt request flag bit in watch prescaler control register
- WPCR:WTIE : Watch interrupt request enable bit in watch prescaler control register
- STBC:SLP : Sleep bit in standby control register
- STBC:STP : Stop bit in standby control register
- WATR:SWT3 to SWT0 : Subclock oscillation stabilization wait time select bits in oscillation stabilization wait time setting register

■ Setting Procedure Example

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

● Initial settings

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

● Processing interrupts

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

12.6   Notes on Using Watch Prescaler

This section provides notes on using the watch prescaler.

■ Notes on Using Watch Prescaler

● When setting interrupt processing in a program

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

● Clearing the watch prescaler

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

● Watch interrupts

In stop mode in which the main clock is used, the watch prescaler performs counting and 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.

MB95330H Series

12.7 Sample Settings for Watch Prescaler

This section provides sample settings for the watch prescaler.

■ Sample Settings

● How to initialize the watch prescaler

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

Operation	Watch timer initialization bit (WCLR)
To initialize the watch prescaler	Set the bit to "1"

● 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.

Operation	Interrupt request enable bit (WTIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

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

Operation	Watch interrupt request flag (WTIF)
To clear an interrupt request	Set the bit to "0".

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.

MB95330H Series

13.2   Configuration of Wild Register Function

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

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

■ Block Diagram of Wild Register Function

Figure 13.2-1 Block Diagram of Wild Register Function
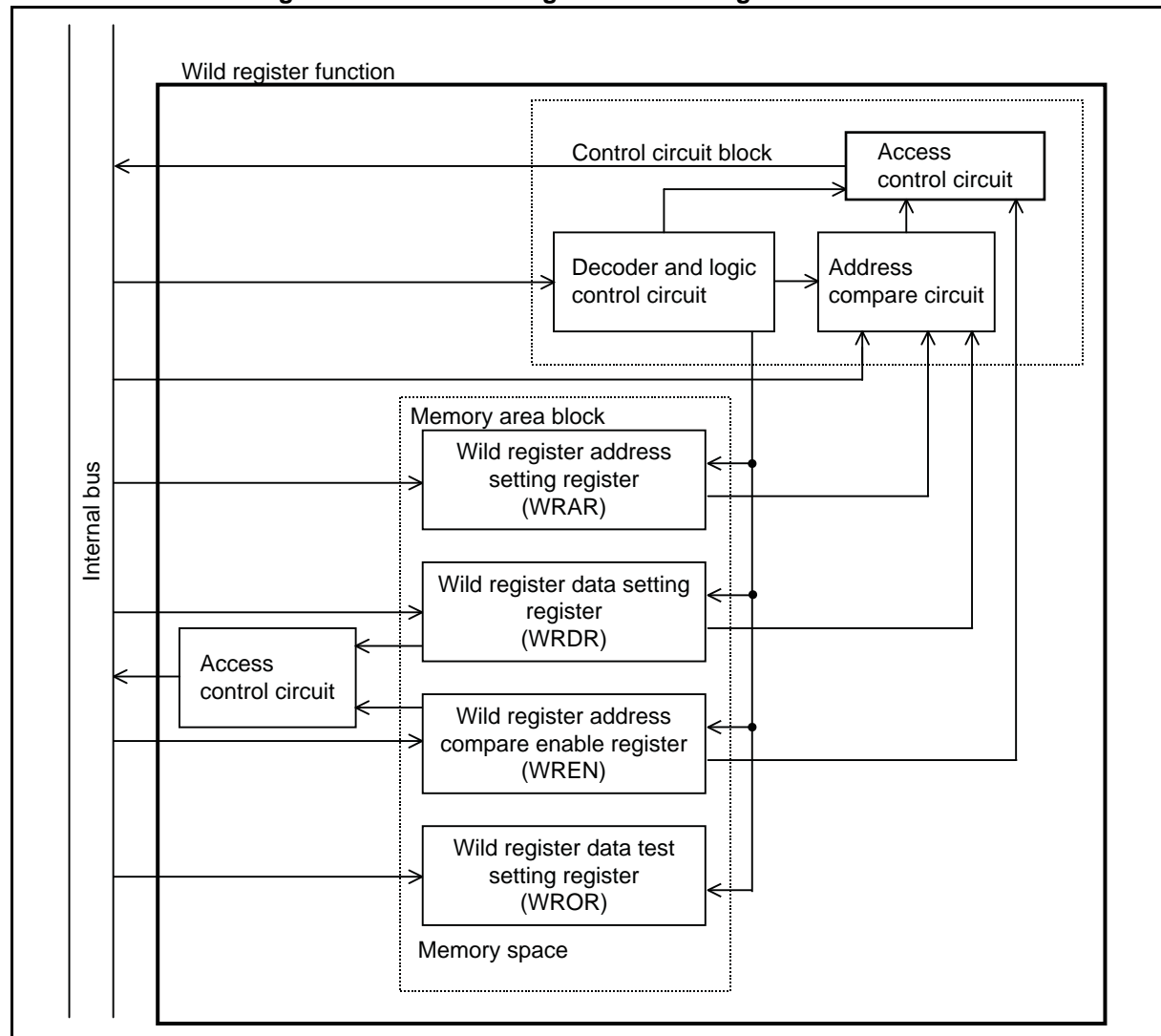
● Memory area block

The memory area block consists of the wild register data setting registers (WRDR), wild register address setting registers (WRAR), wild register address compare enable register

(WREN) and wild register data test setting register (WROR). The wild register function is used to specify the addresses and data that need to be replaced. The wild register address compare enable register (WREN) enables the wild register function for each wild register data setting register (WRDR). In addition, the wild register data test setting register (WROR) enables the normal read function for each wild register data setting register (WRDR).

● Control circuit block

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

MB95330H Series

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

MB95330H Series

13.3.1 Wild Register Data Setting Registers (WRDR0 to WRDR2)

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

■ 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) are used to 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.

MB95330H Series

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 bits. • 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 bits. • 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.

MB95330H Series

13.4 Operations of Wild Register Function

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

■ Procedure for Setting Wild Register Function

Prepare a program that can read the value to be set in the wild register from external memory (e.g. E^2PROM or FRAM) in the user program before using the wild register function. The setting method for the wild register is shown below.

This section does not include information on the method of communications between the external memory and the device.

- Write the address of the built-in ROM code that will be modified to the wild register address setting register (WRAR0 to WRAR2).
- Write a new code to the wild register data setting register (WRDR0 to WRDR2) corresponding to the wild register address setting register to which the address has been written.
- Write "1" to the EN bit in the wild register address compare enable register (WREN) corresponding to the wild register number to enable the wild register function represented by that wild register number.

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

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

Step	Operation	Operation example
1	Read replacement data from a peripheral function outside through a certain communication method.	If the built-in ROM code to be modified is at the address $F011_H$ and the data to be modified is $B5_H$, 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 using the wild register function.

■ Hardware Connection Example

Figure 13.5-1 Typical Hardware Connection Example
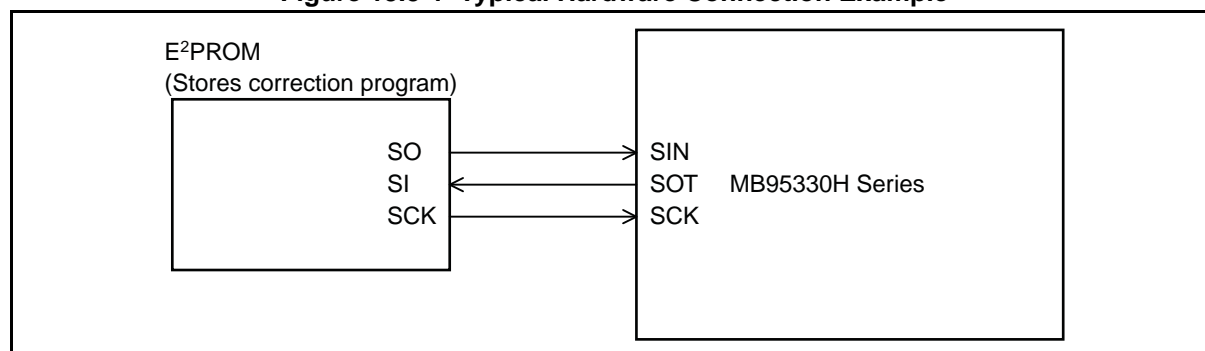
CHAPTER 14
8/16-BIT COMPOSITE TIMER

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

14.1   Overview of 8/16-bit Composite Timer

The 8/16-bit composite timer consists of two 8-bit counters. It can be used as two 8-bit timers, or as a 16-bit timer if the two counters are connected in cascade.
The 8/16-bit composite timer has the following functions:
- **Interval timer function**
- **PWM timer function**
- **PWC timer function (pulse width measurement)**
- **Input capture function**

■ Interval Timer Function (One-shot Mode)

When the interval timer function (one-shot mode) is selected, the counter starts counting from "00_H" as the timer is started. When the counter value matches the value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted, an interrupt request occurs, and the counter stops counting.

■ Interval Timer Function (Continuous Mode)

When the interval timer function (continuous mode) is selected, the counter starts counting from "00_H" as the timer is started. When the counter value matches the value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted, an interrupt request occurs, and the counter counts from "00_H" again. The timer outputs square wave as a result of this repeated operation.

■ Interval Timer Function (Free-run Mode)

When the interval timer function (free-run mode) is selected, the counter starts counting from "00_H". When the counter value matches the value of the 8/16-bit composite timer 00/01 data register, the timer output is inverted and an interrupt request occurs. Under these conditions, if the counter continues to count and reaches "FF_H", it restarts counting from "00_H". The timer outputs square wave as a result of this repeated operation.

■ PWM Timer Function (Fixed-cycle Mode)

When the PWM timer function (fixed-cycle mode) is selected, a PWM signal with a variable "H" pulse width is generated in fixed cycles. The cycle is fixed at "FF_H" in 8-bit operation or at "$FFFF_H$" in 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by setting a specific register.

■ PWM Timer Function (Variable-cycle Mode)

When the PWM timer function (variable-cycle mode) is selected, two 8-bit counters are used to generate an 8-bit PWM signal of variable cycle and duty depending on the cycle and "L" pulse width specified by registers.

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

MB95330H Series

■ 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 $\times$ 2 channels**
- **8-bit comparator (including a temporary latch) $\times$ 2 channels**
- **8/16-bit composite timer 00/01 data register $\times$ 2 channels (T00DR/T01DR), (T10DR/T11DR)**
- **8/16-bit composite timer 00/01 status control register 0 $\times$ 2 channels (T00CR0/ T01CR0), (T10CR0/T11CR0)**
- **8/16-bit composite timer 00/01 status control register 1 $\times$ 2 channels (T00CR1/ T01CR1), (T10CR1/T11CR1)**
- **8/16-bit composite timer 00/01 timer mode control register (TMCR0), (TMCR1)**
- **Output controller $\times$ 2 channels**
- **Control logic $\times$ 2 channels**
- **Count clock selector $\times$ 2 channels**
- **Edge detector $\times$ 2 channels**
- **Noise filter $\times$ 2 channels**

MB95330H Series

■ Block Diagram of 8/16-bit Composite Timer

Figure 14.2-1 Block Diagram of 8/16-bit Composite Timer

Note: Names in parentheses are those used in timer 10 and timer 11.
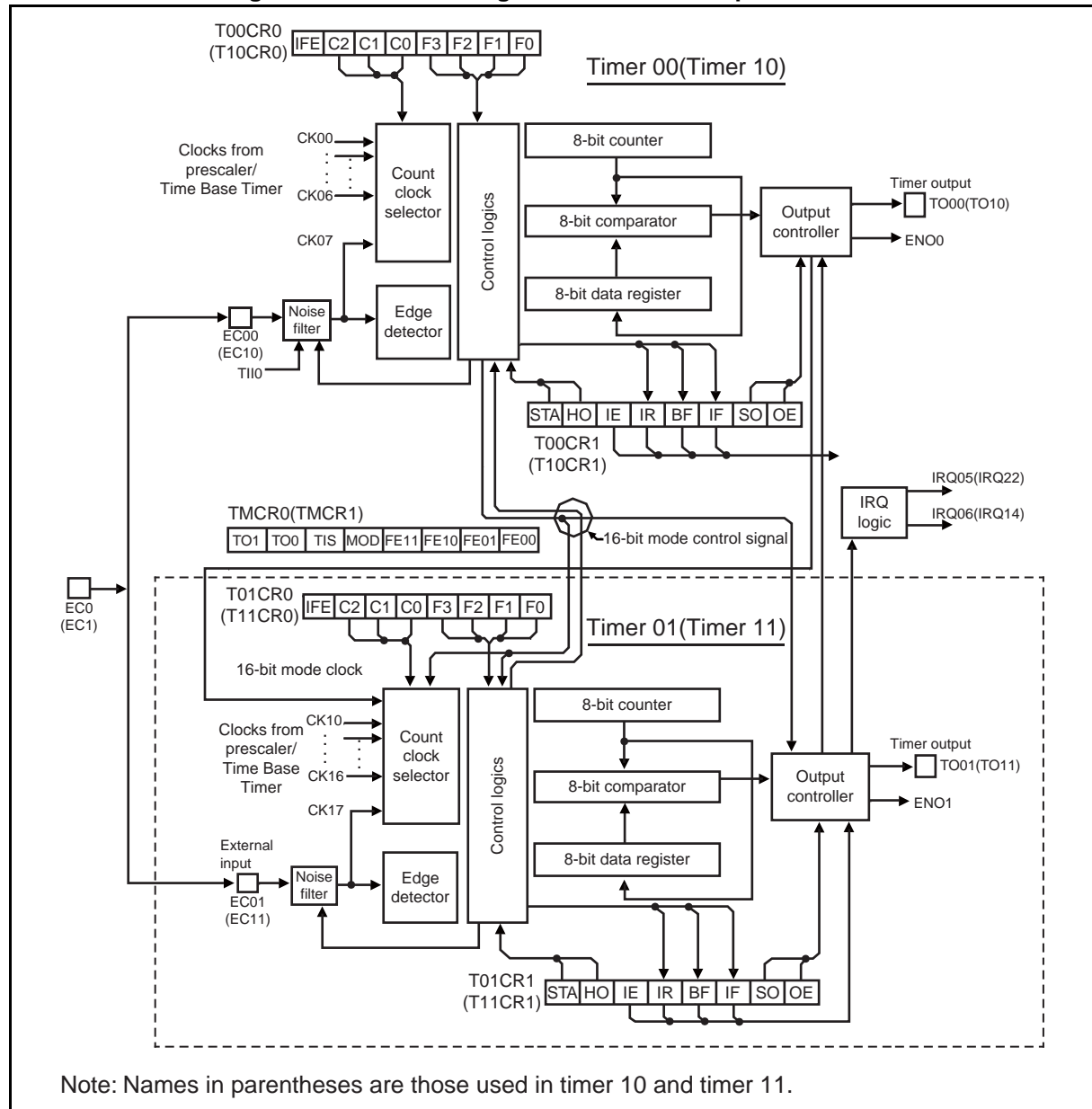
● 8-bit counter

> This counter serves as the basis for various timer operations. It can be used either as two 8-bit counters or as a 16-bit counter.

● 8-bit comparator

> The comparator compares the value in the 8/16-bit composite timer 00/01 data register and that in the counter. It incorporates a latch that temporarily stores the 8/16-bit composite timer 00/01 data register value.

● 8/16-bit composite timer 00/01 data register (T00DR/T01DR) [8/16-bit composite timer 10/
  11 data register (T10DR/T11DR)]

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) [8/16-bit
  composite timer 10/11 status control registers 0 (T10CR0/T11CR0) in timer 10/11]

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) [8/16-bit
  composite timer 10/11 status control registers 1 (T10CR1/T11CR1) in timer 10/11]

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) [8/16-bit composite
  timer 10/11 timer mode control register (TMCR1) in timer 10/11]

This register is used to select the noise filter function, 8-bit or 16-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.

● TII0 internal pin (internally connected to the LIN-UART, available on channel 0)

The TII0 pin serves as the signal input pin for timer 00 on channel 0. Nonetheless, it is connected to the LIN-UART inside the chip. For information about how to use the pin, see CHAPTER 17 "LIN-UART". In addition, the TII0 pin for timer 00 on channel 1 is internally fixed to "0".

MB95330H Series

■ Input Clock

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

14.3 Channels of 8/16-bit Composite Timer

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

■ Channels of 8/16-bit Composite Timer

The MB95330H Series has two channels of 8/16-bit composite timer.

In a channel, there are two 8-bit counters. They can be used as two 8-bit timers or one 16-bit timer.

Table 14.3-1 shows the external pins of each channel and Table 14.3-2 the registers of each channel.

Table 14.3-1 8/16-bit Composite Timer Channels and Corresponding External Pins

Channel	Pin name	Pin function
0	TO00	Timer 00 output
	TO01	Timer 01 output
	EC0	Timer 00 input and timer 01 input
1	TO10	Timer 10 output
	TO11	Timer 11 output
	EC1	Timer 10 input and timer 11 input

Table 14.3-2 8/16-bit Composite Timer Channels and Corresponding Registers

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	T00CR0	8/16-bit composite timer 00 status control register 0
	T01CR0	8/16-bit composite timer 01 status control register 0
	T00CR1	8/16-bit composite timer 00 status control register 1
	T01CR1	8/16-bit composite timer 01 status control register 1
	T00DR	8/16-bit composite timer 00 data register
	T01DR	8/16-bit composite timer 01 data register
	TMCR0	8/16-bit composite timer 00/01 timer mode control register
1	T10CR0	8/16-bit composite timer 10 status control register 0
	T11CR0	8/16-bit composite timer 11 status control register 0
	T10CR1	8/16-bit composite timer 10 status control register 1
	T11CR1	8/16-bit composite timer 11 status control register 1
	T10DR	8/16-bit composite timer 10 data register
	T11DR	8/16-bit composite timer 11 data register
	TMCR1	8/16-bit composite timer 10/11 timer mode control register

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

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

MB95330H Series

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, TO10, TO11, EC0 and EC1. TII0 is for internal chip connection.

● TO00 pin

TO00:

This pin serves as the timer output pin for timer 00 in 8-bit operation or for timers 00 and 01 in 16-bit operation. When the output is enabled (T00CR1:OE = 1) in the interval timer function, PWM timer function, or PWC timer function, this pin becomes an output pin automatically regardless of the port direction register (DDR0:bit5) 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:bit6) 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.

● TO10 pin

TO10:

This pin serves as the timer output pin for timer 10 in 8-bit operation or for timers 10 and 11 in 16-bit operation. When the output is enabled (T10CR1: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 (DDR6:bit2) and functions as the timer output TO10 pin.

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

● TO11 pin

TO11:

This pin serves as the timer output pin for timer 11 in 8-bit operation. When the output is enabled (T11CR1: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 (DDR6:bit3) and functions as the timer output TO11 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.

● EC1 pin

The EC1 pin is connected to the EC10 and EC11 internal pins.

EC10 internal pin:

This pin serves as the external count clock input pin for timer 10 when the interval timer function or PWM timer function is selected, or as the signal input pin for timer 10 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 EC1 pin to "0" to make the pin as an input port.

EC11 internal pin:

This pin serves as the external count clock input pin for timer 11 when the interval timer function or PWM timer function is selected, or as the signal input pin for timer 11 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 EC1 pin to "0" to make the pin as an input port.

MB95330H Series

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

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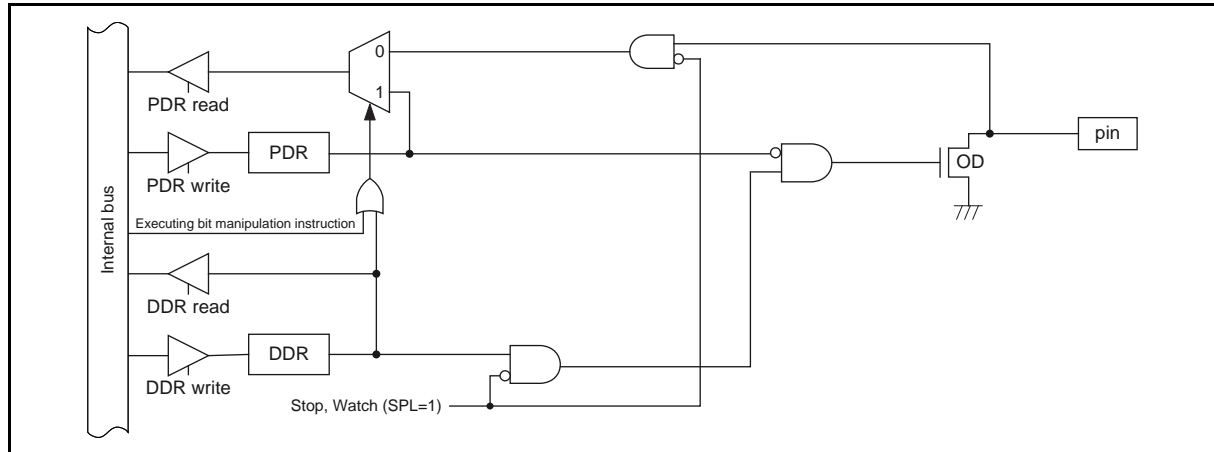

Figure 14.4-2 Block Diagram of Pin EC0 (P04/INT04/AN04/SIN/HCLK1/EC0) of 8/16-bit Composite Timer
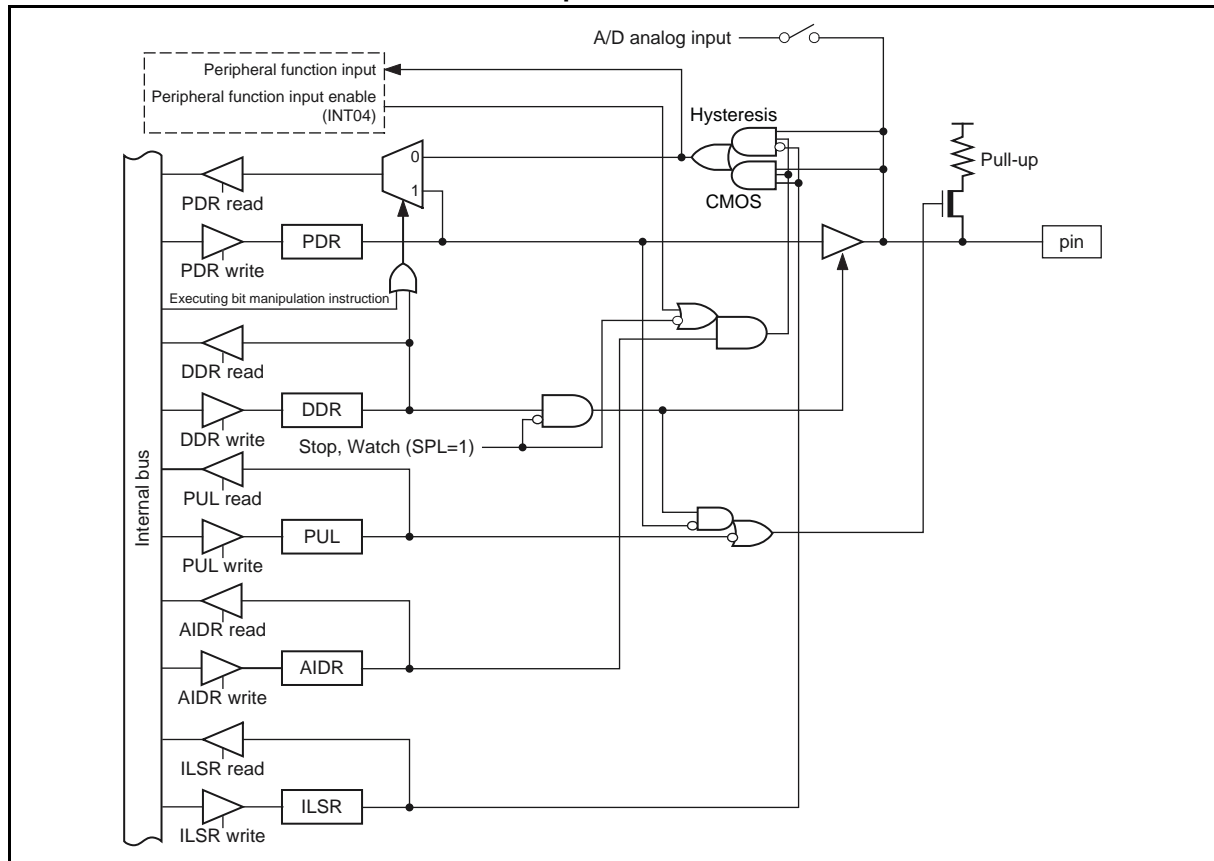
Figure 14.4-3 Block Diagram of Pin TO00 (P05/INT05/AN05/TO00/HCLK2) of 8/16-bit Composite Timer
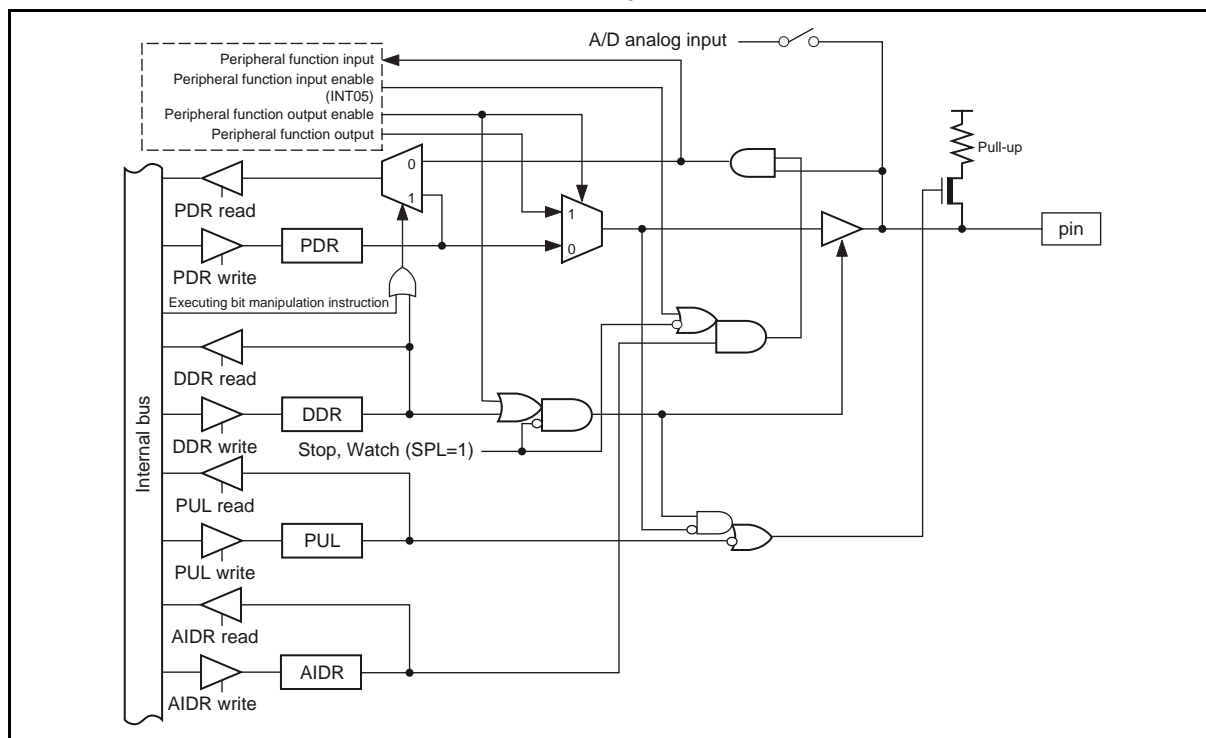

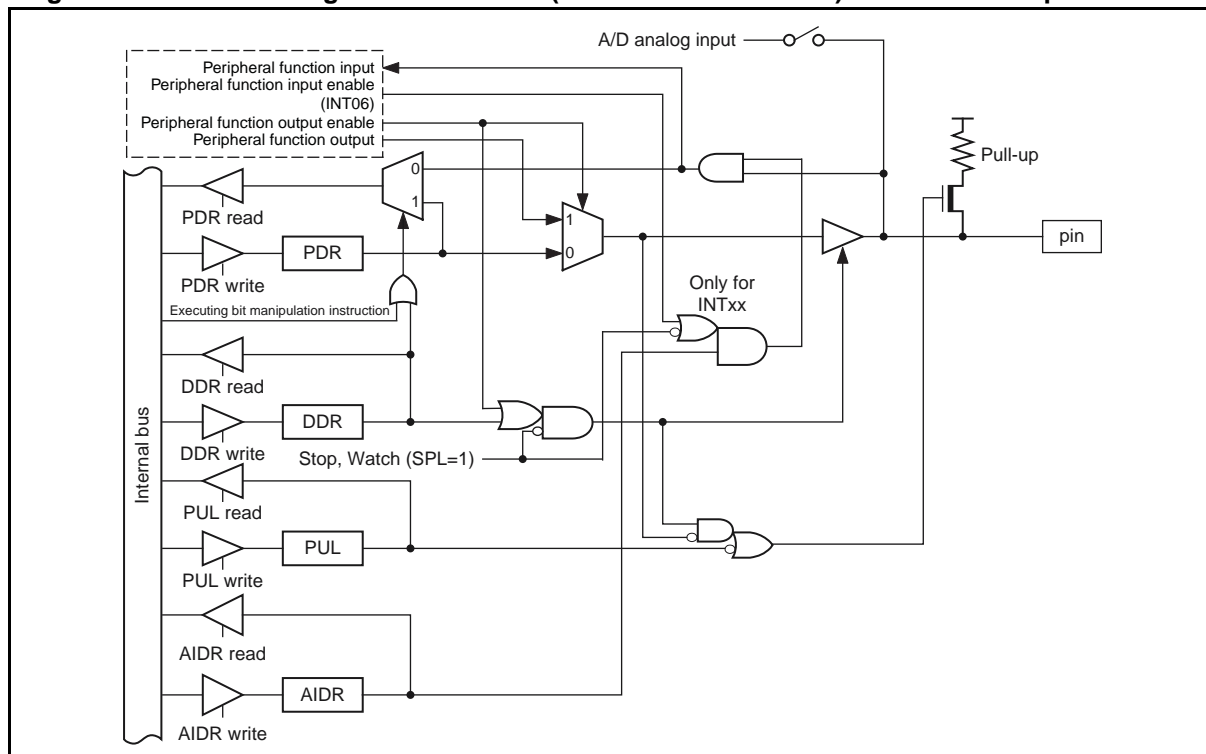**Figure 14.4-4 Block Diagram of Pin TO01 (P06/INT06/AN06/TO01) of 8/16-bit Composite Timer**

MB95330H Series

Figure 14.4-5 Block Diagram of Pins TO10 and TO11 (P62/TO10/PPG00/OPT0 and P63/TO11/ PPG01/OPT1) of 8/16-bit Composite Timer
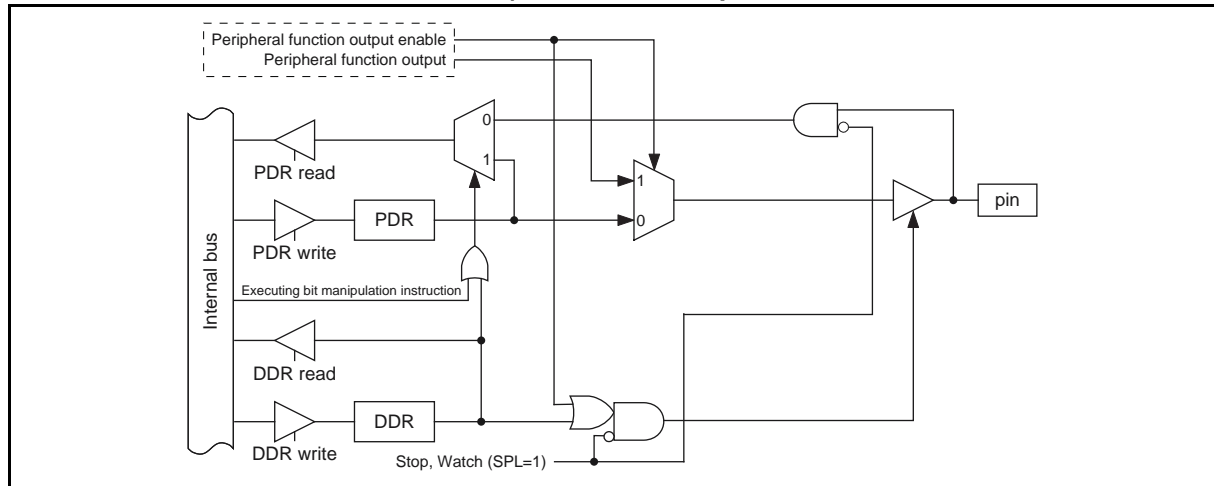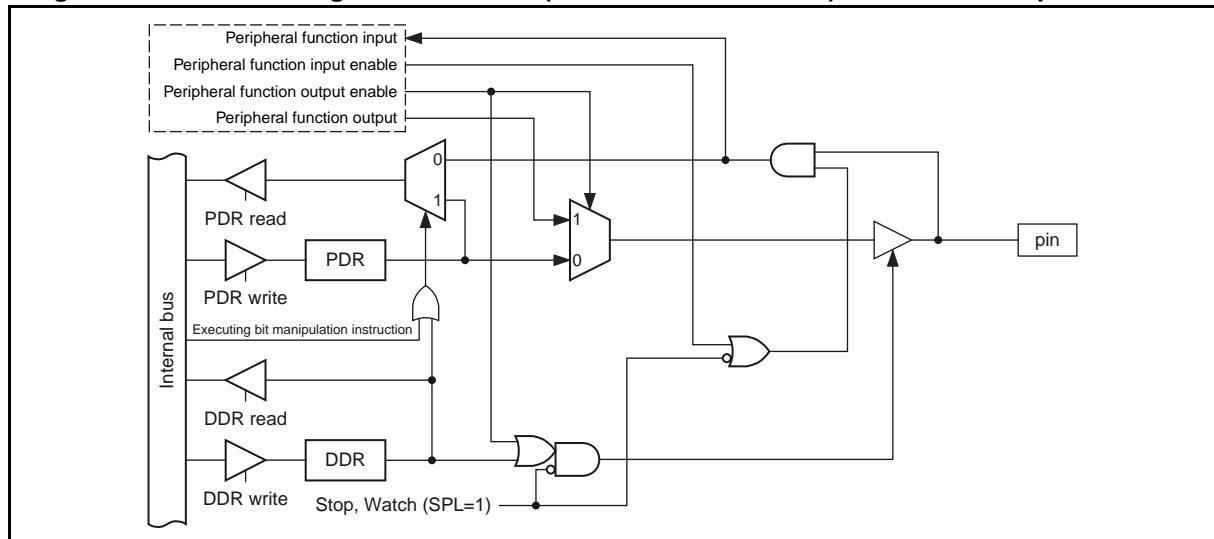

Figure 14.4-6 Block Diagram of Pin EC1 (P64/EC1/PPG10/OPT2) of 8/16-bit Composite Timer

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 (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.)

MB95330H Series

■ Registers of 8/16-bit Composite Timer 1

Figure 14.5-2 Registers of 8/16-bit Composite Timer 1

8/16-bit composite timer 10/11 status control register 0 (T10CR0/T11CR0)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11CR0	0F97$_H$	IFE	C2	C1	C0	F3	F2	F1	F0	00000000$_B$
T10CR0	0F98$_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit composite timer 10/11 status control register 1 (T10CR1/T11CR1)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11CR1	0038$_H$	STA	HO	IE	IR	BF	IF	SO	OE	00000000$_B$
T10CR1	0039$_H$	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

8/16-bit composite timer 10/11 data register (T10DR/T11DR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11DR	0F99$_H$	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000$_B$
T10DR	0F9A$_H$	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	

8/16-bit composite timer 10/11 timer mode control register (TMCR1)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	0F9B$_H$	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)

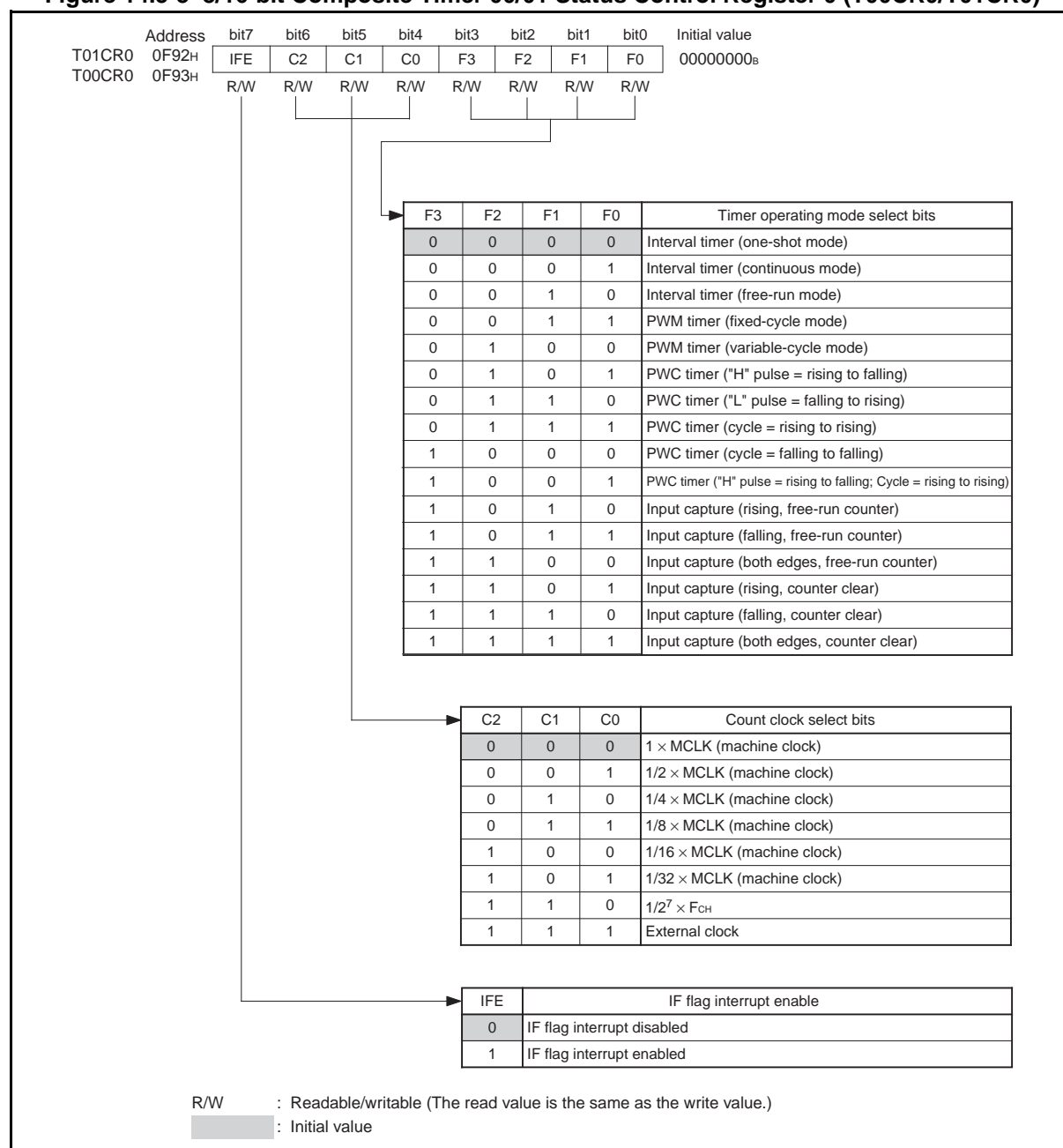**The 8/16-bit composite timer 00/01 status control register 0 (T00CR0/T01CR0) selects the timer operation mode, selects the count clock, and enables or disables IF flag interrupts. The T00CR0 and T01CR0 registers correspond to timers 00 and 01 respectively.**

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

Figure 14.5-3 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	0F92H	IFE	C2	C1	C0	F3	F2	F1	F0	00000000B
T00CR0	0F93H	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	$1 \times$ MCLK (machine clock)
0	0	1	$1/2 \times$ MCLK (machine clock)
0	1	0	$1/4 \times$ MCLK (machine clock)
0	1	1	$1/8 \times$ MCLK (machine clock)
1	0	0	$1/16 \times$ MCLK (machine clock)
1	0	1	$1/32 \times$ MCLK (machine clock)
1	1	0	$1/2^7 \times F_{CH}$
1	1	1	External clock

IFE	IF flag interrupt enable
0	IF flag interrupt disabled
1	IF flag interrupt enabled

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

MB95330H Series

Table 14.5-1 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control Register 0 (T00CR0/T01CR0) (1 / 2)

Bit name		Function
bit7	IFE: IF flag interrupt enable	This bit enables or disables IF flag interrupts. **Writing "0"**: disables IF flag interrupts. **Writing "1"**: an IF flag interrupt request is output when both the IE bit (T00CR1/T01CR1:IE) and the IF flag (T00CR1/T01CR1:IF) are set to "1".
bit6 to bit4	C2, C1, C0: Count clock select bits	These bits select the count clock. • The count clock is generated by the prescaler. See Section 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".

C2	C1	C0	Count clock
0	0	0	$1 \times$ MCLK (machine clock)
0	0	1	$1/2 \times$ MCLK (machine clock)
0	1	0	$1/4 \times$ MCLK (machine clock)
0	1	1	$1/8 \times$ MCLK (machine clock)
1	0	0	$1/16 \times$ MCLK (machine clock)
1	0	1	$1/32 \times$ MCLK (machine clock)
1	1	0	$1/2^7 \times F_{CH}$
1	1	1	External clock

Table 14.5-1 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control Register 0 (T00CR0/T01CR0) (2 / 2)

Bit name		Function
bit3 to bit0	F3, F2, F1, F0: Timer operating mode select bits	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).

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)

MB95330H Series

14.5.2 8/16-bit Composite Timer 10/11 Status Control Register 0 (T10CR0/T11CR0)

The 8/16-bit composite timer 10/11 status control register 0 (T10CR0/T11CR0) selects the timer operation mode, selects the count clock, and enables or disables IF flag interrupts. The T10CR0 and T11CR0 registers correspond to timers 10 and 11 respectively.

■ **8/16-bit Composite Timer 10/11 Status Control Register 0 (T10CR0/T11CR0)**

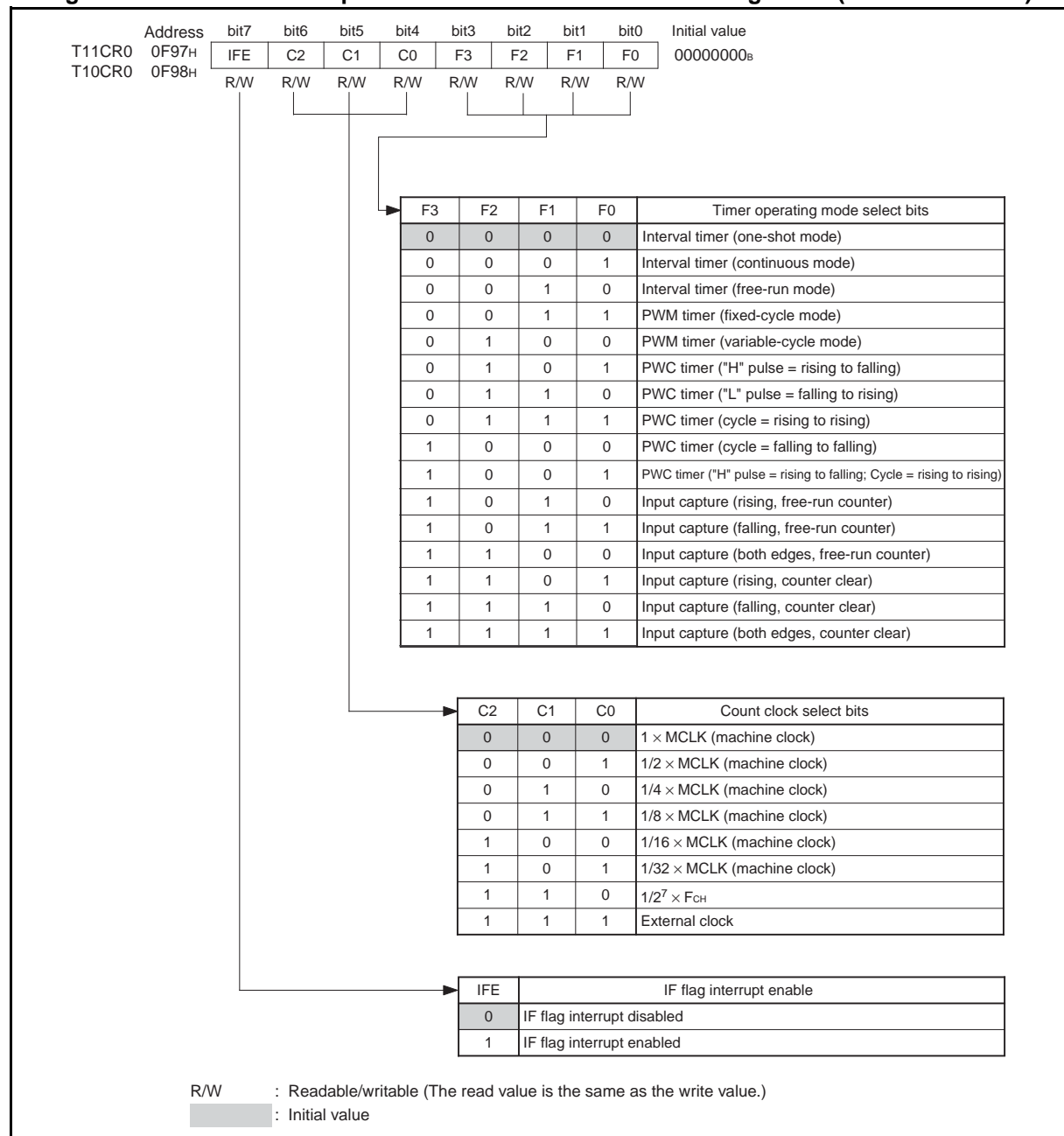**Figure 14.5-4 8/16-bit Composite Timer 10/11 Status Control Register 0 (T10CR0/T11CR0)**

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11CR0	0F97H	IFE	C2	C1	C0	F3	F2	F1	F0	00000000B
T10CR0	0F98H									
		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	$1 \times$ MCLK (machine clock)
0	0	1	$1/2 \times$ MCLK (machine clock)
0	1	0	$1/4 \times$ MCLK (machine clock)
0	1	1	$1/8 \times$ MCLK (machine clock)
1	0	0	$1/16 \times$ MCLK (machine clock)
1	0	1	$1/32 \times$ MCLK (machine clock)
1	1	0	$1/2^7 \times F_{CH}$
1	1	1	External clock

IFE	IF flag interrupt enable
0	IF flag interrupt disabled
1	IF flag interrupt enabled

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

▨ : Initial value

Table 14.5-2 Functions of Bits in 8/16-bit Composite Timer 10/11 Status Control Register 0 (T10CR0/T11CR0) (1 / 2)

Bit name		Function
bit7	IFE: IF flag interrupt enable	This bit enables or disables IF flag interrupts. **Writing "0"**: disables IF flag interrupts. **Writing "1"**: an IF flag interrupt request is output when both the IE bit (T10CR1/T11CR1:IE) and the IF flag (T10CR1/T11CR1:IF) are set to "1".
bit6 to bit4	C2, C1, C0: Count clock select bits	These bits select the count clock. • The count clock is generated by the prescaler. See Section 6.12 "Operation of Prescaler". • Write access to these bits is nullified in timer operation (T10CR1/T11CR1:STA = 1). • The clock selection of T11CR0 (timer 11) 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". *(see table below)*

C2	C1	C0	Count clock
0	0	0	$1 \times$ MCLK (machine clock)
0	0	1	$1/2 \times$ MCLK (machine clock)
0	1	0	$1/4 \times$ MCLK (machine clock)
0	1	1	$1/8 \times$ MCLK (machine clock)
1	0	0	$1/16 \times$ MCLK (machine clock)
1	0	1	$1/32 \times$ MCLK (machine clock)
1	1	0	$1/2^7 \times F_{CH}$
1	1	1	External clock

MB95330H Series

Table 14.5-2 Functions of Bits in 8/16-bit Composite Timer 10/11 Status Control Register 0 (T10CR0/T11CR0) (2 / 2)

Bit name		Function
bit3 to bit0	F3, F2, F1, F0: Timer operating mode select bits	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 T10CR0 (timer 10) register or T11CR0 (timer 11) register. If one of the timers starts operating (T10CR1/T11CR1: 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 (TMCR1:MOD = 1), if the composite timer starts operating using the PWM timer function (variable-cycle mode) (T10CR1/T11CR1:STA = 1), the MOD bit is set to "0" automatically. • Write access to these bits is nullified in timer operation (T10CR1/T11CR1:STA = 1).

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)

14.5.3   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 14.5-5 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	0036H	STA	HO	IE	IR	BF	IF	SO	OE	00000000B
T00CR1	0037H									
		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 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 14.5-3 Functions of Bits in 8/16-bit Composite Timer 00/01 Status Control Register 1 (T00CR1/T01CR1) (1 / 2)

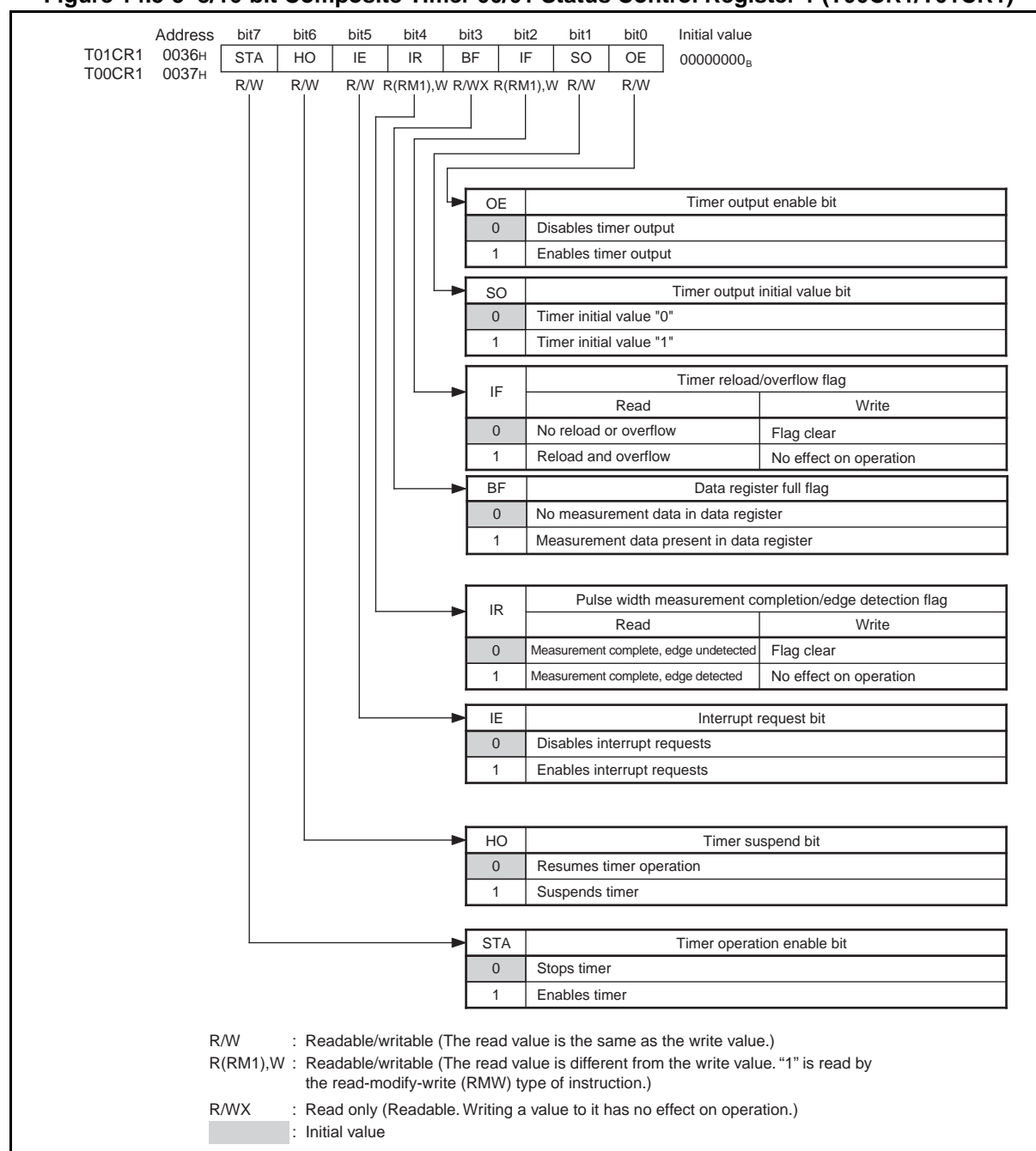| Bit name		Function
bit7	STA: Timer operation enable bit	This bit enables or stops the timer operation. **Writing "0"**: stops the timer operation and sets the count value to "00$_H$". • With the PWM timer function (variable-cycle mode) in use (T00CR0/T01CR0: F3, F2, F1, F0 = 0100$_B$), the STA bit 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 14.5-3 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 (T00DR/T01DR) immediately after pulse width measurement is complete. • In 8-bit operation, this bit is set to "0" when the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is read. • The 8/16-bit composite timer 00/01 data register (T00DR/T01DR) holds data if this bit is set to "1". With this bit being "1", even when the next edge is detected, the count value is not transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), and the next measurement result is thus lost. Nonetheless, there is an exception. With the F3 bit to F0 bit in the T00CR0/T01CR0 register having been set to "1001_B", even though the BF bit is set to "1", the "H" pulse measurement result is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), while the cycle measurement result is not transferred to the 8/16-bit composite timer 00/01 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse. • The BF bit in the T00CR1 (timer 00) register is set to "0" when the T01DR (timer 01) register is read during 16-bit operation. • The BF bit in T01CR1 (timer 01) register is set to "0" during 16-bit operation. • This bit is "0" when any timer function other than the PWC timer function is selected. • Writing a value to this bit has no effect on operation.
bit2	IF: Timer reload/overflow flag	This bit is used to detect the count value match and the counter overflow. • With the interval timer function (one-shot or continuous) or the PWM timer function (variable-cycle mode) in use, this bit is set to "1" if the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) value matches the count value. • With the PWC timer function of the input capture function in use, this bit is set to "1" if a counter overflow occurs. • If this bit is read by a read-modify-write (RMW) instruction, it always returns "1". • Writing "0" to this bit sets it to "0". • Writing "1" to this bit has no effect on operation. • The bit becomes "0" if the PWM function (variable-cycle mode) is selected. • The IF bit in the T01CR1 (timer 01) register is "0" in 16-bit operation.
bit1	SO: Timer output initial value bit	The timer output (TMCR0:TO1/TO0) initial value is set by writing a value to this bit. The value in this bit is reflected in the timer output when the timer operation enable bit (T00CR1/T01CR1:STA) changes from "0" to "1". • In 16-bit operation (TMCR0:MOD = 1), use the SO bit in the T00CR1 (timer 00) register to set the timer output initial value. In this case, the value of the SO bit in the other one has no effect on operation. • During timer operation (T00CR1:STA = 1 or T01CR1:STA = 1), the write access to this bit is invalid. However, in 16-bit operation, although a value can be written to the SO bit in the T01CR1 (timer 01) register even during timer operation, the value written has no direct effect on the timer output. • When the PWM timer function (fixed cycle mode or variable cycle mode) or the input capture function is in use, the value of this bit has no effect on operation.
bit0	OE: Timer output enable bit	This bit enables or disables timer output. **Writing "0"**: no timer output is supplied to the external pin. In this case, the external pin serves as a general-purpose port. **Writing "1"**: the time output (TMCR0:TO1/TO0) is supplied to the external pin.

MB95330H Series

14.5.4   8/16-bit Composite Timer 10/11 Status Control Register 1 (T10CR1/T11CR1)

The 8/16-bit composite timer 10/11 status control register 1 (T10CR1/T11CR1) controls the interrupt flag, timer output, and timer operations. T10CR1 and T11CR1 registers correspond to timers 10 and 11 respectively.

■ 8/16-bit Composite Timer 10/11 Status Control Register 1 (T10CR1/T11CR1)

Figure 14.5-6 8/16-bit Composite Timer 10/11 Status Control Register 1 (T10CR1/T11CR1)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11CR1	0038H	STA	HO	IE	IR	BF	IF	SO	OE	00000000B
T10CR1	0039H	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 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.)
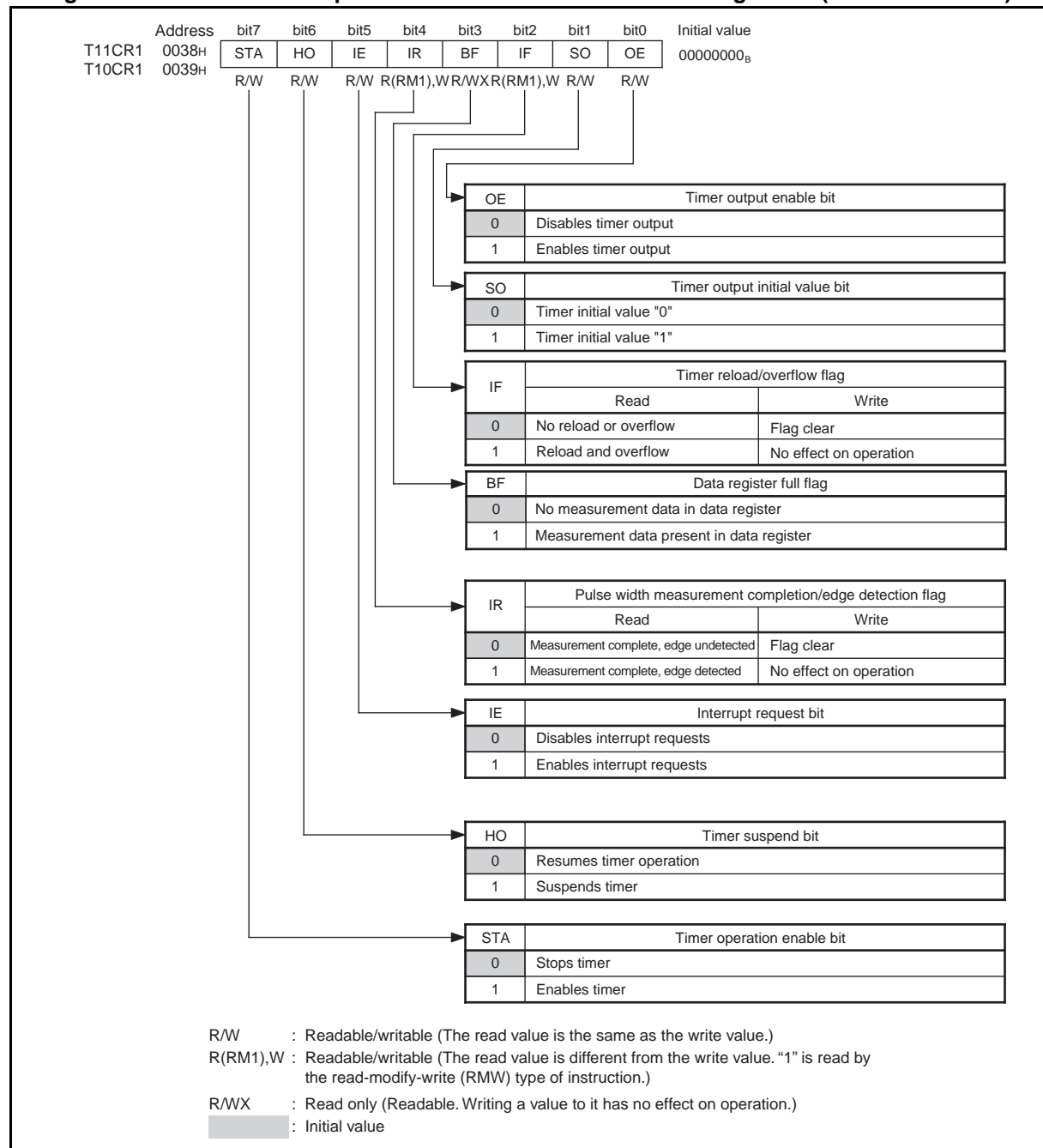░░░   : Initial value

Table 14.5-4 Functions of Bits in 8/16-bit Composite Timer 10/11 Status Control Register 1 (T10CR1/T11CR1) (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 (T10CR0/T11CR0: F3, F2, F1, F0 = 0100$_B$), the STA bit in either the T10CR1 (timer 10) or the T11CR1 (timer 11) 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 (TMCR1:MOD = 1), use the STA bit in the T10CR1 (timer 10) 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 (T10CR0/T11CR0:C2, C1, C0), timer operation select bits (T10CR0/T11CR0:F3, F2, F1, F0), timer output initial value bit (T10CR1/T11CR1:SO), 16-bit mode enable bit (TMCR1:MOD), and filter function select bits (TMCR1: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 (T10CR1/T11CR1:STA = 1), writing "0" to the bit resumes the timer operation. • With the PWM timer function (variable-cycle mode) in used (T10CR0/T11CR0: F3, F2, F1, F0=0100$_B$), the HO bit in either T10CR1 (timer 10) or T11CR1 (timer 11) 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 (TMCR1:MOD = 1), use the HO bit in the T10CR1 (timer 10) 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 (T10CR1/T11CR1:IR) or timer reload/overflow flag (T10CR1/T11CR1:IF) is "1". However, an interrupt request from the timer reload/overflow flag (T10CR1/T11CR1:IF) is not output unless the IF flag interrupt enable (T10CR0/T11CR0: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 T11CR1 (timer 11) 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.

MB95330H Series

**Table 14.5-4 Functions of Bits in 8/16-bit Composite Timer 10/11 Status Control Register 1
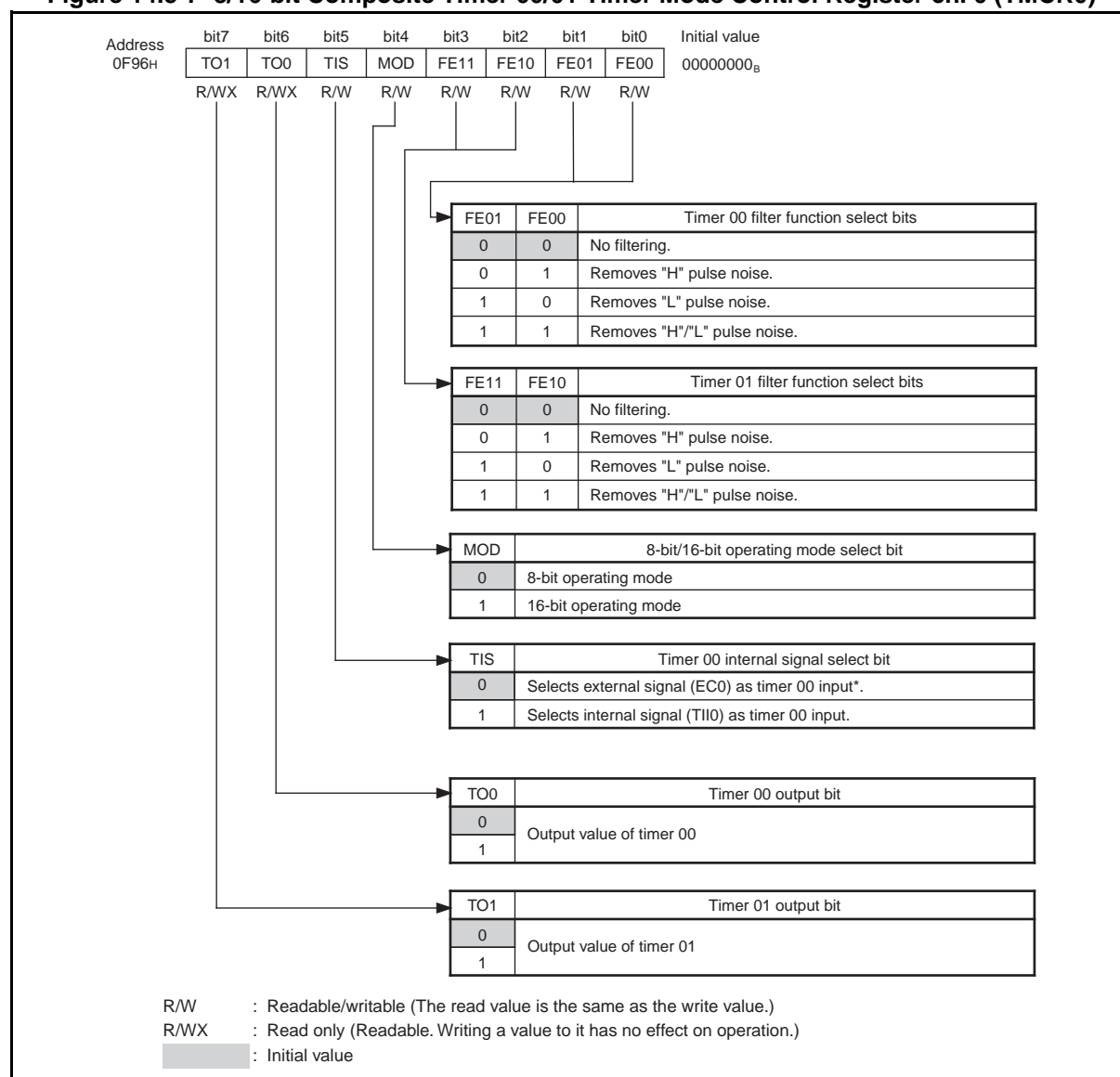(T10CR1/T11CR1) (2 / 2)**

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 10/11 data register (T10DR/T11DR) immediately after pulse width measurement is complete. • In 8-bit operation, this bit is set to "0" when the 8/16-bit composite timer 10/11 data register (T10DR/T11DR) is read. • The 8/16-bit composite timer 10/11 data register (T10DR/T11DR) 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 10/11 data register (T10DR/T11DR), and the next measurement result is thus lost. Nonetheless, there is an exception. With the F3 bit to F0 bit in the T10CR0/T11CR0 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 10/11 data register (T10DR/T11DR), while the cycle measurement result is not transferred to the 8/16-bit composite timer 10/11 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse. • The BF bit in the T10CR1 (timer 10) register is set to "0" when the T11DR (timer 11) register is read during 16-bit operation. • The BF bit in T11CR1 (timer 11) 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 10/11 data register (T10DR/T11DR) 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 T11CR1 (timer 11) register is "0" in 16-bit operation.
bit1	SO: Timer output initial value bit	The timer output (TMCR1: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 (T10CR1/T11CR1:STA) changes from "0" to "1". • In 16-bit operation (TMCR1:MOD = 1), use the SO bit in the T10CR1 (timer 10) 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 (T10CR1:STA = 1 or T11CR1: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 T11CR1 (timer 11) 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 (TMCR1:TO1/TO0) is supplied to the external pin.

14.5.5   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 14.5-7 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	00000000B
	R/WX	R/WX	R/W	R/W	R/W	R/W	R/W	R/W	

FE01	FE00	Timer 00 filter function select bits
0	0	No filtering.
0	1	Removes "H" pulse noise.
1	0	Removes "L" pulse noise.
1	1	Removes "H"/"L" pulse noise.

FE11	FE10	Timer 01 filter function select bits
0	0	No filtering.
0	1	Removes "H" pulse noise.
1	0	Removes "L" pulse noise.
1	1	Removes "H"/"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	Selects internal signal (TII0) as timer 00 input.

TO0	Timer 00 output bit
0	Output value of timer 00
1	

TO1	Timer 01 output bit
0	Output value of timer 01
1	

R/W   : Readable/writable (The read value is the same as the write value.)
R/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 P04 by setting the SYSC register. For details, see CHAPTER 31 "SYSTEM CONFIGURATION CONTROLLER".

MB95330H Series

Table 14.5-5 Functions of Bits in 8/16-bit Composite Timer 00/01 Timer Mode Control Register ch. 0 (TMCR0) (1 / 2)

	Bit name	Function
bit7	TO1: Timer 01 output bit	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.
bit6	TO0: Timer 00 output bit	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.
bit5	TIS: Timer 00 internal signal select bit	This bit selects the signal input to timer 00 when the PWC timer function or input capture function is selected. **Writing "0"**: selects the external signal (EC0) as the signal input for timer 00. **Writing "1"**: selects the internal signal (TII0) as the signal input for timer 00. The EC0 input can be assigned to P12 or P04 by setting the SYSC register. For details, see Section 31.2 "System Configuration Register (SYSC)" in CHAPTER 31 "SYSTEM CONFIGURATION CONTROLLER".
bit4	MOD: 8-bit/16-bit operating mode select bit	This bit selects 8-bit or 16-bit operating mode. **Writing "0"**: allows timers 00 and 01 to operate as separate 8-bit timers. **Writing "1"**: allows timers 00 and 01 to operate as a 16-bit timer. • While this bit is "1", if the timer starts operating (T00CR1/T01CR1:STA = 1) with the PWM timer function (variable-cycle mode), this bit is automatically set to "0". • During timer operation (T00CR1:STA = 1 or T01CR1:STA = 1), the write access to this bit is invalid.
bit3, bit2	FE11, FE10: Timer 01 filter function select bits	These bits select the filter function for the external signal (EC0) to timer 01 when the PWC timer function or the input capture function is selected. | FE11	FE10	Timer 01 filter
0	0	No filtering
0	1	Removing "H" pulse noise
1	0	Removing "L" pulse noise
1	1	Removing "H"/"L" pulse noise | • 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.).

Table 14.5-5 Functions of Bits in 8/16-bit Composite Timer 00/01 Timer Mode Control Register ch. 0 (TMCR0) (2 / 2)

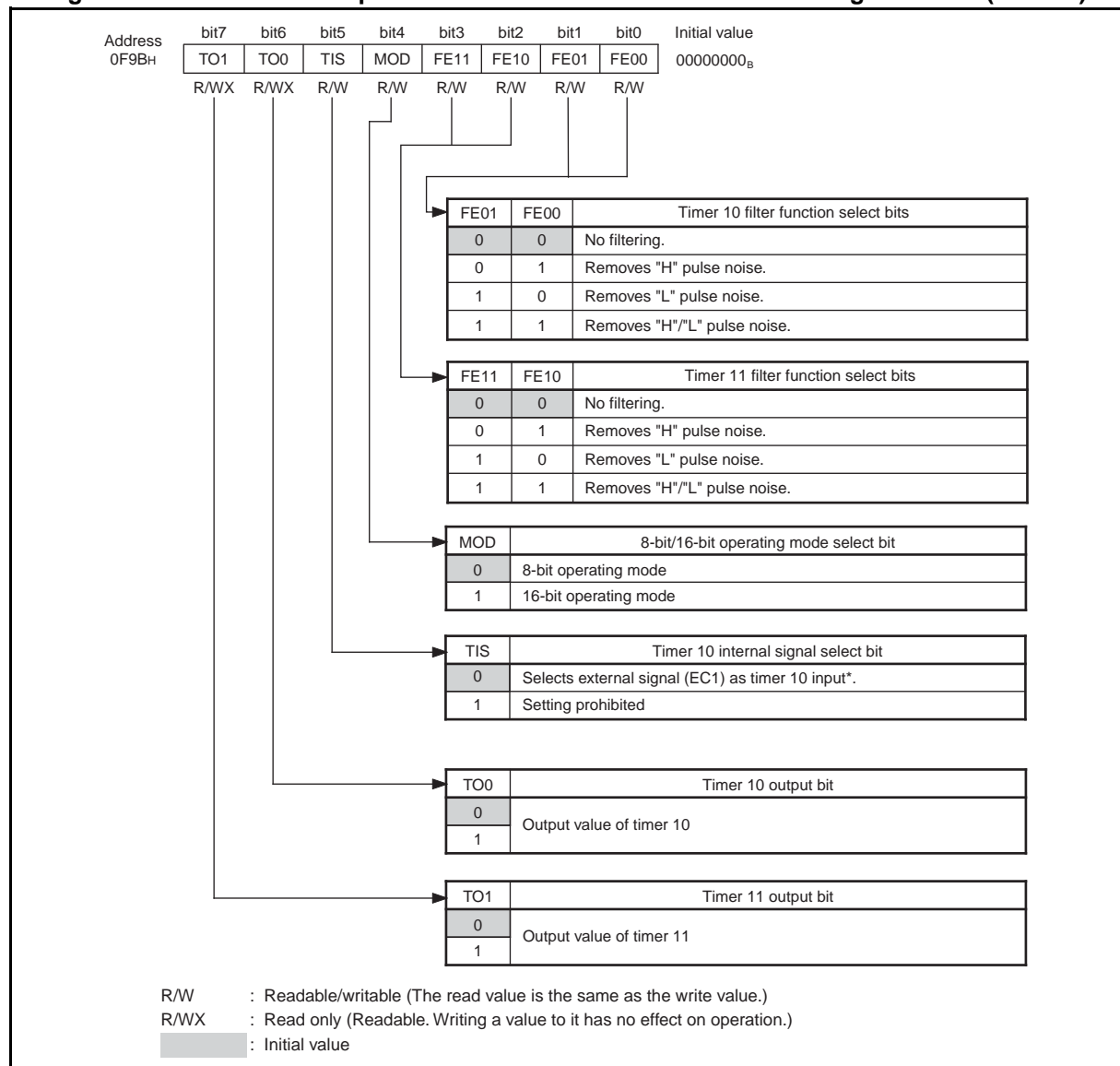| Bit name		Function
bit1, bit0	FE01, FE00: Timer 00 filter function select bits	These bits select the filter function for the external signal (EC0) to timer 00 when the PWC timer function or the input capture function is selected. <table><tr><td>FE01</td><td>FE00</td><td>Timer 00 filter</td></tr><tr><td>0</td><td>0</td><td>No filtering</td></tr><tr><td>0</td><td>1</td><td>Removing "H" pulse noise</td></tr><tr><td>1</td><td>0</td><td>Removing "L" pulse noise</td></tr><tr><td>1</td><td>1</td><td>Removing "H"/"L" pulse noise</td></tr></table> • During timer operation (T00CR1:STA = 1), the write access to these bits is invalid. • The settings of these bits have no effect on operation when the interval timer function or the PWM timer function is selected (the filter function does not operate.).

MB95330H Series

14.5.6   8/16-bit Composite Timer 10/11 Timer Mode Control Register ch. 1 (TMCR1)

The 8/16-bit composite timer 10/11 timer mode control register ch. 1 (TMCR1) selects the filter function, 8-bit or 16-bit operating mode, and signal input to timer 10 and indicates the timer output value. This register serves both timer 10 and timer 11.

■ **8/16-bit Composite Timer 10/11 Timer Mode Control Register ch. 1 (TMCR1)**

Figure 14.5-8 8/16-bit Composite Timer 10/11 Timer Mode Control Register ch. 1 (TMCR1)

Address 0F9B_H

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	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 10 filter function select bits
0	0	No filtering.
0	1	Removes "H" pulse noise.
1	0	Removes "L" pulse noise.
1	1	Removes "H"/"L" pulse noise.

FE11	FE10	Timer 11 filter function select bits
0	0	No filtering.
0	1	Removes "H" pulse noise.
1	0	Removes "L" pulse noise.
1	1	Removes "H"/"L" pulse noise.

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

TIS	Timer 10 internal signal select bit
0	Selects external signal (EC1) as timer 10 input*.
1	Setting prohibited

TO0	Timer 10 output bit
0	Output value of timer 10
1	

TO1	Timer 11 output bit
0	Output value of timer 11
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 EC1 input is assigned to P64.

Table 14.5-6 Functions of Bits in 8/16-bit Composite Timer 10/11 Timer Mode Control Register ch. 1 (TMCR1) (1 / 2)

Bit name	Function
bit7	TO1: Timer 11 output bit	This bit indicates the output value of timer 11. When the timer starts operation (T10CR1/T11CR1: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 (T10CR1/T11CR1:STA = 0), this bit holds the last value. • With the PWM timer function (variable-cycle mode) having been selected, if the timer stops operating (T10CR1/T11CR1:STA = 0), this bit holds the last value. • When the timer operating mode select bits (T10CR0/T11CR0: 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.
bit6	TO0: Timer 10 output bit	This bit indicates the output value of timer 10. When the timer starts operation (T10CR1/T11CR1: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 (T10CR1/T11CR1:STA = 0), this bit holds the last value. • With the PWM timer function (variable-cycle mode) having been selected, if the timer stops operating (T10CR1/T11CR1:STA = 0), this bit holds the last value. • When the timer operating mode select bits (T10CR0/T11CR0: 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.
bit5	TIS: Timer 10 internal signal select bit	This bit selects the signal input to timer 10 when the PWC timer function or input capture function is selected. **Writing "0"**: selects the external signal (EC1) as the signal input for timer 10. **Writing "1"**: Writing "1" to TIS is prohibited because it selects the internal signal (TII0) as signal input for timer 10 but the TII0 pin for ch. 1 is internally fixed at "0". The EC1 input is assigned to P64.
bit4	MOD: 8-bit/16-bit operating mode select bit	This bit selects 8-bit or 16-bit operating mode. **Writing "0"**: allows timers 10 and 11 to operate as separate 8-bit timers. **Writing "1"**: allows timers 10 and 11 to operate as a 16-bit timer. • While this bit is "1", if the timer starts operating (T10CR1/T11CR1:STA = 1) with the PWM timer function (variable-cycle mode), this bit is automatically set to "0". • During timer operation (T10CR1:STA = 1 or T11CR1:STA = 1), the write access to this bit is invalid.
bit3, bit2	FE11, FE10: Timer 11 filter function select bits	These bits select the filter function for the external signal (EC1) to timer 11 when the PWC timer function or the input capture function is selected. | FE11	FE10	Timer 11 filter
---	---	---
0	0	No filtering
0	1	Removing "H" pulse noise
1	0	Removing "L" pulse noise
1	1	Removing "H"/"L" pulse noise	 • During timer operation (T10CR1: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.).

MB95330H Series

Table 14.5-6 Functions of Bits in 8/16-bit Composite Timer 10/11 Timer Mode Control Register ch. 1 (TMCR1) (2 / 2)

Bit name		Function
bit1, bit0	FE01, FE00: Timer 10 filter function select bits	These bits select the filter function for the external signal (EC1) to timer 10 when the PWC timer function or the input capture function is selected. FE01/FE00 table below • During timer operation (T10CR1: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.).

FE01	FE00	Timer 10 filter
0	0	No filtering
0	1	Removing "H" pulse noise
1	0	Removing "L" pulse noise
1	1	Removing "H"/"L" pulse noise

14.5.7   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 14.5-9 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.

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 (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.

MB95330H Series

● 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 (T00DR/T01DR) is used to read PWC measurement results. When PWC measurement is completed, the counter value is transferred to this register and the BF bit is set to "1".

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

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

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

If new data is written to the 8/16-bit composite timer 00/01 data register, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, 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 (T00DR/T01DR) is used to read input capture results. When an edge specified is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register.

If new data is written to the 8/16-bit composite timer 00/01 data register, the stored measurement data is replaced with the new data. Therefore, do not write data to the register. In 16-bit operation, 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 14.5-10 shows the T00DR and T01DR registers read from and written to during 16-bit operation.

Figure 14.5-10 Read and Write Operations of T00DR and T01DR Registers during 16-bit Operation

MB95330H Series
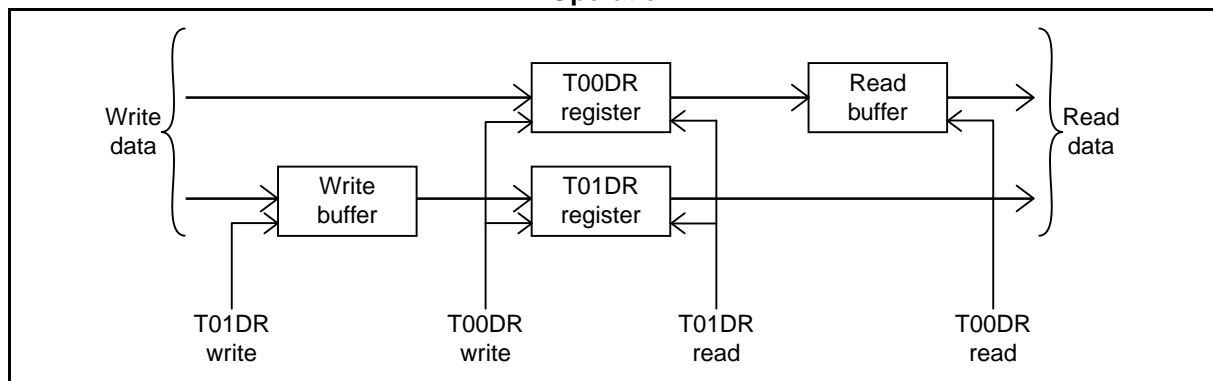
14.5.8   8/16-bit Composite Timer 10/11 Data Register ch. 1 (T10DR/T11DR)

The 8/16-bit composite timer 10/11 data register (T10DR/T11DR) 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 T10DR and T11DR registers correspond to timers 10 and 11 respectively.

■ 8/16-bit Composite Timer 10/11 Data Register ch. 1 (T10DR/T11DR)

Figure 14.5-11 8/16-bit Composite Timer 10/11 Data Register (T10DR/T11DR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T11DR	0F99$_H$	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000$_B$
T10DR	0F9A$_H$	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 10/11 data register (T10DR/T11DR) is used to set the interval time. When the timer starts operating (T10CR1/T11CR1: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.

In 16-bit operation, write the upper timer data to T11DR and lower timer data to T10DR, and write or read T11DR first and then T10DR.

● PWM timer function (fixed-cycle)

The 8/16-bit composite timer 10/11 data register (T10DR/T11DR) is used to set "H" pulse width time. When the timer starts operating (T10CR1/T11CR1: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 T11DR and lower timer data to T10DR, and write or read T11DR first and then T10DR.

● PWM timer function (variable-cycle)

The 8/16-bit composite timer 10 data register (T10DR) and 8/16-bit composite timer 11 data register (T11DR) are used to set "L" pulse width time and cycle respectively. When the timer starts operating (T10CR1/T11CR1: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 T10DR value transferred to the latch matches the timer 10 counter value, the timer output becomes "H" and the counting continues until the T11DR value transferred to the latch matches the timer 11 counter value. When the T11DR value transferred to the latch of the 8-bit comparator matches the timer 11 counter value, the values of the T10DR register and the T11DR 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 T11DR and lower timer data to T10DR, and read T11DR first and then T10DR.

● PWC timer function

The 8/16-bit composite timer 10/11 data register (T10DR/T11DR) 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 10/11 data register is read, the BF bit is set to "0". While the BF bit is "1", no data is transferred to the 8/16-bit composite timer 10/11 data register.

There is an exception. With the F3 bit to F0 bit in the T10CR0/T11CR0 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 10/11 data register, while the cycle measurement result is not transferred to the 8/16-bit composite timer 10/11 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse.

When reading the 8/16-bit composite timer 10/11 data register, ensure that the BF bit is not cleared accidentally.

If new data is written to the 8/16-bit composite timer 10/11 data register, 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 T11DR and lower timer data to T10DR, and read T11DR first and then T10DR.

● Input capture function

The 8/16-bit composite timer 10/11 data register (T10DR/T11DR) 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 10/11 data register.

If new data is written to the 8/16-bit composite timer 10/11 data register, 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 T11DR and lower timer data to T10DR, and read T11DR first and then T10DR.

● Read and write operations

Read and write operations of T10DR and T11DR are performed in the following manner in 16-bit operation or when the PWM timer function (variable-cycle) is selected.

- Read from T11DR: In addition to the read access to T11DR, the value of T10DR is also stored in the internal read buffer at the same time.
- Read from T10DR: The internal read buffer is read.
- Write to T11DR: Data is written to the internal write buffer.
- Write to T10DR: In addition to the write access to T10DR, the value of the internal write buffer is stored in T11DR at the same time.

Figure 14.5-12 shows the T10DR and T11DR registers read from and written to during 16-bit operation.

Figure 14.5-12 Read and Write Operations of T10DR and T11DR Registers during 16-bit Operation
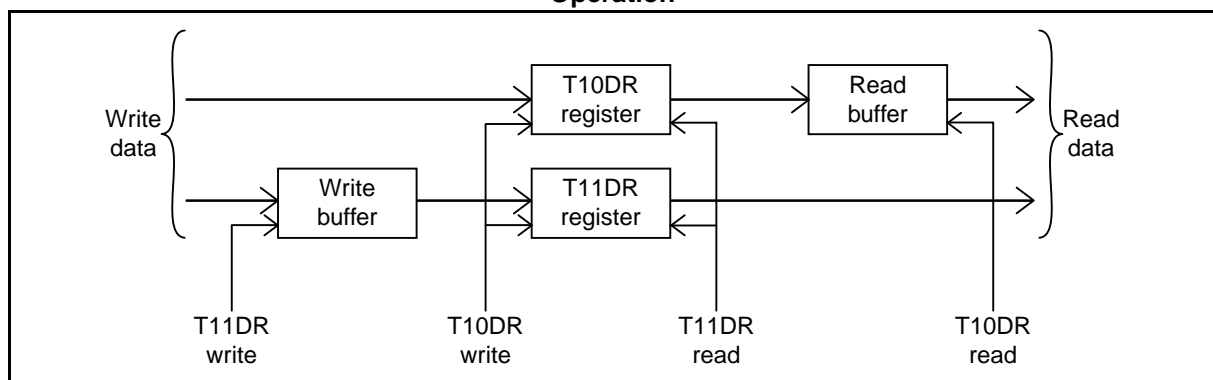
14.6   Interrupts of 8/16-bit Composite Timer

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 10 interrupt**
- **Timer 11 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

MB95330H Series

■ Timer 10 Interrupt

Table 14.6-3 shows the timer 10 interrupt and its sources.

Table 14.6-3 Timer 10 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	T10CR1:IF	T10CR1:IF	T10CR1:IR
Interrupt enable	T10CR1:IE and T10CR0:IFE	T10CR1:IE and T10CR0:IFE	T10CR1:IE

■ Timer 11 Interrupt

Table 14.6-4 shows the timer 11 interrupt and its sources.

Table 14.6-4 Timer 11 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	T11CR1:IF	T11CR1:IF	T11CR1:IR
Interrupt enable	T11CR1:IE and T11CR0:IFE	T11CR1:IE and T11CR0:IFE	T11CR1:IE

■ **Registers and Vector Table Addresses Related to Interrupts of 8/16-bit Composite Timer**

Table 14.6-5 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	$FFEE_H$	$FFEF_H$
8/16-bit composite timer ch. 1 (lower) / Timer 10	IRQ22	ILR5	L22	$FFCE_H$	$FFCF_H$
8/16-bit composite timer ch. 1 (upper) / Timer 11	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.

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 (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 (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator when the counter starts counting.

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)

MB95330H Series
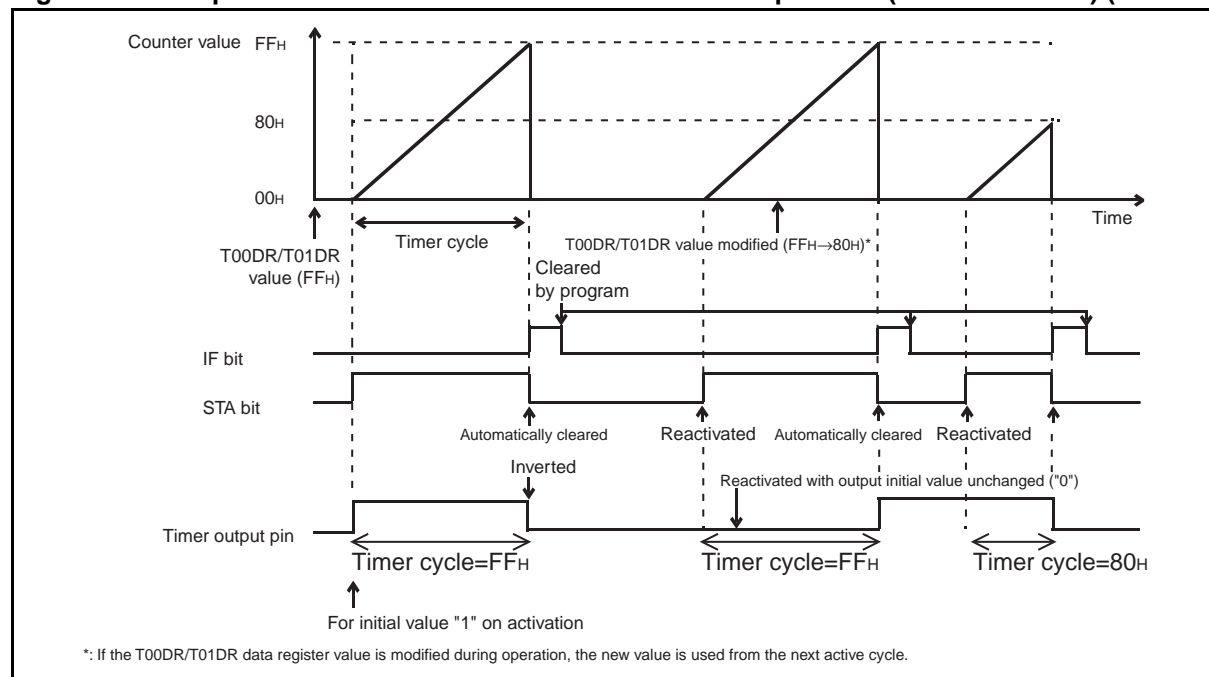
■ Operation of Interval Timer Function (One-shot Mode) (Timer 1)

The register settings shown in Figure 14.7-3 are required to use the interval timer function.

Figure 14.7-3 Settings of Interval Timer Function (One-shot Mode) (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	0	0
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	❍	❍
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T10DR/T11DR	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 (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR), the timer output (TMCR1:TO0/TO1) is inverted, the interrupt flag (T10CR1/T11CR1:IF) is set to "1", the start bit (T10CR1/T11CR1:STA) is set to "0", and the counter stops counting.

The value of the 8/16-bit composite timer 10/11 data register (T10DR/T11DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator when the counter starts counting.

Figure 14.7-4 shows the operation of the interval timer function (timer 1) in 8-bit operation.

Figure 14.7-4 Operation of Interval Timer Function in 8-bit Operation (One-shot Mode) (Timer 1)
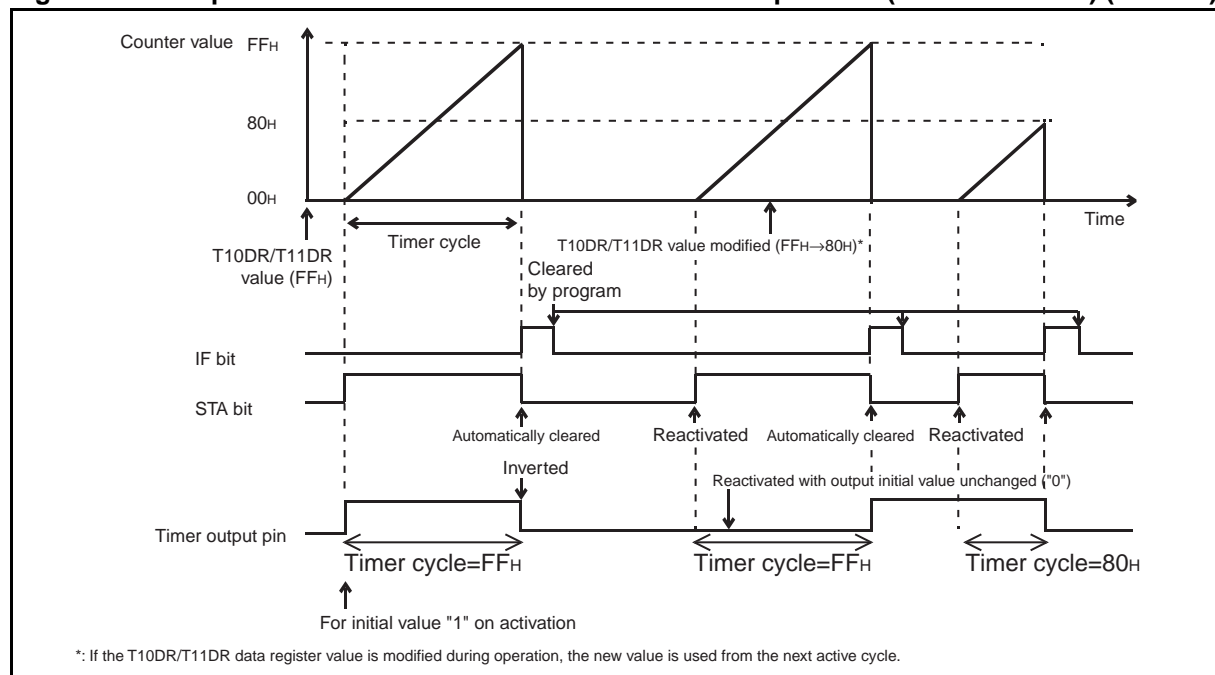
14.8 Operation of Interval Timer Function (Continuous Mode)

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).

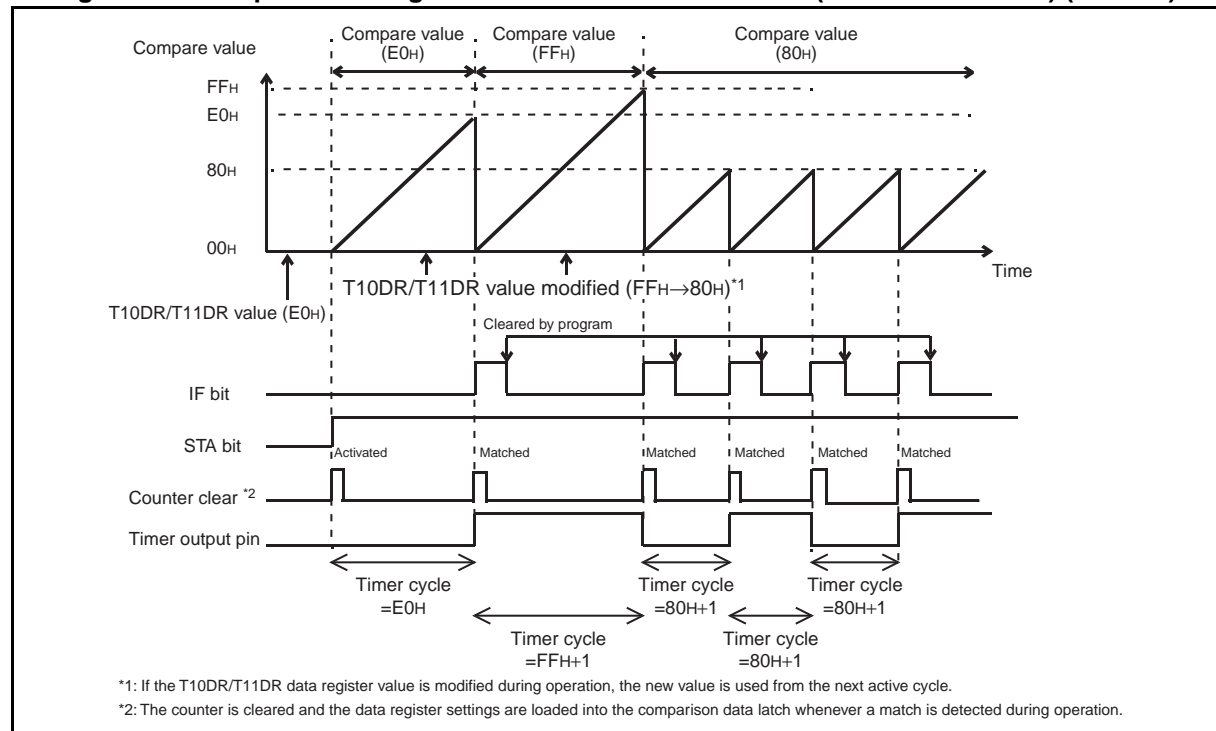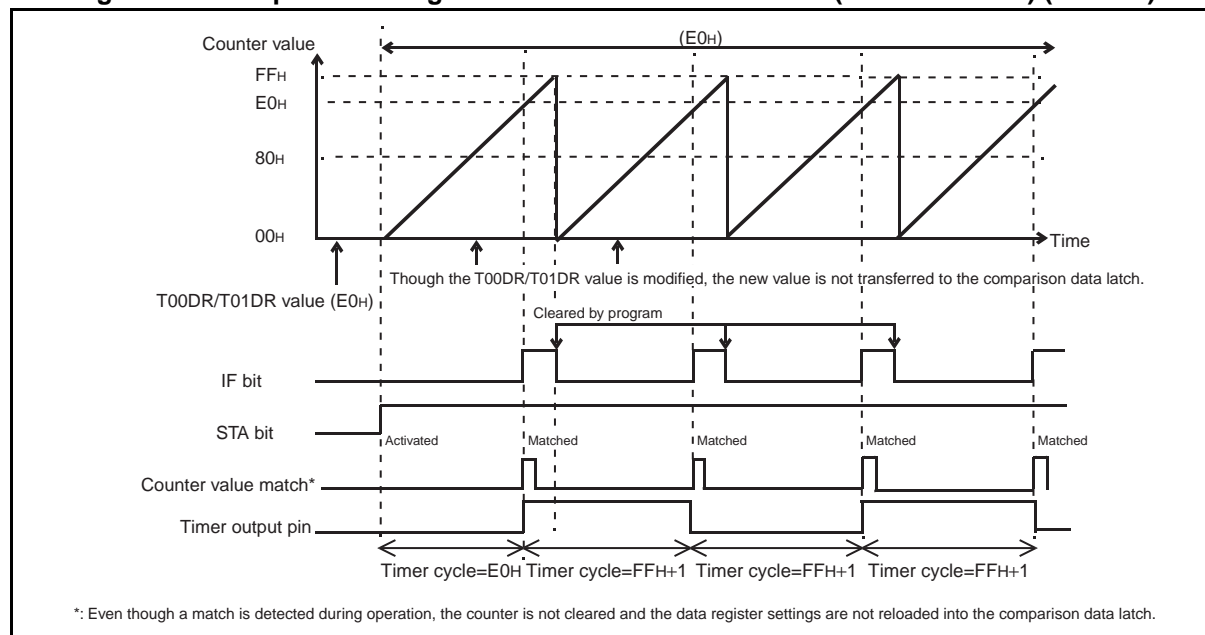**Figure 14.8-1 Settings for Interval Timer Function (Continuous Mode) (Timer 0)**

	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 (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1", and the counter returns to "00_H" and restarts counting. The timer outputs square wave as a result of this continuous operation.

The value of the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected.

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.
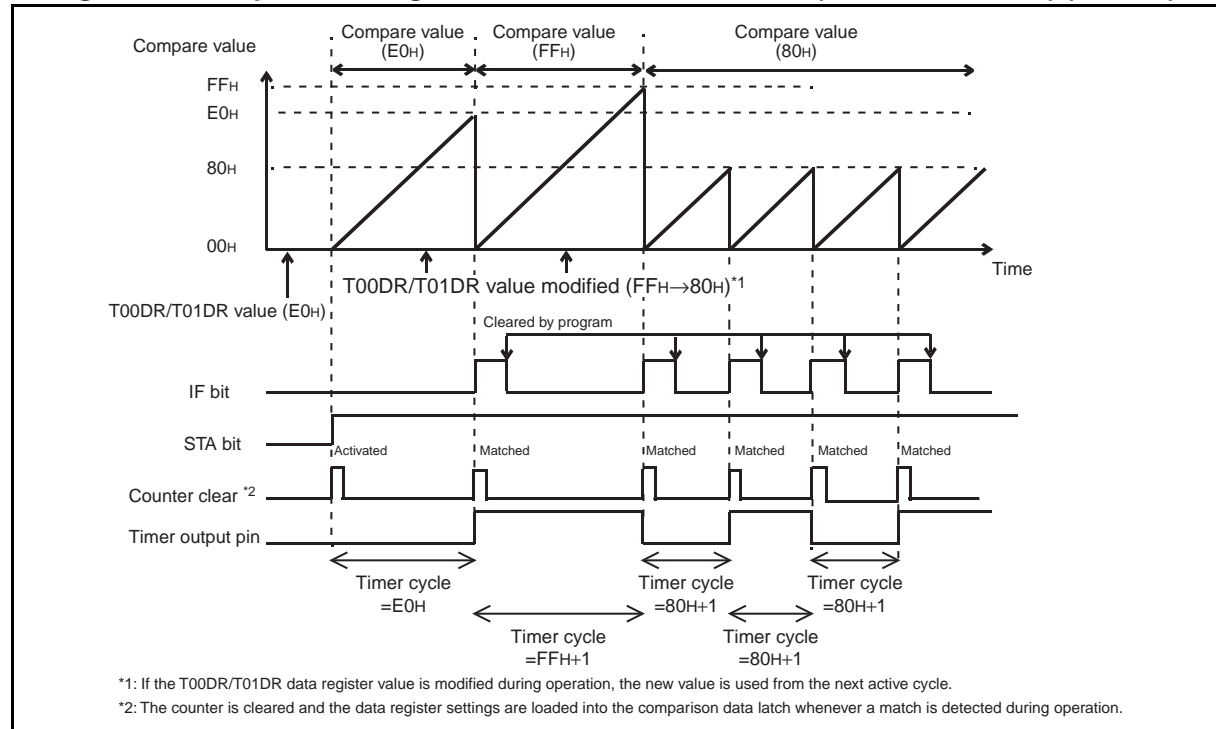
■ Operation of Interval Timer Function (Continuous Mode) (Timer 1)

The register settings shown in Figure 14.8-3 are required to use interval timer function (continuous mode).

**Figure 14.8-3 Settings for Interval Timer Function (Continuous Mode) (Timer 1)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	0	1
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	✕	✕	❍	❍	❍
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	✕	❍	❍	❍	❍	❍
T10DR/T11DR	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 (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR), the timer output bit (TMCR1:TO0/TO1) is inverted, the interrupt flag (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR) 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 operating, the timer output bit (TMCR1:TO0/TO1) holds the last value.

Figure 14.8-4 Operation Diagram of Interval Timer Function (Continuous Mode) (Timer 1)

*1: If the T10DR/T11DR 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 (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted and the interrupt flag (T00CR1/T01CR1:IF) is set to "1". If the counter continues to count with the above settings and then reaches "FF_H", it returns to "00_H" and restarts counting. The timer outputs square wave as a result of this continuous operation.

The value of the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected.

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)

■ **Operation of Interval Timer Function (Free-run Mode) (Timer 1)**

The settings shown in Figure 14.9-3 are required to use the interval timer function (free-run mode).

Figure 14.9-3 Settings for Interval Timer Function (Free-run Mode) (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	0	1	0
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	×	×	○	○	○
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	○	○	×	○	○	○	○	○
T10DR/T11DR	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 (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR), the timer output bit (TMCR1:TO0/TO1) is inverted and the interrupt flag (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR) 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 (TMCR1:TO0/TO1) holds the last value.

Figure 14.9-4 Operation Diagram of Interval Timer Function (Free-run Mode) (Timer 1)

MB95330H Series
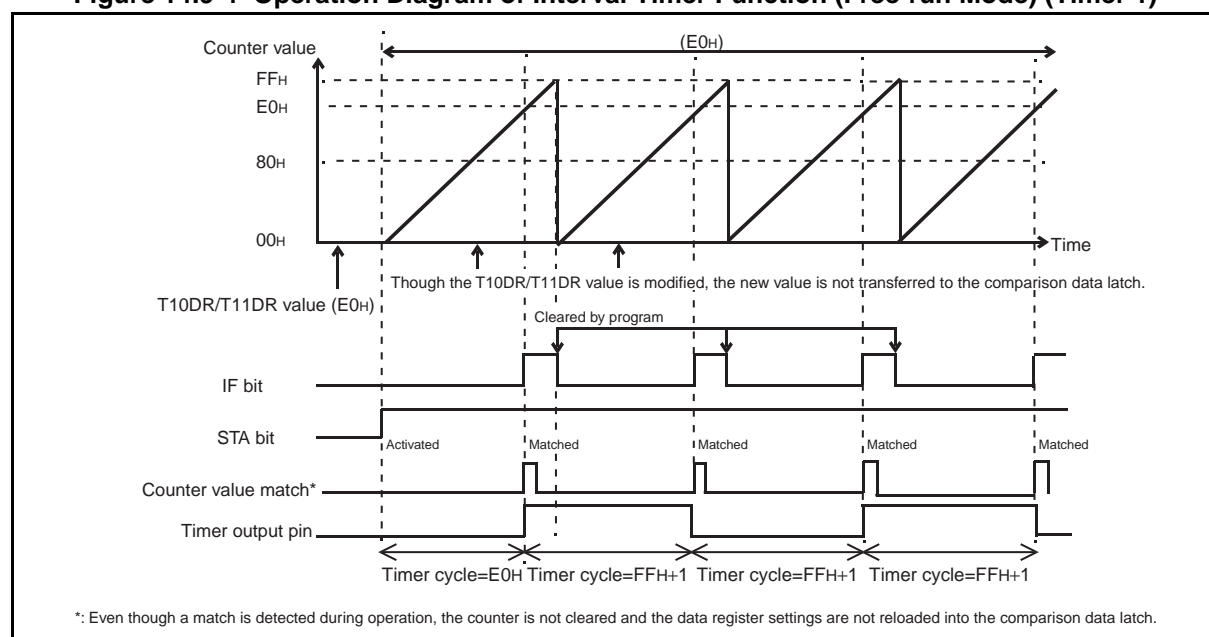
14.10 Operation of PWM Timer Function (Fixed-cycle mode)

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 (T00DR/T01DR).

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

The value of the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

The "H" pulse is one count clock shorter than the setting value in the output waveform immediately after activation of the timer (write "1" to the STA bit), the "H" pulse is one count clock shorter than the value set in the T00DR/T01DR register.

Figure 14.10-2 Operation Diagram of PWM Timer Function (Fixed-cycle Mode) (Timer 0)

T00DR/T01DR register value: "00$_H$" (duty ratio = 0%)

Counter value   00$_H$ → FF$_H$ 00$_H$ →

PWM waveform "H"
              "L"

T00DR/T01DR register value: "80$_H$" (duty ratio = 50%)

Counter value   00$_H$ → 80$_H$ → FF$_H$ 00$_H$ →

PWM waveform "H"
              "L"

T00DR/T01DR register value: "FF$_H$" (duty ratio = 99.6%)

Counter value   00$_H$ → FF$_H$ 00$_H$ →

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
      (T00CR0/T01CR0:STA = 0).

MB95330H Series
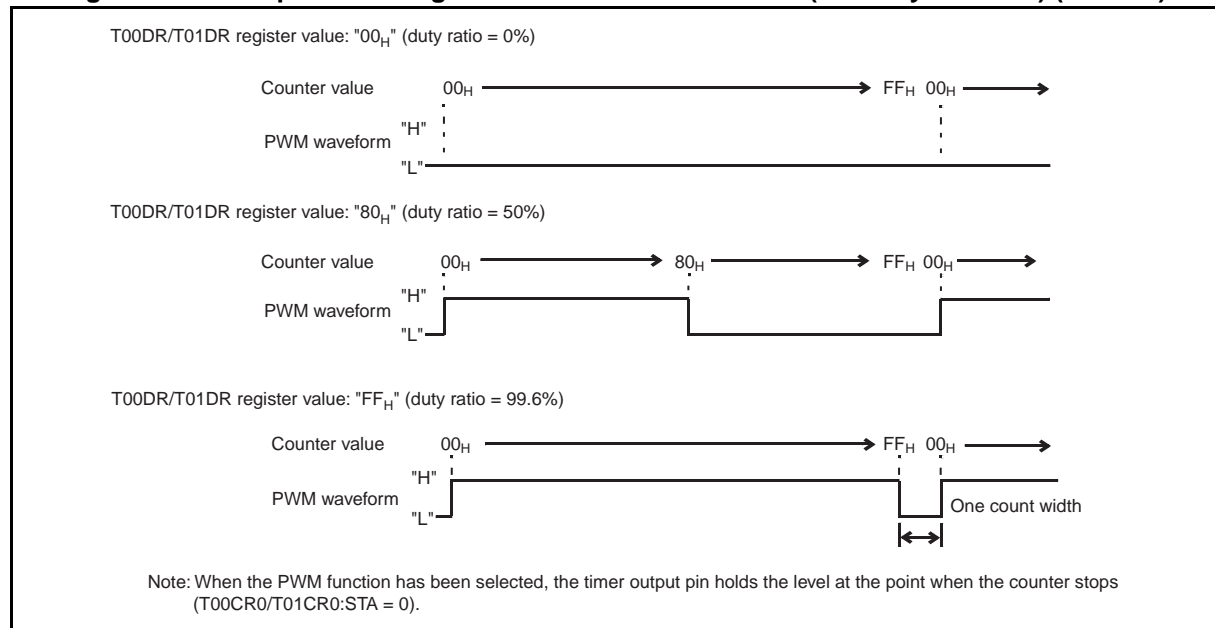
■ Operation of PWM Timer Function (Fixed-cycle Mode) (Timer 1)

The settings shown in Figure 14.10-3 are required to use the PWM timer function (fixed-cycle mode).

Figure 14.10-3 Settings for PWM Timer Function (Fixed-cycle Mode) (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	0	1	1
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	❍	❍	×	×	×	×	×	❍
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	❍	❍	❍	❍	❍
T10DR/T11DR	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 (TO10/TO11). 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 10/11 data register (T10DR/T11DR).

This function has no effect on the interrupt flag (T10CR1/T11CR1:IF). Since each cycle always starts with "H" pulse output, the timer output initial value setting bit (T10CR1/T11CR1:SO) has no effect on operation.

The value of the 8/16-bit composite timer 10/11 data register (T10DR/T11DR) 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 (TMCR1: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 T10DR/T11DR register.

Figure 14.10-4 Operation Diagram of PWM Timer Function (Fixed-cycle Mode) (Timer 1)

T10DR/T11DR register value: "00$_H$" (duty ratio = 0%)

Counter value   00$_H$ ——————————————→ FF$_H$ 00$_H$ ———→

PWM waveform   "H"
                "L"

T10DR/T11DR register value: "80$_H$" (duty ratio = 50%)

Counter value   00$_H$ ————→ 80$_H$ ————→ FF$_H$ 00$_H$ ———→

PWM waveform   "H"
                "L"

T10DR/T11DR register value: "FF$_H$" (duty ratio = 99.6%)

Counter value   00$_H$ ——————————————→ FF$_H$ 00$_H$ ———→

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 (T00DR/T01DR).

The 8/16-bit composite timer 00/01 data register value is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a comparison match with each counter value is detected.

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

The count clock must be selected for both timers 00 and 01. Selecting different count clocks for the two timers is prohibited.

When the timer stops operating, the timer output bit (TMCR0:TO0) holds the last output value.

If the 8/16-bit composite timer 00/01 data register is modified during operation, the data

written will become valid from the cycle immediately after the detection of a synchronous match.

Figure 14.11-2 Operation Diagram of PWM Timer Function (Variable-cycle Mode) (Timer 0)
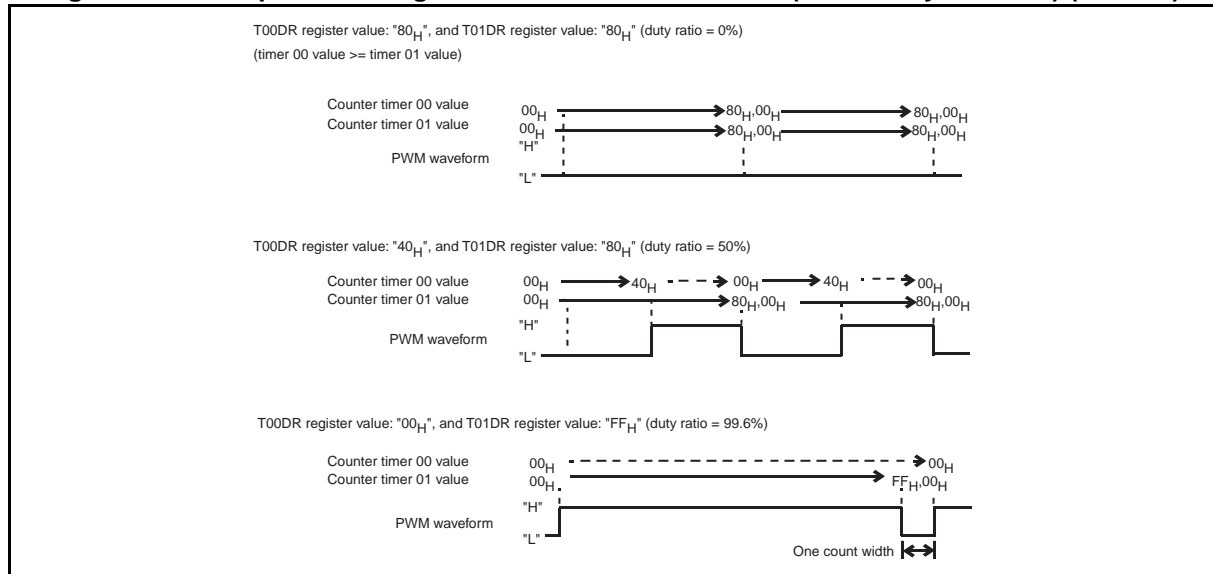
■ **Operation of PWM Timer Function (Variable-cycle Mode) (Timer 1)**

The settings shown in Figure 14.11-3 are required to use the PWM timer function (variable-cycle mode).

Figure 14.11-3 Settings for PWM Timer Function (Variable-cycle Mode) (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	0	1	0	0
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	×	×	❍	×	×
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	×	×	❍	❍	❍	❍
T10DR	Sets "L" pulse width (compare value)							
T11DR	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 10 and 11 are used. PWM signal of any cycle and of any duty is output from the timer output pin (TO10). The cycle is specified by the 8/16-bit composite timer 11 data register (T11DR), and the "L" pulse width is specified by the 8/16-bit composite timer 10 data register (T10DR).

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 T10CR1:STA = 1 or T11CR1:STA = 1) sets the mode bit (TMCR1:MOD) to "0". As the first cycle always begins with "L" pulse output, the timer initial value setting bit (T10CR1/T11CR1:SO) has no effect on operation.

An interrupt flag (T10CR1/T11CR1:IF) is set when the 8-bit counter corresponding to that interrupt flag matches the value in its corresponding 8/16-bit composite timer 10/11 data register (T10DR/T11DR).

The 8/16-bit composite timer 10/11 data register value is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a comparison match with each counter value is detected.

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

The count clock must be selected for both timers 10 and 11. Selecting different count clocks for the two timers is prohibited.

When the timer stops operating, the timer output bit (TMCR1:TO0) holds the last output value.

If the 8/16-bit composite timer 10/11 data register is modified during operation, the data written will become valid from the cycle immediately after the detection of a synchronous match.

Figure 14.11-4 Operation Diagram of PWM Timer Function (Variable-cycle Mode) (Timer 1)

MB95330H Series
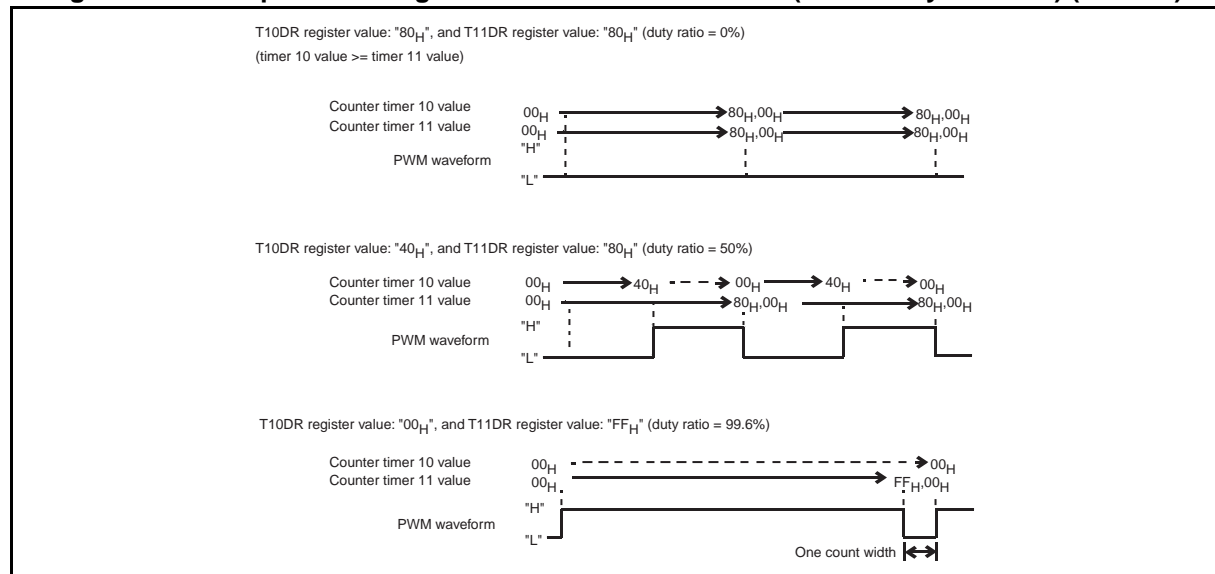
14.12 Operation of PWC Timer Function

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 (T00DR/T01DR), and the interrupt flag (T00CR1/T01CR1:IR) and the buffer full flag (T00CR1/T01CR1:BF) are set to "1". The buffer full flag is set to "0" when the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) is read.

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

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

To measure the time exceeding the range of the counter, software can be used to count the number of counter overflows. When the counter overflows, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of overflows. In addition, the timer output is inverted due to the overflow. The timer output initial value can be set by the timer output initial value bit (T00CR1/T01CR1:SO).

When the timer stops operating, the timer output bit (TMCR0:TO1/TO0) holds the last value.

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

MB95330H Series
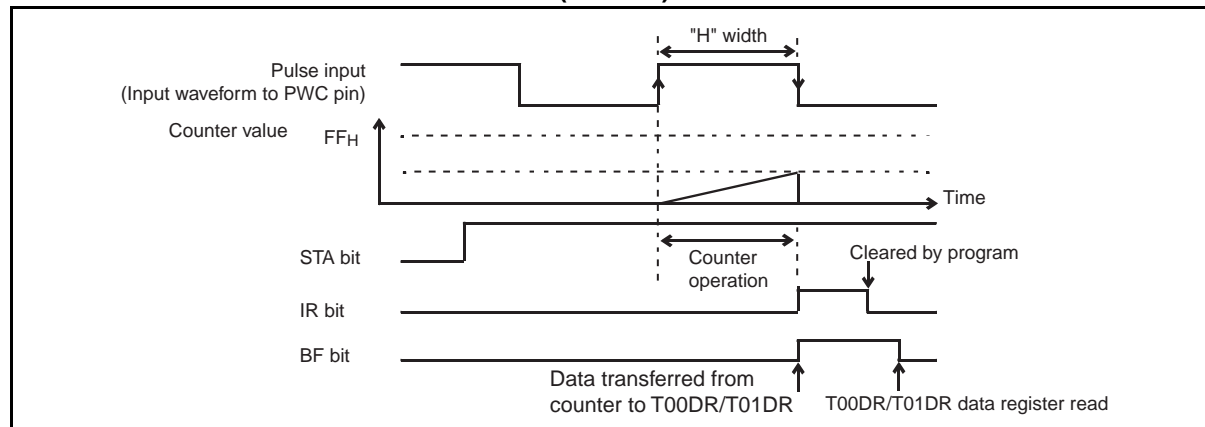
■ Operation of PWC Timer Function (Timer 1)

The settings shown in Figure 14.12-3 are required to use the PWC timer function.

Figure 14.12-3 Settings for PWC Timer Function (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	❍	❍	❍	❍
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	❍	❍	❍	❍	✕
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	❍	❍	❍	❍	❍	❍	❍	❍
T10DR/T11DR	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 (T10CR0/T11CR0: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 10/11 data register (T10DR/T11DR), and the interrupt flag (T10CR1/T11CR1:IR) and the buffer full flag (T10CR1/T11CR1:BF) are set to "1". The buffer full flag is set to "0" when the 8/16-bit composite timer 10/11 data register (T10DR/T11DR) is read.

If the buffer full flag is set to "1", the 8/16-bit composite timer 10/11 data register holds data. Even if the next edge is detected during that time, the next measurement result is lost since the count value has not been transferred to the 8/16-bit composite timer 10/11 data register.

There is an exception. With the F3 bit to F0 bit in the T10CR0/T11CR0 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 10/11 data register, while the cycle measurement result is not transferred to the 8/16-bit composite timer 10/11 data register. Therefore, in order to perform cycle measurement, the "H" pulse measurement result must be read before a cycle is completed. In addition, the result of "H" pulse measurement and that of cycle measurement are lost if they are not read before the completion of the next "H" pulse.

To measure the time exceeding the range of the counter, software can be used to count the number of counter overflows. When the counter overflows, the interrupt flag (T10CR1/T11CR1: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 (T10CR1/T11CR1:SO).

When the timer stops operating, the timer output bit (TMCR1:TO1/TO0) holds the last value.

**Figure 14.12-4 Operation Diagram of PWC Timer (Example of H-pulse Width Measurement)
(Timer 1)**

MB95330H Series
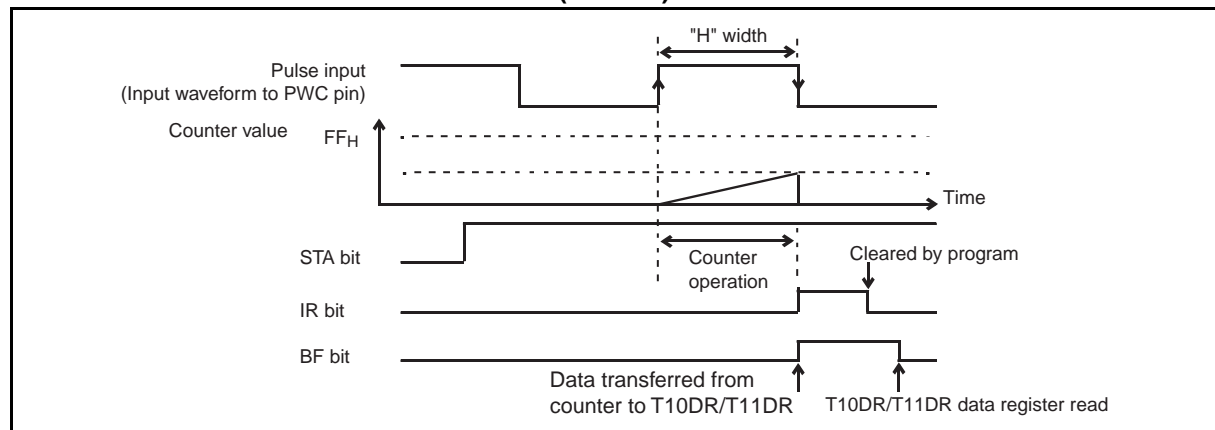
14.13   Operation of Input Capture Function

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 (T00DR/T01DR) immediately after an edge of the external signal input is detected. The target edge to be detected is selected by the timer operating mode select bits (T00CR0/T01CR0:F3, F2, F1, F0).

This function is available in free-run mode and clear mode, which can be selected by the timer operating mode select bits.

In clear mode, the counter starts counting from "00_H". When an edge is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR), the interrupt flag (T00CR1/T01CR1:IR) is set to "1", and the counter returns to "00_H" and restarts counting.

In free-run mode, when an edge is detected, the counter value is transferred to the 8/16-bit composite timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) is set to "1". In this case, the counter continues to count without being cleared.

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

To measure the time exceeding the range of the counter, software can be used to count the number of counter overflows. When the counter overflows, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of overflows. In addition, the timer output is inverted due to the overflow. The timer output initial value can be set by the timer output initial value bit (T00CR1/T01CR1:SO).

Note:

See Section 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)

MB95330H Series
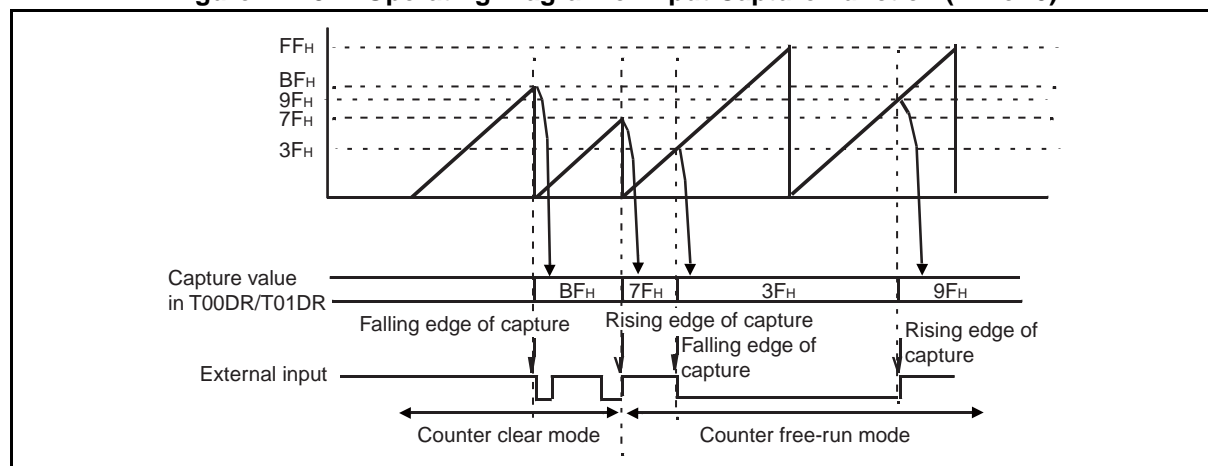
■ Operation of Input Capture Function (Timer 1)

The settings shown in Figure 14.13-3 are required to use the input capture function.

Figure 14.13-3 Settings for Input Capture Function (Timer 1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T10CR0/T11CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	❍	❍	❍	❍	❍	❍	❍	❍
T10CR1/T11CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	❍	❍	❍	×	❍	×	×
TMCR1	TO1	TO0	TIS	MOD	FE11	FE10	FE01	FE00
	×	×	❍	❍	❍	❍	❍	❍
T10DR/T11DR	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 10/11 data register (T10DR/T11DR) 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 (T10CR0/T11CR0: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 10/11 data register (T10DR/T11DR), the interrupt flag (T10CR1/T11CR1: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 10/11 data register (T10DR/T11DR) and the interrupt flag (T10CR1/T11CR1: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 (T10CR1/T11CR1: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 (T10CR1/T11CR1: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 (T10CR1/T11CR1:SO).

Note:

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

Figure 14.13-4 Operating Diagram of Input Capture Function (Timer 1)

MB95330H Series
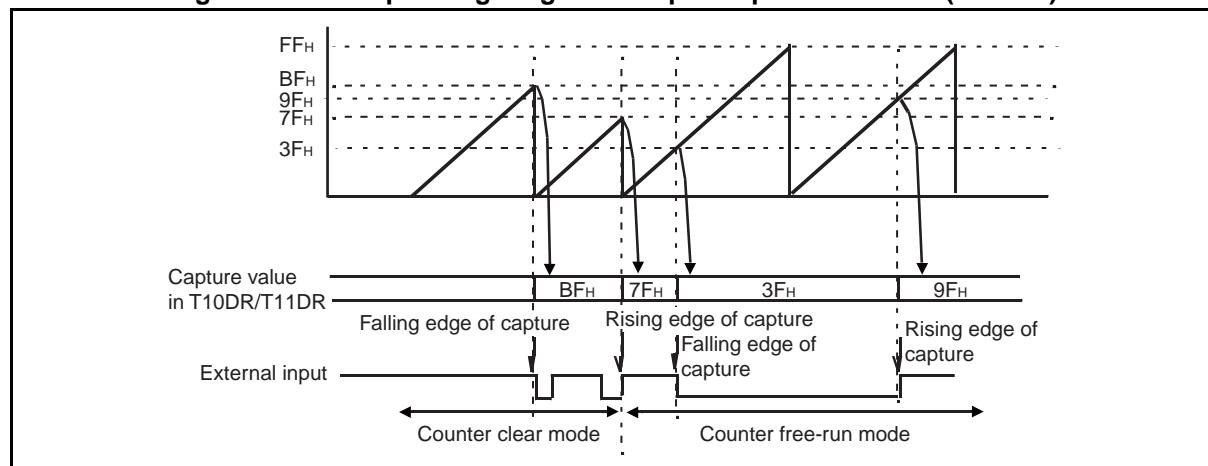
14.14 Operation of Noise Filter

This section describes the operation of the noise filter of the 8/16-bit composite timer.

When the input capture function or PWC timer function is selected, a noise filter can be used to eliminate the pulse noise of the signal from the external input pin (EC0/EC1). H-pulse noise, L-pulse noise, or H/L-pulse noise elimination can be selected by setting the FE11, FE10, FE01 and FE00 bits in the TMCR0/TMCR1 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
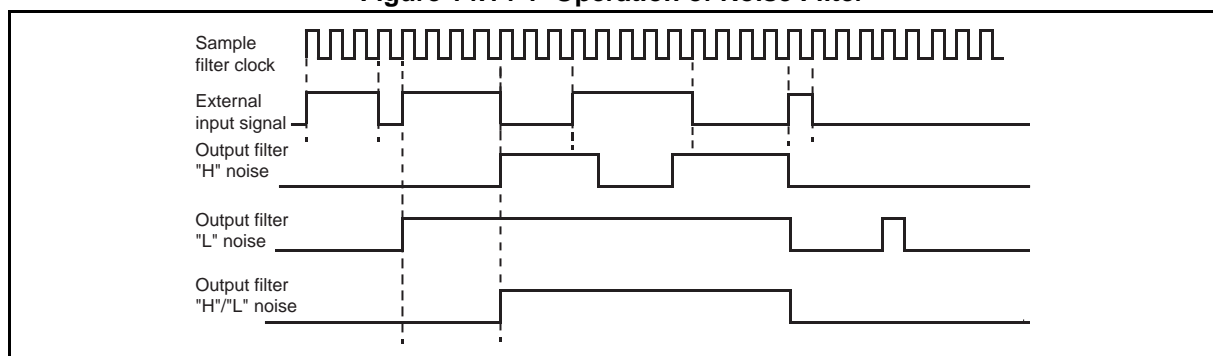
14.15 States in Each Mode during Operation

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/T10CR1/T11CR1:HO = 1) request is made during operation.

■ When Interval Timer, Input Capture, or PWC Function Is Selected

Figure 14.15-1 shows how the counter value changes when the microcontroller transits to watch mode or stop mode, or a suspend request is made during the operation of the 8/16-bit composite timer.

The counter stops operating while holding the value when the microcontroller transits to stop mode or watch mode. When the stop mode or watch mode is released by an interrupt, the counter resumes operating with the last value that it holds. Therefore, the first interval time or the initial external clock count value is incorrect. Always initialize the counter value after the microcontroller is released from stop mode or watch mode.

Figure 14.15-1 Operations of Counter in Standby Mode or in Pause (Not Serving as PWM Timer)
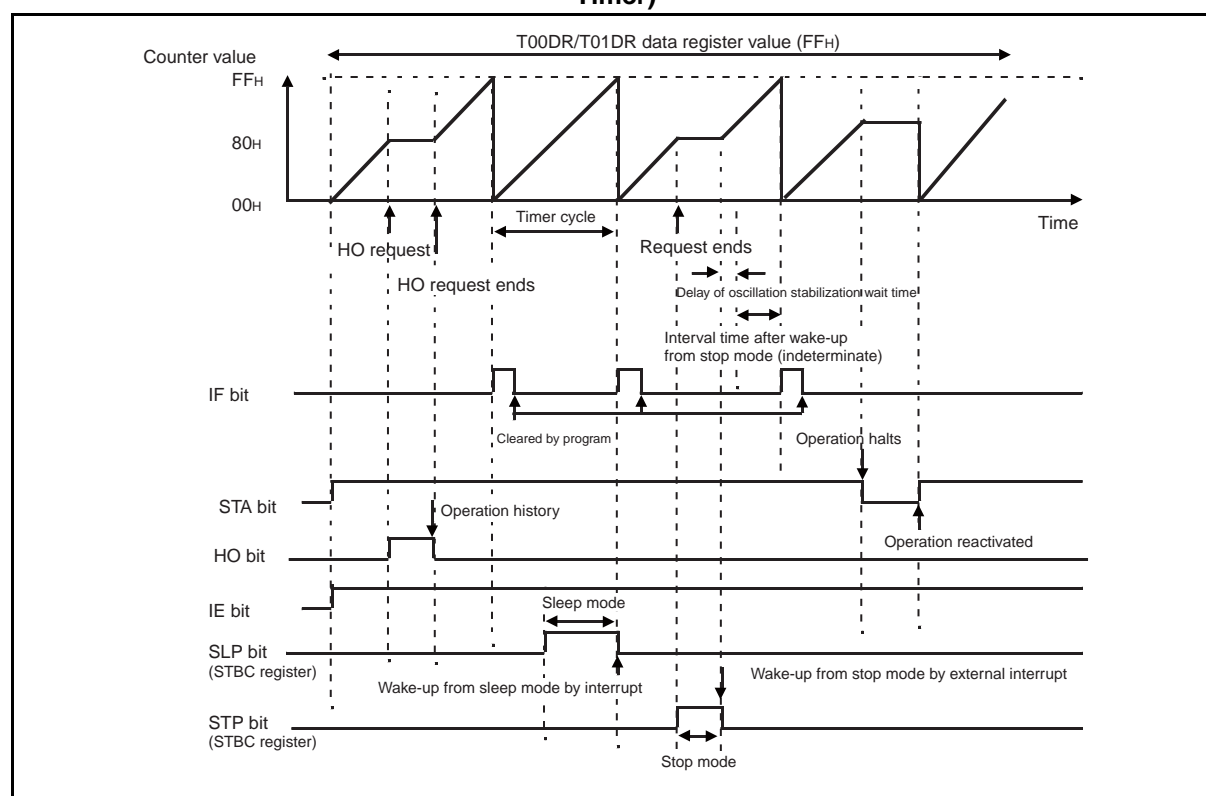
Figure 14.15-2 Operations of Counter in Standby Mode or in Pause (Serving as PWM Timer)
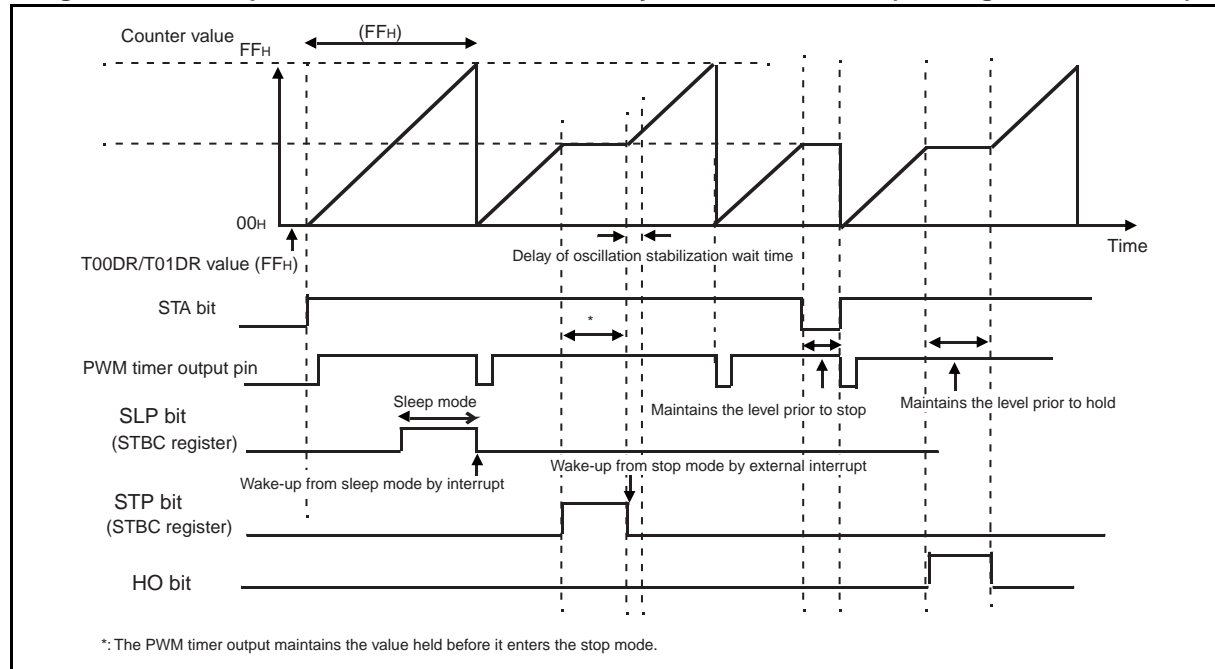
*: The PWM timer output maintains the value held before it enters the stop mode.

14.16 Notes on Using 8/16-bit Composite Timer

This section provides notes on using the 8/16-bit composite timer.

■ Notes on Using 8/16-bit Composite Timer

- To switch the timer function with the timer operating mode select bits (T00CR0/T01CR0/T10CR0/T11CR0:F3, F2, F1, F0), stop the timer operation first (T00CR1/T01CR1/T10CR1/T11CR1:STA = 0), then clear the interrupt flag (T00CR1/T01CR1/T10CR1/T11CR1:IF, IR), the interrupt enable bits (T00CR1/T01CR1/T10CR1/T11CR1:IE, T00CR0/T01CR0/T10CR0/T11CR0:IFE) and the buffer full flag (T00CR1/T01CR1/T10CR1/T11CR1: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/T10CR0/T11CR0: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/T10DR/T11DR), and pulse width measurement completion/edge detection flag (T00CR1/T01CR1/T10CR1/T11CR1: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/TMCR1:MOD = 0) of the PWM timer function (variable-cycle mode), when modifying the 8/16-bit composite timer 00/01 data register ch. 0 (T00DR/T01DR) during counter operation, modify T01DR first and then T00DR. The same setting sequence requirement is also applicable to the 8/16-bit composite timer 10/11 data register ch. 1 (T10DR/T11DR).

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.

MB95330H Series

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
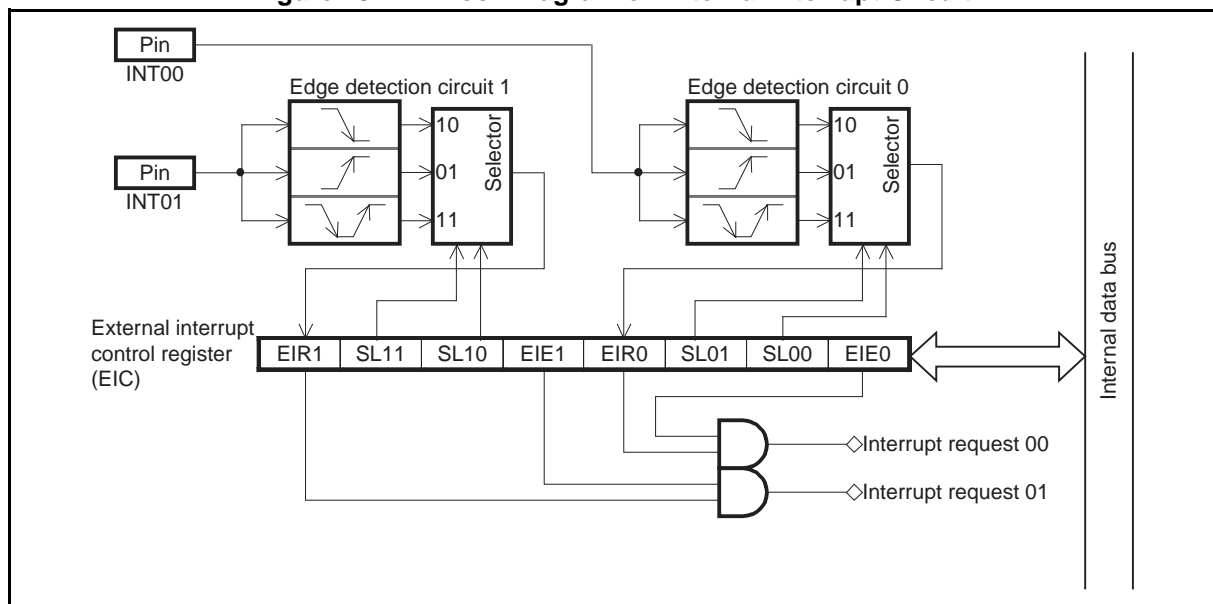
● Edge detection circuit

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 MB95330H Series has five units of external interrupt circuit.

Table 15.3-1 shows the pins of the external interrupt circuit and Table 15.3-2 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
4	INT08	External interrupt input ch. 8
	INT09	External interrupt input ch. 9

Table 15.3-2 Registers of External Interrupt Circuit

Unit	Register abbreviation	Corresponding register (Name in this manual)
0	EIC00	
1	EIC10	
2	EIC20	EIC: External Interrupt Control register
3	EIC30	
4	EIC01	

In the following sections, only details of unit 0 of the external interrupt circuit are provided.

Details of other units of the external interrupt circuit are the same as those of unit 0.

MB95330H Series

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 MB95330H Series, the pins of the external interrupt circuit are the INT00 to INT09 pins.

● INT00 to INT09 pins

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

INT00 to INT09:   If a pin of INT00 to INT09 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 INT09).

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 INT01 (P00/INT00/AN00, P01/INT01/AN01) of External Interrupt Circuit
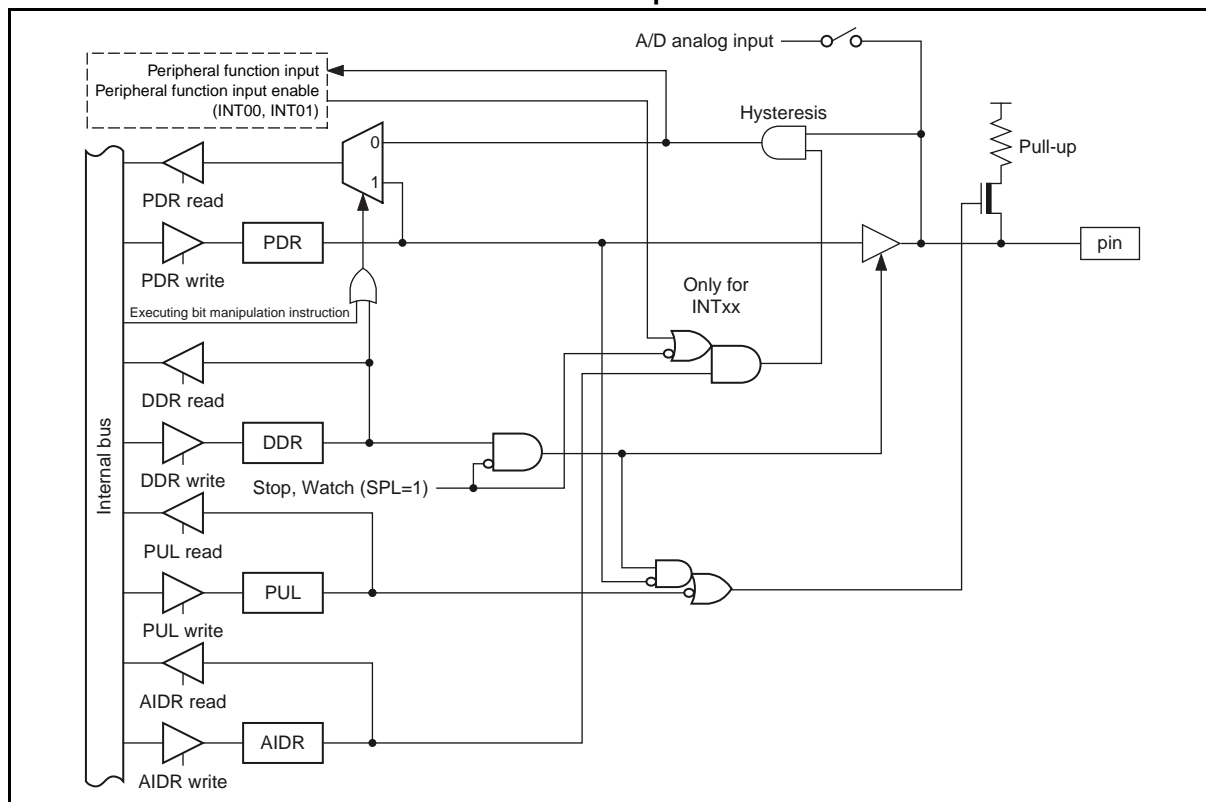
Figure 15.4-2 Block Diagram of Pins INT02, INT03 and INT05 (P02/INT02/AN02/SCK, P03/INT03/AN03/SOT, P05/INT05/AN05/TO00/HCLK2) of External Interrupt Circuit

MB95330H Series

Figure 15.4-3 Block Diagram of Pin INT04 (P04/INT04/AN04/SIN /HCLK1/EC0) of External Interrupt Circuit

Figure 15.4-4 Block Diagram of Pin INT06 (P06/INT06/AN06/TO01) of External Interrupt Circuit
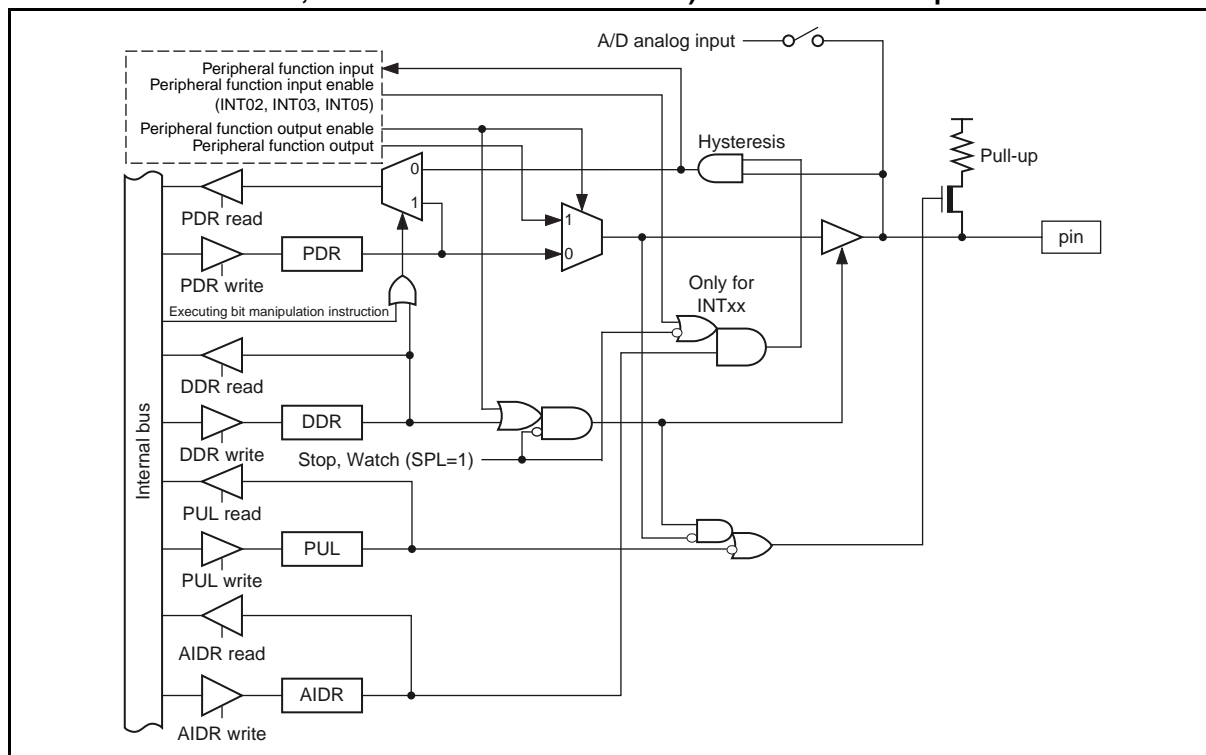
Figure 15.4-5 Block Diagram of Pin INT07 (P07/INT07/AN07) of External Interrupt Circuit
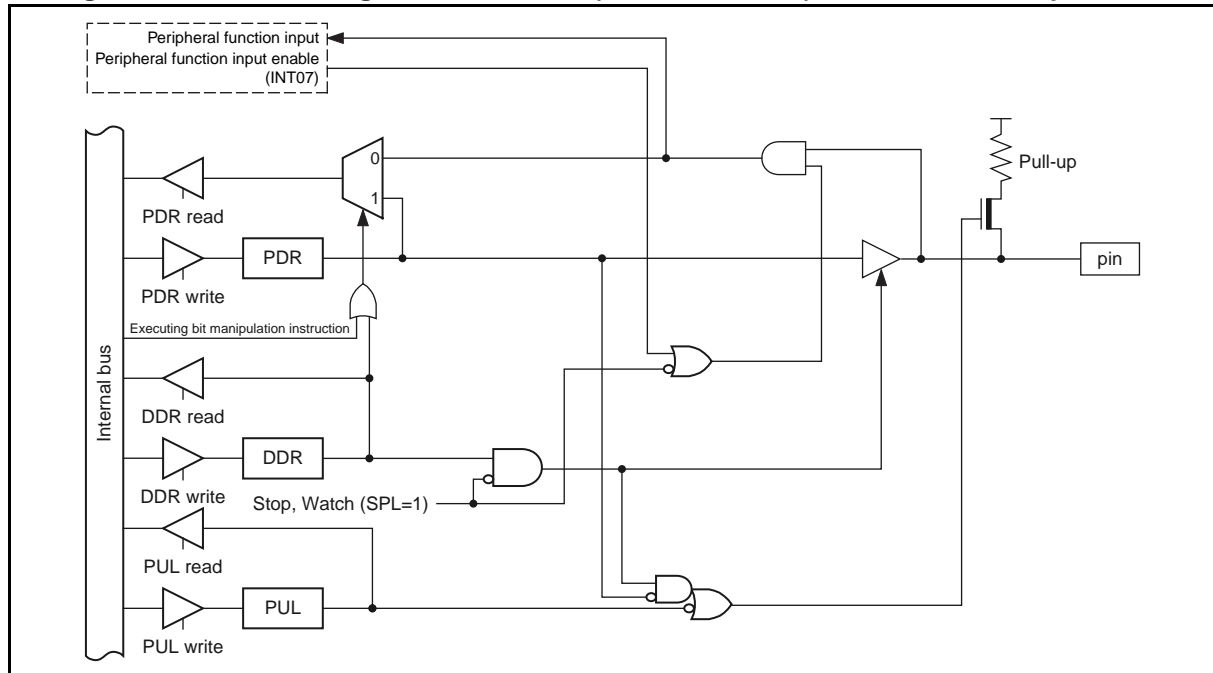
Figure 15.4-6 Block Diagram of Pins INT08 and INT09 (P60/INT08/SDA/DTTI, P61/INT09/SCL/TI1) of External Interrupt Circuit

MB95330H Series
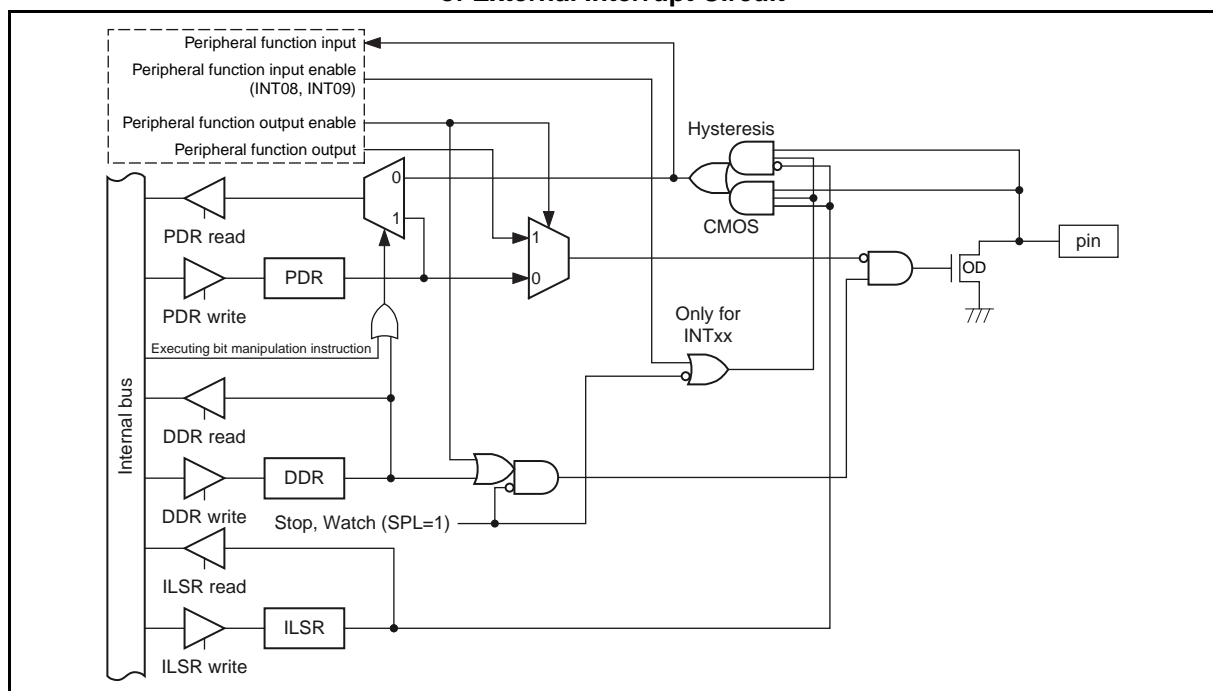
15.5 Registers of External Interrupt Circuit

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	
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC01	004C$_H$	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.)

15.5.1 External Interrupt Control Register (EIC00)

The external interrupt control register (EIC00) is used to select the edge polarity for the external interrupt input and control interrupts. Except for addresses, the configuration of the EIC registers (EIC01, EIC10, EIC20 and EIC30) of other units is identical to that of EIC00.

■ **External Interrupt Control Register (EIC00)**

Figure 15.5-2 External Interrupt Control Register (EIC00)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
EIC00	0048H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000B
EIC10	0049H									
EIC20	004AH	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
EIC30	004BH									
EIC01	004CH									

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

MB95330H Series

Table 15.5-1 Functions of Bits in External Interrupt Control Register (EIC00)

Bit name		Function
bit7	EIR1: External interrupt request flag bit 1	This flag is set to "1" when the edge selected by the edge polarity select bits (SL11, SL10) is input to the external interrupt pin INT01. • When this bit and the interrupt request enable bit 1 (EIE1) are set to "1", an interrupt request is 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 INT01. 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 INT00. • 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 INT00. 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					
External interrupt ch. 8	IRQ21	ILR5	L21	$FFD0_H$	$FFD1_H$
External interrupt ch. 9					

ch.: Channel

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

MB95330H Series

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 (INT00, INT01) matches the polarity of the edge selected by the external interrupt control register (EIC: SL00, SL01, SL10 and SL11), the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1" and the interrupt request is generated.

Always set the interrupt request enable bit to "0" when not using an external interrupt to wake up the device from standby mode.

When setting the edge polarity select bit (SL), set the interrupt request enable bit (EIE) to "0" to prevent the interrupt request from being generated accidentally. Also clear the interrupt request flag bit (EIR) to "0" after changing the edge polarity.

Figure 15.7-1 shows the operations for setting the INT00 pin as an external interrupt input.

Figure 15.7-1 Operations of External Interrupt
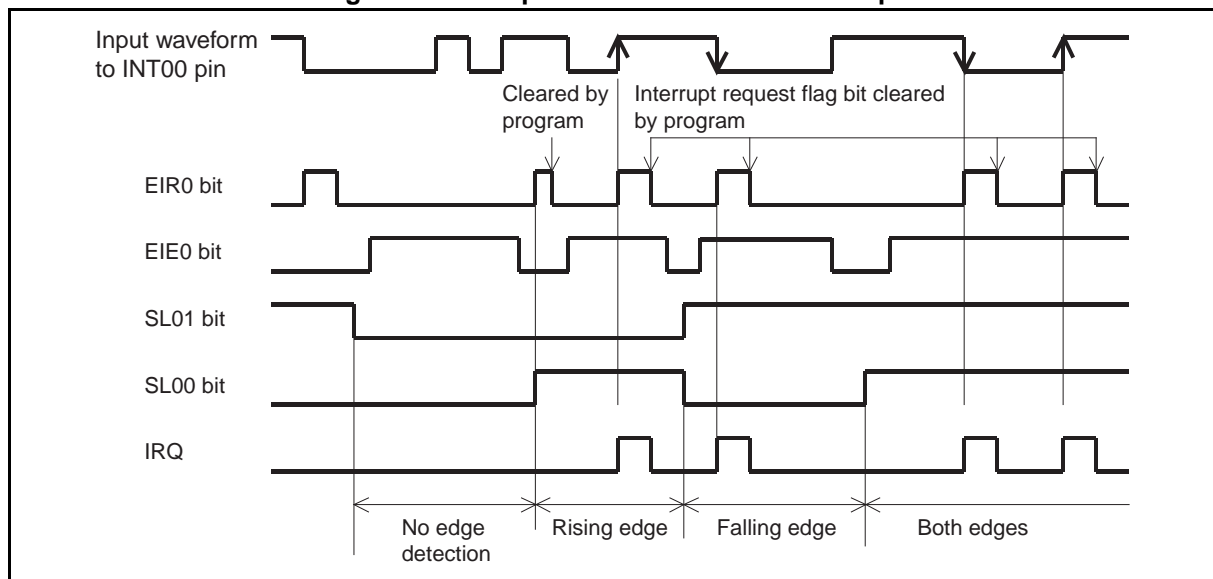
■ Setting Procedure Example

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).

MB95330H Series

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 Sample Settings for External Interrupt Circuit

This section provides sample settings for the external interrupt circuit.

■ Sample Settings

● Detection levels and setting methods

Four detection levels are available: no edge detection, rising edge, falling edge, both edges

The detection level bits (EIC: SL01, SL00 or EIC: SL11, SL10) are used.

Operating mode	Detection level bits (SL01,SL00)
No edge detection	Set 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 or DDR6) to "0".

Operation	Direction bit (P00 to P07, P60 and P61)	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".
Using INT08 pin for external interrupt	DDR6: P60	Set to "0".
Using INT09 pin for external interrupt	DDR6: P61	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: $0FFF4_H$
ch. 8	Interrupt level register (ILR5) Address: $0007E_H$	#21 Address: $0FFD0_H$
ch. 9	Interrupt level register (ILR5) Address: $0007E_H$	#21 Address: $0FFD0_H$

● 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 request	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 (TRG1, UCK0, UI0, EC1 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.

MB95330H Series

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
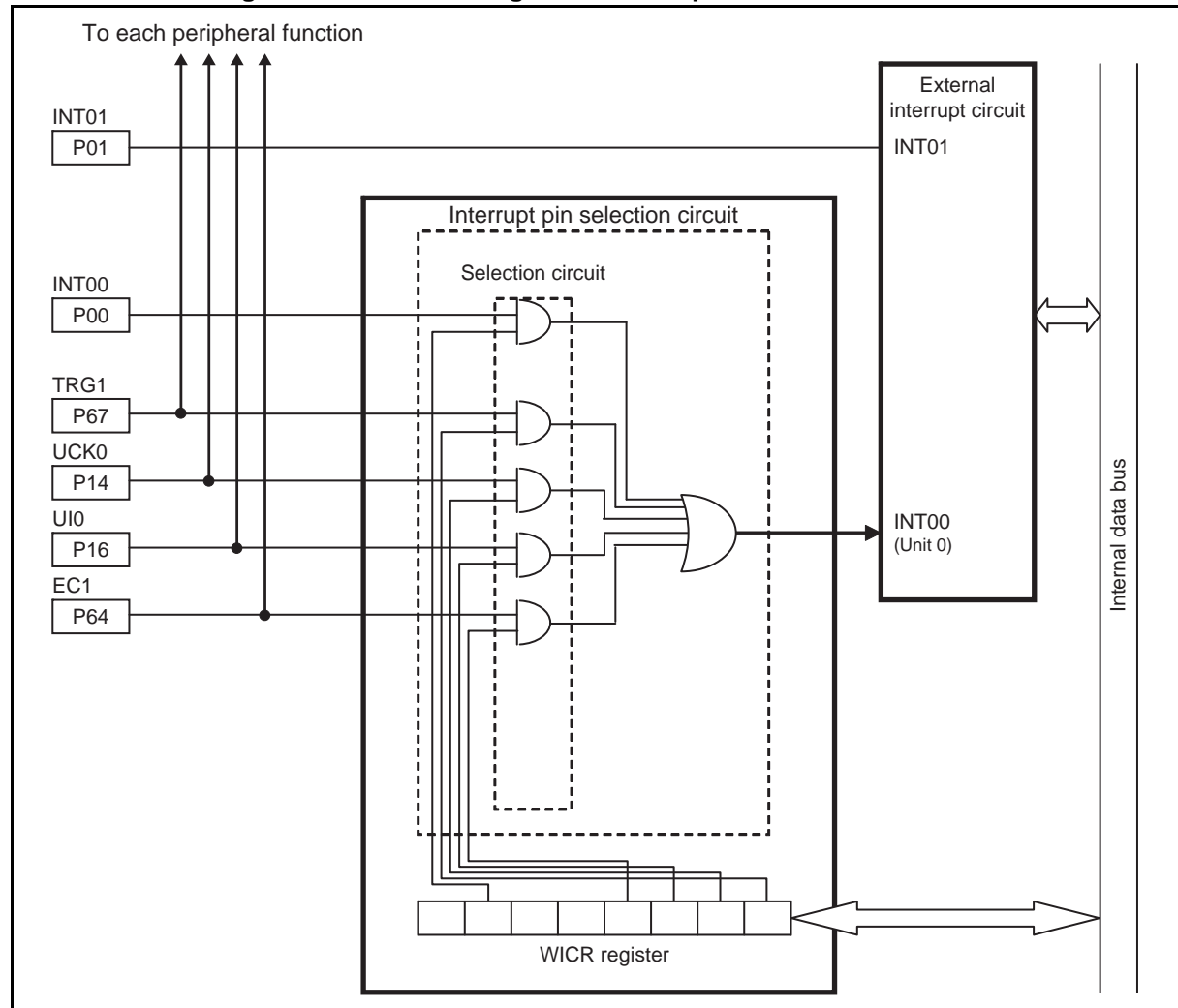
- WICR register (interrupt pin selection circuit control register)

  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 TRG1, UCK0, UI0, EC1 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
TRG1	16-bit PPG timer (trigger input)
UCK0	UART/SIO (clock input/output)
UI0	UART/SIO (data input)
EC1	8/16-bit composite timer (event input)

MB95330H Series

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	-	-	EC1	UI0	UCK0	TRG1	01000000_B
	R0/WX	R/W	R0/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.)
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	-	-	EC1	UI0	UCK0	TRG1	01000000B
	R0/WX	R/W	R0/WX	R0/WX	R/W	R/W	R/W	R/W	

TRG1	TRG1 interrupt pin select bit
0	Deselects TRG1 as interrupt input pin
1	Selects TRG1 as interrupt input pin

UCK0	UCK0 interrupt pin select bit
0	Deselects UCK0 as interrupt input pin
1	Selects UCK0 as interrupt input pin

UI0	UI0 interrupt pin select bit
0	Deselects UI0 as interrupt input pin
1	Selects UI0 as interrupt input pin

EC1	EC1 interrupt pin select bit
0	Deselects EC1 as interrupt input pin
1	Selects EC1 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

MB95330H Series

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, bit4	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation
bit3	EC1: EC1 interrupt pin select bit	This bit is used to determine whether to select the EC1 pin as an interrupt input pin. **Writing "0"** to the bit deselects the EC1 pin as an interrupt input pin and the circuit treats the EC1 pin input as being fixed at "0". **Writing "1"** to the bit selects the EC1 pin as an interrupt input pin and the circuit passes the EC1 pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the EC1 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit.
bit2	UI0: UI0 interrupt pin select bit	This bit is used to determine whether to select the UI0 pin as an interrupt input pin. **Writing "0"** to the bit deselects the UI0 pin as an interrupt input pin and the circuit treats the UI0 pin input as being fixed at "0". **Writing "1"** to the bit selects the UI0 pin as an interrupt input pin and the circuit passes the UI0 pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the UI0 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit.
bit1	UCK0: UCK0 interrupt pin select bit	This bit is used to determine whether to select the UCK0 pin as an interrupt input pin. **Writing "0"** to the bit deselects the UCK0 pin as an interrupt input pin and the circuit treats the UCK0 pin input as being fixed at "0". **Writing "1"** to the bit selects the UCK0 pin as an interrupt input pin and the circuit passes the UCK0 pin input to INT00 (ch. 0) of the external interrupt circuit. In this case, the input signal to the UCK0 pin can generate an external interrupt if INT00 (ch. 0) operation is enabled in the external interrupt circuit.
bit0	TRG1: TRG1 interrupt pin select bit	This bit is used to determine whether to select the TRG1 pin as an interrupt input pin. **Writing "0"** to the bit deselects the TRG1 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 TRG1 pin as an interrupt input pin and the circuit passes the TRG1 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.

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 Section 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 set to SL01, SL00 bits in 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 TRG1 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

LIN-UART

This chapter describes the functions and operations of the LIN-UART.

17.1 Overview of LIN-UART

The LIN (Local Interconnect Network)-UART is a general-purpose serial data communication interface for synchronous or asynchronous (start-stop synchronization) communication with external devices. In addition to a bi-directional communication function (normal mode) and master/slave communication function (multiprocessor mode: supports both master and slave operation), the LIN-UART also supports special functions with the LIN bus.

■ Functions of LIN-UART

The LIN-UART is a general-purpose serial data communication interface for exchanging serial data with other CPUs and peripheral devices. Table 17.1-1 lists the functions of the LIN-UART.

Table 17.1-1 Functions of LIN-UART

	Function
Data buffer	Full-duplex double-buffer
Serial input	The LIN-UART oversamples received data for five times to determine the received value by majority of sampling values (only asynchronous mode).
Transfer mode	• Clock-synchronous (Select start/stop synchronization, or start/stop bits) • Clock-asynchronous (Start/stop bits available)
Baud rate	• Dedicated baud rate generator provided (made of a 15-bit reload counter) • The external clock can be input. It can be adjusted by the reload counter.
Data length	• 7 bits (not in synchronous or LIN mode) • 8 bits
Signal type	NRZ (Non Return to Zero)
Start bit timing	Synchronization with the start bit falling edge in asynchronous mode.
Reception error detection	• Framing error • Overrun error • Parity error (Not supported in operating mode 1)
Interrupt request	• Receive interrupts (reception completed, reception error detected, LIN synch break detected) • Transmit interrupts (transmit data empty) • Interrupt requests to TII0 (LIN synch field detected: LSYN)
Master/slave mode communication function (Multiprocessor mode)	Capable of 1 (master) to n (slaves) communication (supports both the master and slave system)
Synchronous mode	Transmit side/receive side of serial clock
Pin access	Serial I/O pin states can be read directly.
LIN bus option	• Master device operation • Slave device operation • LIN synch break detection • LIN synch break generation • Detection of LIN synch field start/stop edges connected to the 8/16-bit composite timer
Synchronous serial clock	Continuous output to the SCK pin enabled for synchronous communication using the start/stop bits
Clock delay option	Special synchronous clock mode for delaying the clock (used in Special Peripheral Interface (SPI))

MB95330H Series

The LIN-UART operates in four different modes. The operating mode is selected by the MD0 and MD1 bits in the LIN-UART serial mode register (SMR). Operating mode 0 and operating mode 2 are used for bi-directional serial communication; mode 1 for master/slave communication; and mode 3 for LIN master/slave communication.

Table 17.1-2 LIN-UART Operating Modes

Operating mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multiprocessor mode	7 bits or 8 bits +1*	-	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	
3	LIN mode	8 bits	-	Asynchronous	1 bit	LSB first

- : Unavailable

* : "+1" is the address/data select bit (AD) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operating modes.

Table 17.1-3 LIN-UART Operating Modes

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

- Mode 1 supports both master and slave operation for the multiprocessor mode.
- The communication format of Mode 3 is fixed: 8-bit data, no parity, stop bit 1, LSB-first.

17.2   Configuration of LIN-UART

LIN-UART is made up of the following blocks.
- **Reload counter**
- **Receive control circuit**
- **Receive shift register**
- **LIN-UART receive data register (RDR)**
- **Transmit control circuit**
- **Transmit shift register**
- **LIN-UART transmit data register (TDR)**
- **Error detection circuit**
- **Oversampling circuit**
- **Interrupt generation circuit**
- **LIN synch break/synch field detection circuit**
- **Bus idle detection circuit**
- **LIN-UART serial control register (SCR)**
- **LIN-UART serial mode register (SMR)**
- **LIN-UART serial status register (SSR)**
- **LIN-UART extended status control register (ESCR)**
- **LIN-UART extended communication control register (ECCR)**

MB95330H Series

■ **Block Diagram of LIN-UART**

Figure 17.2-1 Block Diagram of LIN-UART
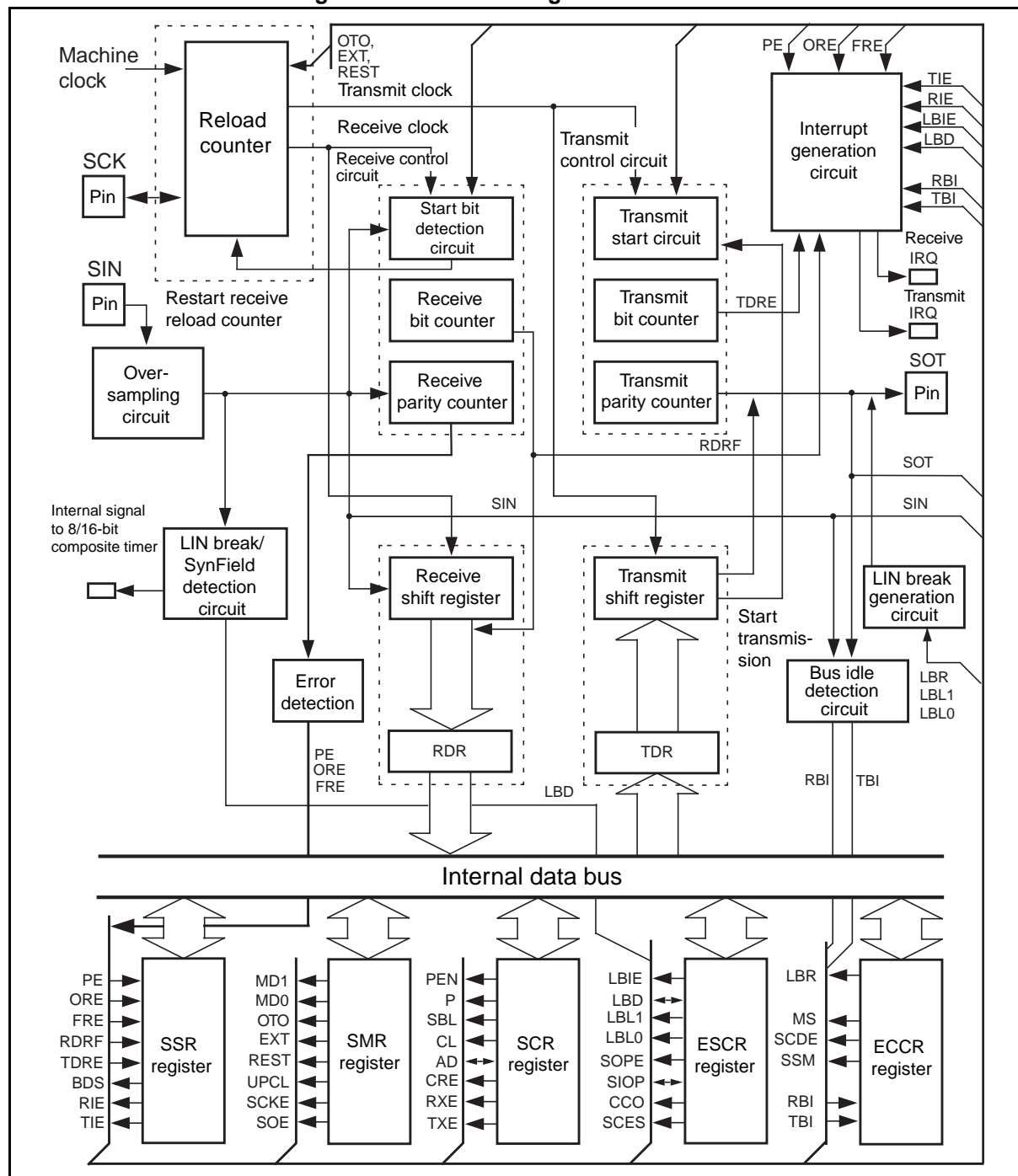
● Reload counter

> This block is a 15-bit reload counter functioning as a dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/receive clock from the external or internal clock. The count value in the transmit reload counter is read from the baud rate generator1, 0 (BGR 1 and BGR 0).

● Receive control circuit

This block consists of a receive bit counter, a start bit detection circuit, and a receive parity counter. The receive bit counter counts the receive data bits and sets a flag in the LIN-UART receive data register when the reception of one data is completed according to the specified data length. If the receive interrupt has been enabled, a receive interrupt request is made. The start bit detection circuit detects a start bit in a serial input signal. When a start bit is detected, the circuit sends a signal to the reload counter in synchronization with the start bit falling edge. The receive parity counter calculates the parity of the received data.

● Receive shift register

The circuit captures received data from the SIN pin while performing bit shifting of received data. The receive shift register transfers received data to the RDR register.

● LIN-UART receive data register (RDR)

This register retains the received data. Serial input data is converted and stored in the LIN-UART receive data register.

● Transmit control circuit

This block consists of a transmit bit counter, a transmit start circuit, and a transmit parity counter. The transmit bit counter counts the transmit data bits and sets a flag in the transmit data register when the transmission of one data is completed according to the specified data length. If the transmit interrupt has been enabled, a transmit interrupt request is made. The transmit start circuit starts transmission when data is written to the TDR. The transmit parity counter generates a parity bit for data to be transmitted if the data has a parity.

● Transmit shift register

Data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register, and then the transmit shift register outputs the data to the SOT pin while performing bit shifting of the data.

● LIN-UART transmit data register (TDR)

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

● Error detection circuit

This circuit detects errors occurring at the end of reception. If an error occurs, a corresponding error flag is set.

● Oversampling circuit

In asynchronous mode, the oversampling circuit oversamples received data for five times to determine the received value by majority of sampling values. The circuit stops operating in synchronous mode.

● Interrupt generation circuit

This circuit controls all interrupt sources. An interrupt is generated immediately provided that

the corresponding interrupt enable bit has been set.

● LIN synch break/synch field detection circuit

This circuit detects a LIN synch break when the LIN master node transmits a message header. The LBD flag is set when the LIN synch break is detected. An internal signal is output to 8/16-bit composite timer in order to detect the first and the fifth falling edges of the LIN synch field and to measure the actual serial clock synchronization transmitted by the master node.

● LIN synch break generation circuit

This circuit generates a LIN synch break with a length set.

● Bus idle detection circuit

If this circuit detects that no transmission or reception is in progress, it sets the TBI flag bit or the RBI flag bit to "1" respectively.

● LIN-UART serial control register (SCR)

Its operating functions are as follows:

- Setting the use of the parity bit
- Parity bit select
- Setting stop bit length
- Setting data length
- Selecting the frame data format in mode 1
- Clearing the error flag
- Enabling/disabling transmission
- Enabling/disabling reception

● LIN-UART serial mode register (SMR)

Its operating functions are as follows:

- Selecting the LIN-UART operating mode
- Selecting the clock input source
- Selecting between one-to-one connection to the external clock and connection to the reload counter
- Resetting the dedicated reload timer
- LIN-UART software reset (maintaining register settings)
- Enabling/disabling output to the serial data pin
- Enabling/disabling output to the clock pin

● LIN-UART serial status register (SSR)

Its operating functions are as follows:

- Checking transmission/reception or error status
- Selecting the transfer direction (LSB-first or MSB-first)
- Enabling/disabling receive interrupts
- Enabling/disabling transmit interrupts

● Extended status control register (ESCR)

Its operating functions are as follows:

- Enabling/disabling LIN synch break interrupts
- LIN synch break detection
- Selecting LIN synch break length
- Direct access to SIN pin and SOT pin
- Setting continuous clock output in LIN-UART synchronous clock mode
- Sampling clock edge selection

● LIN-UART extended communication control register (ECCR)

Its operating functions are as follows:

- Bus idle detection
- Synchronous clock setting
- LIN synch break generation

■ Input Clock

The LIN-UART uses a machine clock or an input signal from the SCK pin as an input clock.

The input clock is used as the transmission/reception clock source of the LIN-UART.

MB95330H Series

17.3   LIN-UART Pins

This section describes LIN-UART pins.

■ LIN-UART Pins

The LIN-UART pins are also used as general-purpose ports. Table 17.3-1 lists the LIN-UART pin functions and settings for using them.

Table 17.3-1 LIN-UART Pins

Pin name	Pin function	Settings required for using pin
SIN	Serial data input	Set to the input port (DDR: corresponding bit = 0)
SOT	Serial data output	Enable output. (SMR:SOE = 1)
SCK	Serial clock input/output	Set to the input port when this pin is used for clock input. (DDR: corresponding bit = 0)
		Enable output when this pin is used as an clock output pin. (SMR:SCKE = 1)

■ Block Diagrams of LIN-UART Pins

Figure 17.3-1 Block Diagram of Pins SCK and SOT(P02/INT02/AN02/SCK and P03/INT03/AN03/SOT) of LIN-UART
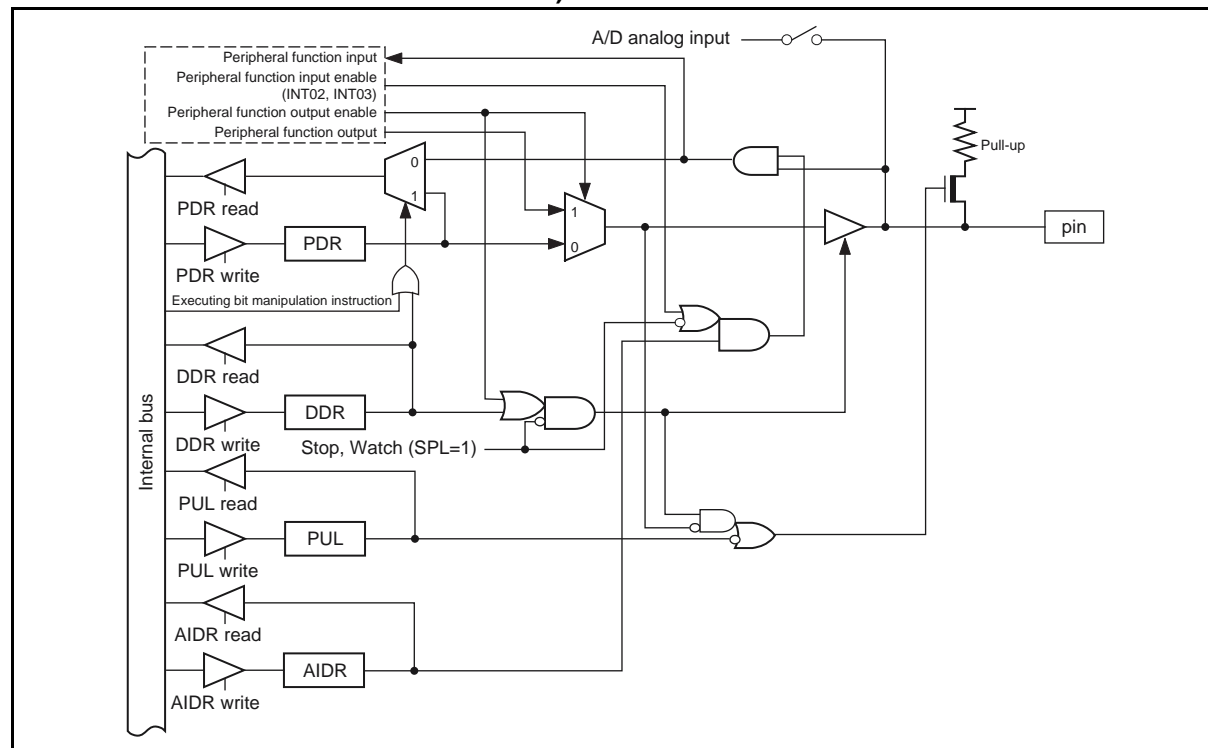
Figure 17.3-2 Block Diagram of Pin SIN (P04/INT04/AN04/SIN/HCLK1/EC0) of LIN-UART

MB95330H Series
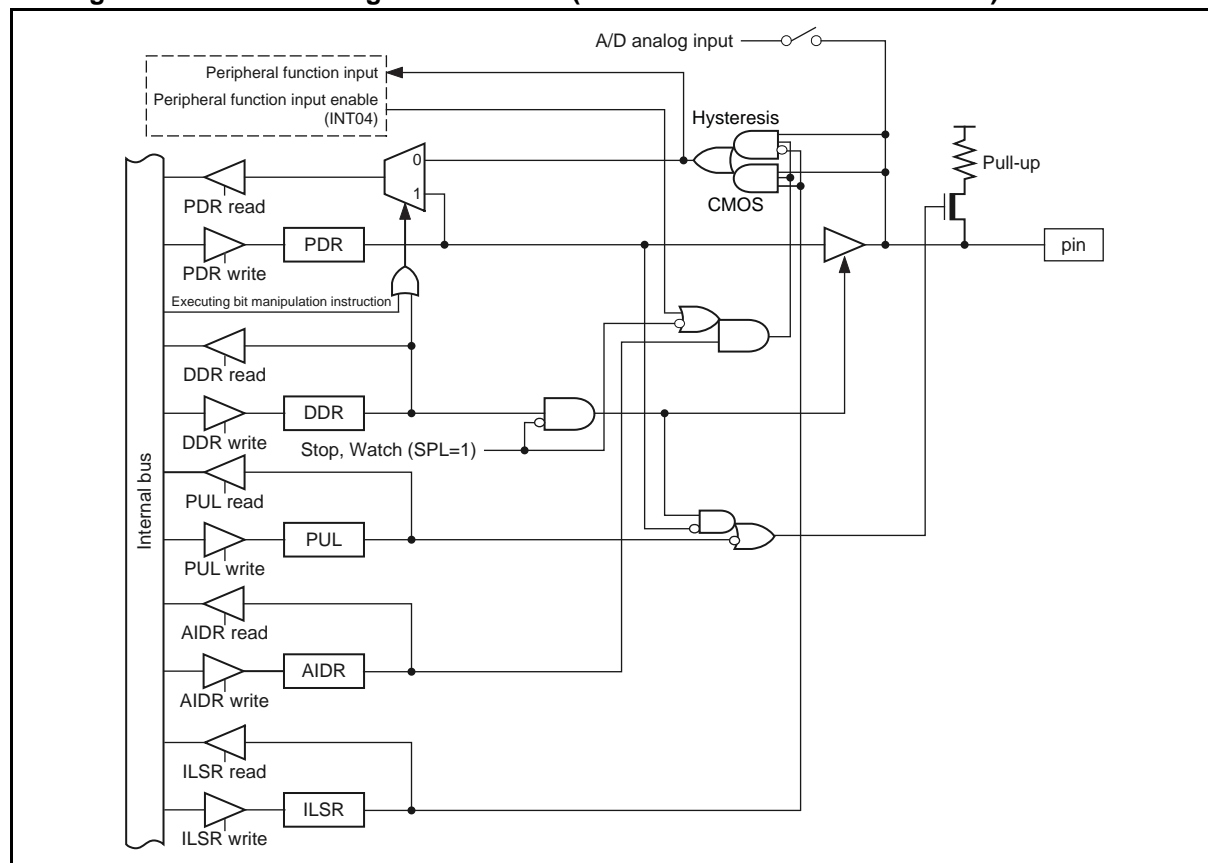
17.4 Registers of LIN-UART

This section lists the registers of the LIN-UART.

■ Registers of LIN-UART

Figure 17.4-1 Registers of LIN-UART

LIN-UART serial control register (SCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050_H	PEN	P	SBL	CL	AD	CRE	RXE	TXE	00000000_B
	R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W	

LIN-UART serial mode register (SMR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0051_H	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE	00000000_B
	R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W	

LIN-UART serial status register (SSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0052_H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	00001000_B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W	

LIN-UART receive data register/transmit data register (RDR/TDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053_H									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LIN-UART extended status control register (ESCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0054_H	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	00000100_B
	R/W	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	

LIN-UART extended communication control register (ECCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0055_H	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI	$000000XX_B$
	RX,W0	R0,W	R/W	R/W	R/W	RX,W0	R/WX	R/WX	

LIN-UART baud rate generator register 1 (BGR 1)

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

LIN-UART baud rate generator register 0 (BGR 0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$0FBD_H$									00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (The read value is the same as the write value.)
R(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.
RX,W0 : The read value is indeterminate; the write value is "0".
- ; Undefined bit

17.4.1 LIN-UART Serial Control Register (SCR)

The LIN-UART serial control register (SCR) is used to set parity, select the stop bit length and data length, select the frame data format in mode 1, clear the receive error flag, and enable/disable transmission/reception.

■ **LIN-UART Serial Control Register (SCR)**

Figure 17.4-2 LIN-UART Serial Control Register (SCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050H	PEN	P	SBL	CL	AD	CRE	RXE	TXE	00000000B
	R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W	

TXE	Transmit operation enable bit
0	Disable transmission
1	Enable transmission

RXE	Receive operation enable bit
0	Disable reception
1	Enable reception

CRE	Receive error flag clear bit	
	Write	Read
0	No effect	"0" is always read.
1	Clear receive error flag (PE, FRE, ORE)	

AD	Address/data format select bit
0	Data frame
1	Address frame

CL	Data length select bit
0	7-bit
1	8-bit

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

P	Parity select bit
0	Even parity
1	Odd parity

PEN	Parity enable bit
0	No parity
1	With parity

R/W	: Readable/writable (The read value is the same as the write value.)
R0,W	: Write only (Writable. The read value is "0".)
	: Initial value

Table 17.4-1 Functions of Bits in LIN-UART Serial Control Register (SCR)

Bit name		Function
bit7	PEN: Parity enable bit	This bit specifies whether or not to add (at transmission) and detect (at reception) a parity bit. Note: The parity bit is added only in operating mode 0, or in operating mode 2 in which the start/stop bits are to be added to the synchronous data format (ECCR:SSM = 1). This bit is fixed at "0" in operating mode 3 (LIN).
bit6	P: Parity select bit	With the parity bit having been enabled (SCR:PEN = 1), setting this bit to "1" selects the odd parity and setting this bit to "0" selects the even parity.
bit5	SBL: Stop bit length select bit	This bits sets the bit length of the stop bit (frame end mark in transmit data) in operating mode 0, 1 (asynchronous) or in operating mode 2 (synchronous) in which the start/stop bits are to be added to the synchronous data format (ECCR:SSM = 1). This bit is fixed at "0" in operating mode 3 (LIN). Note: At reception, only the first bit of the stop bit is always detected.
bit4	CL: Data length select bit	This bit specifies the data length to be transmitted and received. This bit is fixed at "1" in operating mode 2 and operating mode 3.
bit3	AD: Address/data format select bit	This bit specifies the data format for the frame to be transmitted and received in multiprocessor mode (mode 1). Write a value to this bit in master mode; read this bit in slave mode. The operation in master mode is as follows. **Writing "0":** the data frame is used as the data format. **Writing "1":** the address data frame is used as the data format. The value for the last received data format is read. Note: See Section 17.8 "Notes on Using LIN-UART" for the usage of this bit.
bit2	CRE: Receive error flag clear bit	This bit clears the FRE, ORE, and PE flags in serial status register (SSR). **Writing "0":** has no effect on operation. **Writing "1":** clears the error flag. When this bit is read, it always returns "0".
bit1	RXE: Receive operation enable bit	This bits enables or disables the reception of the LIN-UART. **Writing "0":** disables data frame reception. **Writing "1":** enables data frame reception. The LIN synch break detection in operating mode 3 is not affected by the setting of this bit. Note: When data frame reception is disabled (RXE = 0) while it is in progress, the reception halts immediately. In this case, the integrity of data is not guaranteed.
bit0	TXE: Transmit operation enable bit	This bits enables or disables the transmission of the LIN-UART. **Writing "0":** disables data frame transmission. **Writing "1":** enables data frame transmission. Note: When data frame transmission is disabled (TXE = 0) while it is in progress, the transmission halts immediately. In this case, the integrity of data is not guaranteed.

17.4.2 LIN-UART Serial Mode Register (SMR)

The LIN-UART serial mode register (SMR) is used to select the operating mode, specify the baud rate clock, and enable/disable output to the serial data and clock pins.

■ LIN-UART Serial Mode Register (SMR)

Figure 17.4-3 LIN-UART Serial Mode Register (SMR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0051H	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE	00000000B
	R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W	

SOE	LIN-UART serial data output enable bit
0	General-purpose I/O port
1	LIN-UART serial data output pin

SCKE	LIN-UART serial clock output enable bit
0	General-purpose I/O port or LIN-UART clock input pin
1	LIN-UART serial clock output pin

UPCL	LIN-UART programmable clear bit	
	Write	Read
0	No effect on operation.	"0" is always read.
1	LIN-UART reset	

REST	Reload counter restart bit	
	Write	Read
0	No effect on operation.	"0" is always read.
1	Restarts the reload counter	

EXT	External serial clock source select bit
0	Uses the baud rate generator (reload counter).
1	Uses the external serial clock source.

OTO	One-to-one external clock input enable bit
0	Uses the baud rate generator (reload counter).
1	Uses the external clock directly.

MD1	MD0	Operating mode select bits
0	0	Mode 0: Asynchronous (Normal mode)
0	1	Mode 1: Asynchronous (Multiprocessor mode)
1	0	Mode 2: Synchronous (Normal mode)
1	1	Mode 3: Asynchronous (LIN mode)

R/W	: Readable/writable (The read value is the same as the write value.)
R0,W	: Write only (Writable. The read value is "0".)
	: Initial value

Table 17.4-2 Functions of Bits in LIN-UART Serial Mode Register (SMR)

Bit name		Function
bit7, bit6	MD1, MD0: Operating mode select bits	These bits sets the operating mode. Note: When the mode is changed during communication, exchanging on the LIN-UART is suspended and the LIN-UART waits for the start of the next communication. See table below.

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

Bit name		Function
bit5	OTO: One-to-one external clock input enable bit	**Writing "1":** enables the external clock to be used directly as the LIN-UART serial clock. In operating mode 2 (asynchronous), the external clock is used when the reception side of the serial clock is selected (ECCR:MS = 1). When EXT = 0, the OTO bit is fixed at "0".
bit4	EXT: External serial clock source select bit	This bit selects a clock input. **Writing "0":** selects the clock of the internal baud rate generator (reload counter). **Writing "1":** selects the external serial clock source.
bit3	REST: Reload counter restart bit	This bits restarts the reload counter. **Writing "0":** no effect on operation. **Writing "1":** restarts the reload counter. When this bit is read, it always returns "0".
bit2	UPCL: LIN-UART programmable clear bit (LIN-UART software reset)	This bit resets the LIN-UART. **Writing "0":** no effect on operation. **Writing "1":** resets the LIN-UART immediately (LIN-UART software reset). However, the register settings are maintained. At that time, transmission and reception are suspended. All of the transmit/receive interrupt sources (TDRE, RDRF, LBD, PE, ORE, FRE) are cleared. Reset the LIN-UART after disabling the interrupt and transmission. In addition, after the LIN-UART is reset, the receive data register is cleared (RDR = 00_H), and the reload counter is restarted. When this bit is read, it always returns "0".
bit1	SCKE: LIN-UART serial clock output enable bit	This bit controls the serial clock I/O port. **Writing "0":** the SCK pin functions as a general-purpose I/O port or a serial clock input pin. **Writing "1":** the SCK pin functions as a serial clock output pin, and outputs the clock in operating mode 2 (synchronous). Note: To use the SCK pin as a serial clock input pin (SCKE = 0), enable the use of the input port by setting the bit in the DDR register corresponding to the general-purpose I/O port sharing the same pin with SCK. In addition, select the external clock (EXT = 1) using the external serial clock source select bit. When set as a serial clock output pin (SCKE = 1), the SCK pin functions as a serial clock output pin regardless of the state of the general-purpose I/O port sharing the same pin with SCK.
bit0	SOE: LIN-UART serial data output enable bit	This bit enables or disables output of serial data. **Writing "0":** the SOT pin becomes a general-purpose I/O port. **Writing "1":** the SOT pin becomes a serial data output pin (SOT). When set as a serial data output (SOE = 1), the SOT pin functions as a serial data output pin (SOT) regardless of the state of the general-purpose I/O port sharing the same pin with SOT.

17.4.3 LIN-UART Serial Status Register (SSR)

The LIN-UART serial status register (SSR) is used to check the status of transmission, reception and error, and to enable and disable interrupts.

■ **LIN-UART Serial Status Register (SSR)**

Figure 17.4-4 LIN-UART Serial Status Register (SSR)

Address 0052H	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	00001000B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W	

TIE	Transmit interrupt request enable bit
0	Disables transmit interrupts.
1	Enables transmit interrupts.

RIE	Receive interrupt request enable bit
0	Disables receive interrupts.
1	Enables receive interrupts.

BDS	Transfer direction select bit
0	LSB-first (transfer from the least significant bit)
1	MSB-first (transfer from the most significant bit)

TDRE	Transmit data empty flag bit
0	Transmit data register (TDR) has data.
1	Transmit data register (TDR) is empty.

RDRF	Receive data full flag bit
0	Receive data register (RDR) is empty.
1	Receive data register (RDR) has data.

FRE	Framing error flag bit
0	No framing error.
1	Framing error occurs.

ORE	Overrun error flag bit
0	No overrun error.
1	Overrun error occurs.

PE	Parity error flag bit
0	No parity error.
1	Parity error occurs.

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

Table 17.4-3 Functions of Bits in Serial Status Register (SSR)

Bit name		Function
bit7	PE: Parity error flag bit	This bit detects the parity error in received data. • This bit is set to "1" when a parity error occurs during reception with PE = 1, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • When both the PE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the receive data register (RDR) is invalid.
bit6	ORE: Overrun error flag bit	This bit detects the overrun error in received data. • This bit is set to "1" when an overrun occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • When both the ORE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the receive data register (RDR) is invalid.
bit5	FRE: Framing error flag bit	This bit detects the framing error in received data. • This bit is set to "1" when a framing error occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR). • When both the FRE bit and the RIE bit are "1", a receive interrupt request is output. • When this flag is set, the data in the LIN-UART receive data register (RDR) is invalid.
bit4	RDRF: Receive data full flag bit	This flag shows the status of the LIN-UART receive data register (RDR). • This bit is set to "1" when received data is loaded into RDR, and cleared to "0" by reading the receive data register (RDR). • When both the RDRF bit and the RIE bit are "1", a receive interrupt request is output.
bit3	TDRE: Transmit data empty flag bit	This flag shows the status of the LIN-UART transmit data register (TDR). • This bit is set to "0" by writing the transmit data to TDR, and indicates that the TDR has valid data. When data is loaded into the transmit shift register and data transfer starts, this bit is set to "1", indicating that the TDR does not have valid data. • When both the TDRE bit and the TIE bit are "1", a transmit interrupt request is output. • When the TDRE bit is "1", setting the LBR bit in the LIN-UART extended communication control register (ECCR) to "1" changes the TDRE bit to "0". After the LIN synch break is generated, the TDRE bit returns to "1". Note:　The initial value of TDRE is "1".
bit2	BDS: Transfer direction select bit	This bit specifies whether the transfer of serial data starts from the least significant bit (LSB-first, BDS = 0) or from the most significant bit (MSB-first, BDS = 1). Note:　When data is written to or read from the serial data register, the data on the upper side and that on the lower side are swapped. Therefore, if the BDS bit is modified after data is written to the RDR register, the data in the RDR register becomes invalid. 　　　　In operating mode 3 (LIN), the BDS bit is fixed at "0".
bit1	RIE: Receive interrupt request enable bit	This bit enables or disables the receive interrupt request output to the interrupt controller. When both the RIE bit and the receive data flag bit (RDRF) are "1", or when one or more error flag bits (PE, ORE, FRE) is "1", a receive interrupt request is output.
bit0	TIE: Transmit interrupt request enable bit	This bit enables or disables the transmit interrupt request output to the interrupt controller. When both the TIE bit and the TDRE bit are "1", a transmit interrupt request is output.

17.4.4 LIN-UART Receive Data Register/LIN-UART Transmit Data Register (RDR/TDR)

The LIN-UART receive data register and the LIN-UART transmit data register are located at the same address. If read, they function as the receive data register; if written, they function as the transmit data register.

■ LIN-UART Receive Data Register (RDR)

Figure 17.4-5 shows the bit configuration of LIN-UART receive data register/LIN-UART transmit data register.

Figure 17.4-5 LIN-UART Receive Data Register/LIN-UART Transmit Data Register (RDR/TDR)
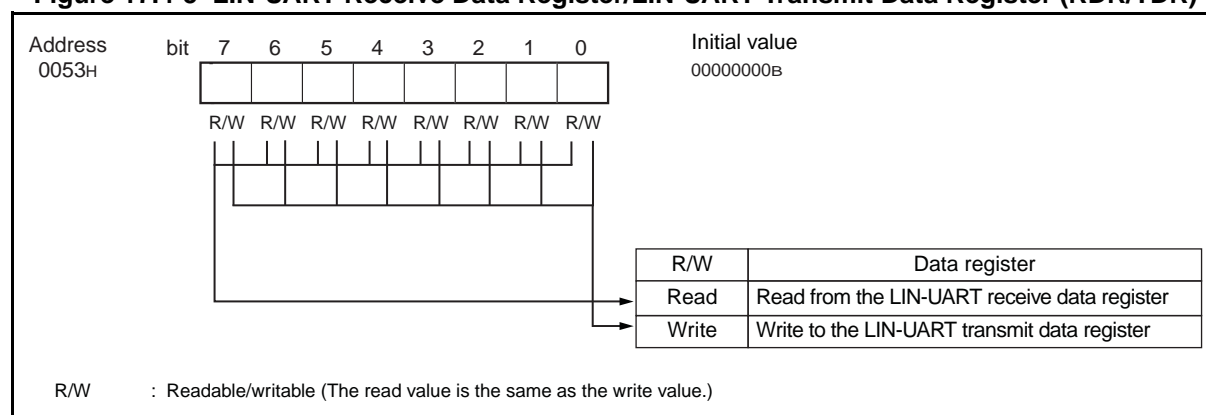
The LIN-UART receive data register (RDR) is the data buffer register for serial data reception.

Serial input data signals transmitted to the serial input pin (SIN pin) are converted by the shift register, and the converted data is stored in the LIN-UART receive data register (RDR).

If the data length is 7 bits, the MSB (RDR:D7) is "0".

The receive data full flag bit (SSR:RDRF) is set to "1" when received data is stored in the LIN-UART receive data register (RDR). If the receive interrupt has been enabled (SSR:RIE = 1), a receive interrupt request is made.

Read the LIN-UART receive data register (RDR) with the receive data full flag bit (SSR:RDRF) being "1". The receive data full flag bit (SSR:RDRF) is automatically cleared to "0" if the LIN-UART receive data register (RDR) is read. In addition, the receive interrupt is cleared when the receive interrupt has been enabled and no errors occur.

When a reception error occurs (any of SSR:PE, ORE, or FRE is "1"), the data in the LIN-UART receive data register (RDR) becomes invalid.

■ **LIN-UART Transmit Data Register (TDR)**

The LIN-UART transmit data register (TDR) is the data buffer register for serial data transmission.

If the data to be transmitted is written to the LIN-UART transmit data register (TDR) when transmission has been enabled (SCR:TXE = 1), the transmit data is transferred to the transmit shift register to convert to serial data, and the serial data is output from the serial data output pin (SOT pin).

If the data length is 7 bits, the data in the MSB (TDR:D7) is invalid.

The transmit data empty flag (SSR:TDRE) is cleared to "0" when transmit data is written to the LIN-UART transmit data register (TDR).

The transmit data empty flag (SSR:TDRE) is set to "1" after the data is transferred to the transmit shift register and data transmission starts.

If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written to TDR. If the transmit interrupt has been enabled, a transmit interrupt is generated. Write the next transmit data to TDR after a transmit interrupt or when the transmit data empty flag (SSR:TDRE) is "1".

Note:

The LIN-UART transmit data register is a write-only register; the receive data register is a read-only register. Since both registers are located at the same address, the write value and the read value are different. Thus, the read-modify-write (RMW) type of instruction, such as the INC instruction and the DEC instruction, cannot be used.

17.4.5 LIN-UART Extended Status Control Register (ESCR)

The LIN-UART extended status control register (ESCR) has the settings for enabling/disabling LIN synch break interrupt, LIN synch break length selection, LIN synch break detection, direct access to the SIN and SOT pins, continuous clock output in LIN-UART synchronous clock mode and sampling clock edge.

■ LIN-UART Extended Status Control Register (ESCR)

Figure 17.4-6 shows the bit configuration of the LIN-UART extended status control register (ESCR). Table 17.4-4 lists the function of each bit.

Figure 17.4-6 LIN-UART Extended Status Control Register (ESCR)

SCES	Sampling clock edge select bit (mode 2)
0	Sampling with rising clock edge (normal)
1	Sampling with falling clock edge (inverted clock)

CCO	Continuous clock output enable bit (mode 2)
0	Disables continuous clock output
1	Enables continuous clock output

SIOP	Serial I/O pin direct access bit	
	Write (SOPE = 1)	Read
0	Fixes SOT pin at "0"	Reads the value of SIN pin
1	Fixes SOT pin at "1"	

SOPE	Serial output pin direct access enable bit
0	Disables serial output pin direct access
1	Enables serial output pin direct access

LBL0	LBL1	LIN synch break length select bits
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

LBD	LIN synch break detection flag bit	
	Write	Read
0	LIN synch break detection flag clear	No LIN synch break detection
1	No effect	With LIN synch break detection

LBIE	LIN synch break detection interrupt enable bit
0	Disables LIN synch break detection interrupt
1	Enables LIN synch break detection interrupt

R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different from the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
▨ : Initial value

Table 17.4-4 Functions of Bits in LIN-UART Extended Status Control Register (ESCR)

Bit name		Function
bit7	LBIE: LIN synch break detection interrupt enable bit	This bit enables or disables LIN synch break detection interrupts. An interrupt is generated when the LIN synch break detection flag (LBD) is "1" and the interrupt is enabled (LBIE = 1). This bit is fixed at "0" in operating mode 1 and operating mode 2.
bit6	LBD: LIN synch break detection flag bit	This bit detects the LIN synch break. This bit is set to "1" when a LIN synch break is detected in operating mode 3 (the serial input is "0" when bit width is 11 bits or more). If "0" is written to the LBD bit, the LBD bit and the interrupt are cleared. Although the bit always returns "1" if read by the read-modify-write (RMW) type of instruction, this does not indicate that a LIN synch break has been detected. Note: To detect a LIN synch break, enable the LIN synch break detection interrupt (LBIE = 1), and then disable the reception (SCR:RXE = 0).
bit5, bit4	LBL1/LBL0: LIN synch break length select bits	These bits specify the bit length for the LIN synch break generation time. The LIN synch break length for reception is always 11 bits.
bit3	SOPE: Serial output pin direct access enable bit*	This bit enables or disables direct writing to the SOT pin. Setting this bit to "1" when serial data output has been enabled (SMR:SOE = 1) enables direct writing to the SOT pin.*
bit2	SIOP: Serial I/O pin direct access bit*	This bit controls direct access to the serial I/O pin. The SIOP bit always returns the value of the SIN pin if read by a normal read instruction. If direct access to the serial output pin is enabled (SOPE = 1), the value written to this bit is reflected in the SOT pin.* Note: When the bit manipulation instruction is used, the SIOP bit returns the bit value of the SOT pin in the read cycle.
bit1	CCO: Continuous clock output enable bit	This bit enables or disables continuous serial clock output from the SCK pin. In operating mode 2 (synchronous) in which the serial clock transmission side is selected, setting the CCO bit to "1" enables the continuous serial clock output from the SCK pin when the SCK pin is used as an clock output pin. Note: When the CCO bit is "1", set the SSM bit in the ECCR register to "1".
bit0	SCES: Sampling clock edge select bit	This bit selects a sampling edge. In operating mode 2 (synchronous) in which the serial clock reception side is selected, setting the SCES bit to "1" switches the sampling edge from the rising edge to the falling edge. In operating mode 2 (synchronous) in which the serial clock transmission side is selected (ECCR:MS = 0), when the SCK pin is used as an clock output pin, the internal serial clock signal and the output clock signal are inverted. In operating mode 0/1/3, set this bit to "0".

*: Interaction between SOPE and SIOP

SOPE	SIOP	Write to SIOP	Read from SIOP
0	R/W	No effect (however, the write value is retained)	Return the SIN value
1	R/W	Write "0" or "1" to SOT	Return the SIN value
1	RMW	Read the SOT value, write "0" or "1"	

17.4.6 LIN-UART Extended Communication Control Register (ECCR)

The LIN-UART extended communication control register (ECCR) is used for the bus idle detection, the synchronous clock setting, and the LIN synch break generation.

■ LIN-UART Extended Communication Control Register (ECCR)

Figure 17.4-7 shows the bit configuration of the LIN-UART extended communication control register (ECCR). Table 17.4-5 lists the function of each bit.

Figure 17.4-7 LIN-UART Extended Communication Control Register (ECCR)

TBI*	Transmit bus idle detection flag bit
0	Transmission in progress
1	No transmission

RBI*	Receive bus idle detection flag bit
0	Reception in progress
1	No reception

Reserved bit
The read value is indeterminate. Always set this bit to "0".

SSM	Start/stop bits mode enable bit (mode 2)
0	No start/stop bits
1	Start/stop bits available

SCDE	Serial clock delay enable bit (mode 2)
0	Disables clock delay
1	Enables clock delay

MS	Serial clock transmission/reception side select bit (mode 2)
0	Transmission side (serial clock generation)
1	Reception side (external serial clock reception)

LBR	LIN synch break generation bit (mode 3)	
	Write	Read
0	No effect	"0" is always read.
1	LIN synch break generation	

Reserved bit
The read value is indeterminate. Always set this bit to "0".

R/W   : Readable/writable (The read value is the same as the write value.)
R/WX   : Read only (Readable. Writing a value to it has no effect on operation.)
R0,W   : Write only (Writable. The read value is "0".)
RX,W0   : The read value is indeterminate; the write value is "0".
X   : Indeterminate
▢   : Initial value
*: This bit is not used when SSM=0 in operating mode 2.

Table 17.4-5 Functions of Bits in LIN-UART Extended Communication Control Register (ECCR)

Bit name		Function
bit7	Reserved bit	The read value is indeterminate. Always set this bit to "0".
bit6	LBR: LIN synch break generation bit	In operating mode 3, if this bit is set to "1", a LIN synch break whose length is specified in the LBL0/LBL1 bit in the ESCR register is generated. In operating mode 0/1/2, set this bit to "0".
bit5	MS: Serial clock transmission/reception side select bit	This bit selects the transmission side/reception side of the serial clock in operating mode 2. If the transmission side (MS = 0) is selected, the LIN-UART generates a synchronous clock. If the reception side (MS = 1) is selected, the LIN-UART receives an external serial clock. In mode 0/1/3, this bit is fixed at "0". Modify this bit only when the SCR:TXE bit is "0". Note: When the reception side is selected, the external clock must be selected as the clock source and the external clock and the external clock input must be enabled (SMR:SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	In operating mode 2 in which the serial clock transmission side is selected, if the SCDE bit is set to "1", a delayed serial clock as shown in Figure 17.7-5 is output. The function of outputting delayed serial clock can be used in the Serial Peripheral Interface (SPI). This bit is fixed at "0" in operating mode 0/1/3.
bit3	SSM: Start/stop bits mode enable bit	In operating mode 2, if this bit is set to "1", the start/stop bits are added to the synchronous data format. In operating mode 0/1/3, this bit is fixed at "0".
bit2	Reserved bit	The read value is indeterminate. Always set this bit to "0".
bit1	RBI: Receive bus idle detection flag bit	If the SIN pin is at "H" level and no reception is performed, this bit is "1". Do not use this bit when SSM = 0 in operating mode 2.
bit0	TBI: Transmit bus idle detection flag bit	If there is no transmission on the SOT pin, this bit is "1". Do not use this bit when SSM = 0 in operating mode 2.

17.4.7 LIN-UART Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) set the division ratio of the serial clock. In addition, the count value in the transmit reload counter is read from this generator.

■ LIN-UART Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 17.4-8 shows the bit configuration of LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 17.4-8 LIN-UART Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
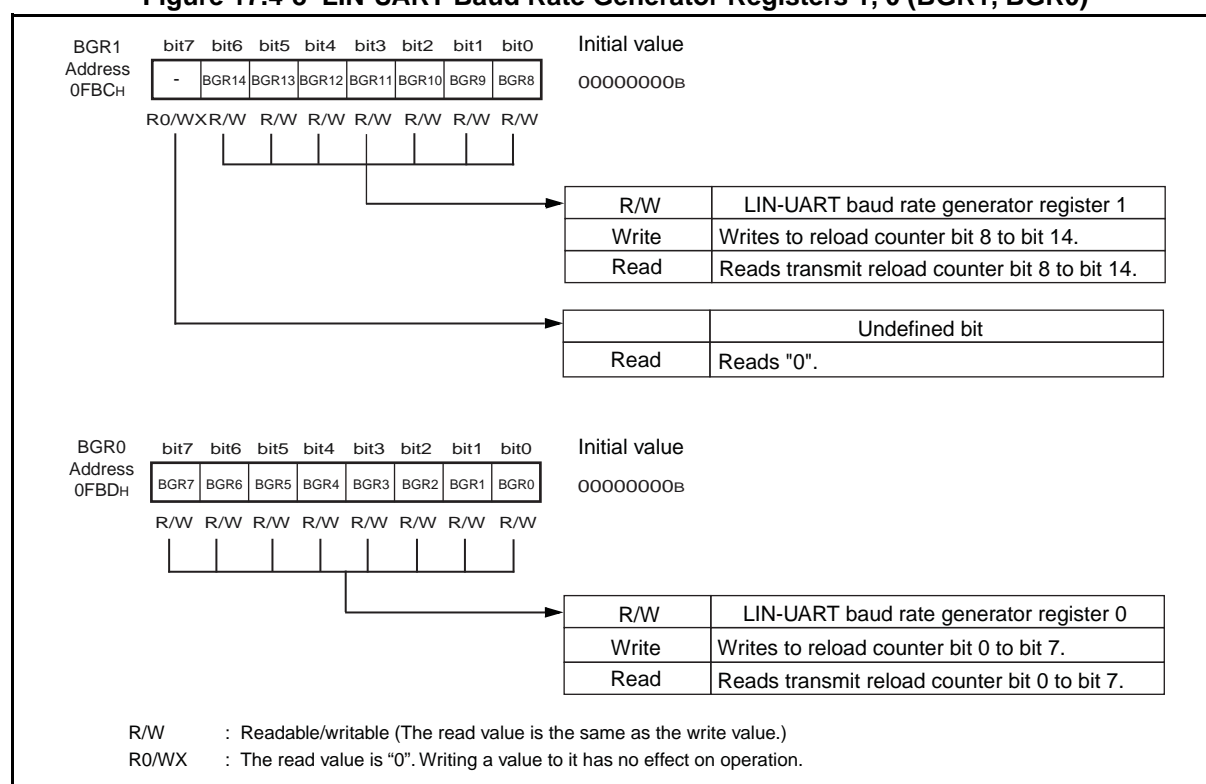
The LIN-UART baud rate generator registers set the division ratio of the serial clock.

BGR1 corresponds to the upper bits and BGR0 to the lower bits. The reload value of the counter can be written to and the transmit reload counter value can be read from BGR1 and BRG0. In addition, BGR1 and BGR0 can be accessed by byte access and word access.

Writing a reload value to the LIN-UART baud rate generator registers causes the reload counter to start counting.

Note:

Write to this register only when the LIN-UART stops.

MB95330H Series

17.5 LIN-UART Interrupts

The LIN-UART has receive interrupts and transmit interrupts, which are generated by the following sources. An interrupt number and an interrupt vector are assigned to each interrupt. In addition, it has a LIN synch field edge detection interrupt function using the 8/16-bit composite timer interrupt.

- **Receive interrupt**

  A receive interrupt occurs when received data is set in the LIN-UART receive data register (RDR), or when a receive error occurs, or when a LIN synch break is detected.

- **Transmit interrupt**

  A transmit interrupt occurs when transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register, and data transmission starts.

■ Receive Interrupt

Table 17.5-1 shows the control bits and interrupt sources of receive interrupts.

Table 17.5-1 Interrupt Control Bits and Interrupt Sources of Receive Interrupts

Interrupt request flag bit	Flag register	Operating mode 0	1	2	3	Interrupt source	Interrupt source enable bit	Interrupt request flag clear
RDRF	SSR	○	○	○	○	Writing received data to RDR	SSR:RIE	Read received data
ORE	SSR	○	○	○	○	Overrun error		Write "1" to receive error flag clear bit (SCR:CRE)
FRE	SSR	○	○	△	○	Framing error		
PE	SSR	○	×	△	×	Parity error		
LBD	ESCR	×	×	×	○	LIN synch break detection	ESCR:LBIE	Write "0" to ESCR:LBD

○ : Bit to be used
× : Unused bit
△ : Usable only when ECCR:SSM = 1

● Receive interrupts

If one of the following operations occurs in reception mode, the bit in the LIN-UART serial status register (SSR) corresponding to that operation is set to "1".

Data reception completed

Received data is transferred from the LIN-UART serial input shift register to the LIN-UART receive data register (RDR) (RDRF = 1).

Overrun error

With RDRF = 1, the next serial data is received while the CPU has not read the RDR register. (ORE = 1).

Framing error

A stop bit reception error occurs (FRE = 1).

Parity error

A parity detection error occurs (PE = 1).

A receive interrupt request is made if the receive interrupt has been enabled (SSR:RIE = 1) when one of the above flag bits is "1".

RDRF flag is automatically cleared to "0" if the LIN-UART receive data register (RDR) is read. All of the error flags are cleared to "0" if "1" is written to the receive error flag clear bit (CRE) in the LIN-UART serial control register (SCR).

Note:

The CRE bit is write-only, and keeps "1" for one clock cycle after "1" is written to the bit.

● LIN synch break interrupts

In operating mode 3, the LIN synch break interrupt functions when the LIN-UART performs LIN slave operation.

The LIN synch break detection flag bit (LBD) in the LIN-UART extended status control register (ESCR) is set to "1" when the internal data bus (serial input) is "0" for 11 bits or longer. The LIN synch break interrupt and the LBD flag are cleared by writing "0" to the LBD flag. The LBD flag must be cleared before the 8/16-bit composite timer interrupt is generated within the LIN synch field.

To detect a LIN synch break, the reception must be disabled (SCR:RXE = 0).

■ Transmit Interrupts

Table 17.5-2 shows the control bit and interrupt source of the transmit interrupt.

Table 17.5-2 Interrupt Control Bit and Interrupt Source of Transmit Interrupt

Interrupt request flag bit	Flag register	Operating mode				Interrupt source	Interrupt source enable bit	Interrupt request flag clear
		0	1	2	3			
TDRE	SSR	○	○	○	○	Transmit register is empty	SSR:TIE	Write transmit data

○: Bit to be used

● Transmit interrupts

The transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is set to "1" when the transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register, and data transmission starts. In this case, if the transmit interrupt has been enabled (SSR:TIE = 1), a transmit interrupt request is made.

Note:

Since the initial value of TDRE is "1" after a hardware reset/software reset, if the TIE bit is set to "1" after a hardware reset/software reset, an interrupt is generated immediately. The TDRE is cleared only by writing data to the LIN-UART transmit data register (TDR).

MB95330H Series

■ LIN Synch Field Edge Detection Interrupt (8/16-bit Composite Timer Interrupt)

Table 17.5-3 shows the control bits and interrupt sources of the LIN synch field edge detection interrupt.

Table 17.5-3 Interrupt Control Bits and Interrupt Sources of LIN Synch Field Edge Detection Interrupt

Interrupt request flag bit	Flag register	Operating mode				Interrupt source	Interrupt source enable bit	Interrupt request flag clear
		0	1	2	3			
IR	T00CR1	×	×	×	❍	First falling edge of the LIN synch field	T00CR1:IE	Write "0" to T00CR1:IR
IR	T00CR1	×	×	×	❍	Fifth falling edge of the LIN synch field		

❍ : Bit to be used
× : Unused bit

● LIN synch field edge detection interrupt (8/16-bit composite timer interrupt)

In operating mode 3, the LIN synch field edge detection interrupt functions when the LIN-UART performs LIN slave operation.

After a LIN synch break is detected, the internal signal (LSYN) is set to "1" at the first falling edge of the LIN synch field, and set to "0" after the fifth falling edge. Between both falling edges, an 8/16-bit composite timer interrupt is generated, provided that the 8/16-bit composite timer has been configured to receive internal signals and detect rising edges and falling edges and the 8/16-bit composite interrupt has been enabled.

The difference in the count values detected by the 8/16-bit composite timer (See Figure 17.5-1) is equivalent to eight bits of the master serial clock. A new baud rate can be calculated from this value. After set, a new baud rate becomes effective from the falling edge detected at the next start bit set.

Figure 17.5-1 Baud Rate Calculation by 8/16-bit Multi-Function Timer

Difference in count values = Capture value 2 - Capture Value 1
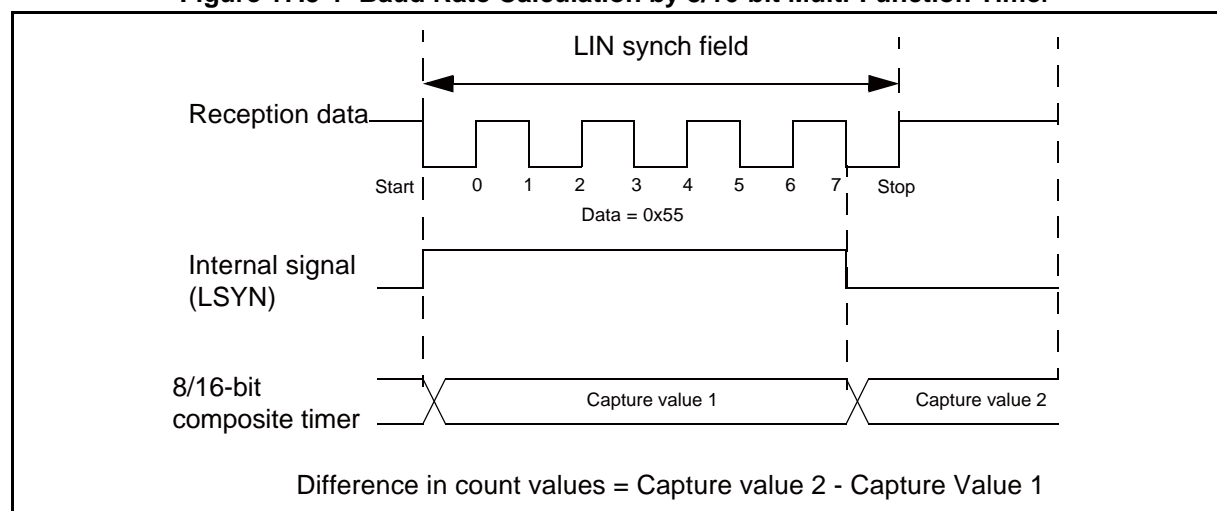
■ **Registers and Vector Table Addresses Related to LIN-UART Interrupts**

Table 17.5-4 Registers and Vector Table Addresses Related to LIN-UART Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
LIN-UART (reception)	IRQ07	ILR1	L07	FFEC$_H$	FFED$_H$
LIN-UART (transmission)	IRQ08	ILR2	L08	FFEA$_H$	FFEB$_H$

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

MB95330H Series

17.5.1 Timing of Receive Interrupt Generation and Flag Set

A receive interrupt is generated when reception is completed (SSR:RDRF) or when a reception error occurs (SSR:PE, ORE, FRE).

■ Timing of Receive Interrupt Generation and Flag Set

Received data is stored in the LIN-UART receive data register (RDR) when the first stop bit is detected in operating mode 0/1/2(SSM =1)/3, or when the last data bit is detected in operating mode 2 (SSM = 0). When reception is completed (SSR:RDRF = 1), or when a reception error occurs (SSR:PE, ORE, FRE = 1), an error flag corresponding to one of the events mentioned above is set. If the receive interrupt has been enabled (SSR:RIE = 1) when an error flag is set, a receive interrupt is generated.

Note:

In all operating modes, when a receive error occurs, data in the LIN-UART receive data register (RDR) becomes invalid.

Figure 17.5-2 shows the timing of reception and flag set.

Figure 17.5-2 Timing of Reception and Flag Set

* 1: The PE flag is always "0" in operating mode 1 and operating mode 3.
* 2: An overrun error occurs if the next data is transferred before received data is read (RDRF = 1).
ST: Start bit, SP: Stop bit, AD: Mode 1 (multiprocessor) address data select bit
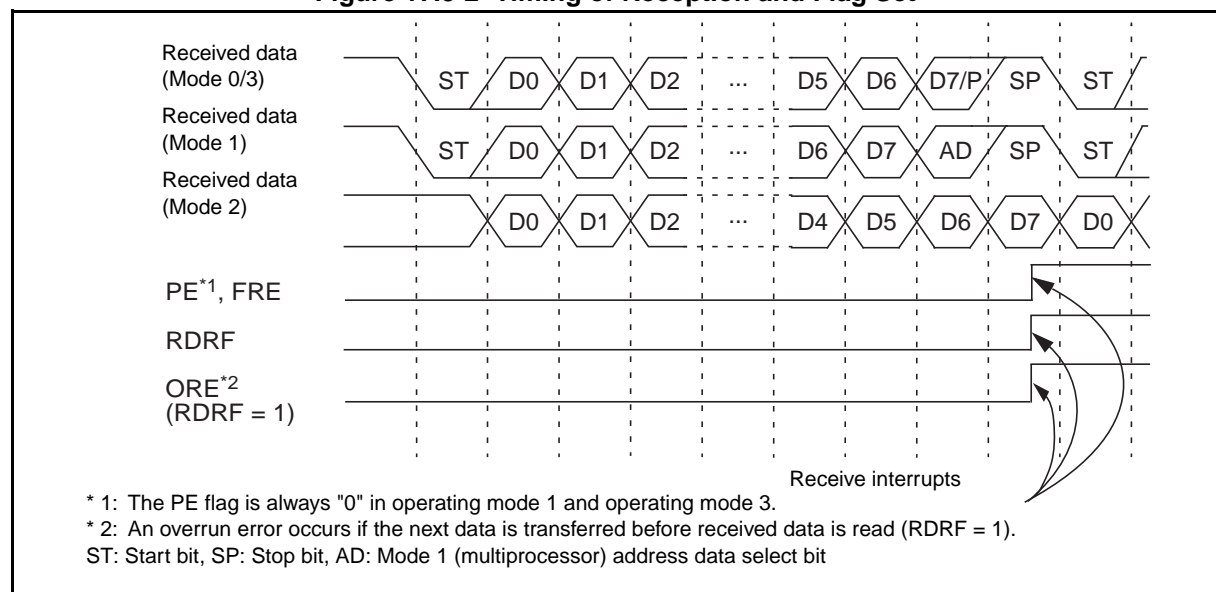
Note:

Figure 17.5-2 does not show all reception operations in mode 0. It only shows two examples of reception operations using different communication formats. One reception operation uses 7-bit data, a parity bit (parity bit = "even parity" or "odd parity") and one stop bit. The other uses 8-bit data, no parity bit and one stop bit.

Figure 17.5-3 ORE Flag Set Timing

MB95330H Series
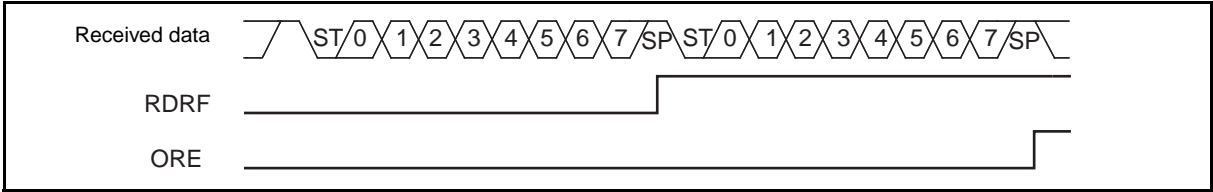
17.5.2 Timing of Transmit Interrupt Generation and Flag Set

A transmit interrupt is generated when transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register and then data transmission starts.

■ Timing of Transmit Interrupt Generation and Flag Set

When the data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register and the transmission of that data starts, the next data can be written to the TDR register (SSR:TDRE = 1). At the start of the data transmission, if the transmit interrupt has been enabled (SSR:TIE = 1), a transmit interrupt is generated.

The TDRE bit is a read-only bit, and is cleared to "0" only when data is written to the LIN-UART transmit data register (TDR).

Figure 17.5-4 shows the timing of transmission and flag set.

Figure 17.5-4 Timing of Transmission and Flag Set

ST: Start bit, D0 to D7: Data bits, P: Parity, SP: Stop bit
AD: Address data selection bit (mode 1)
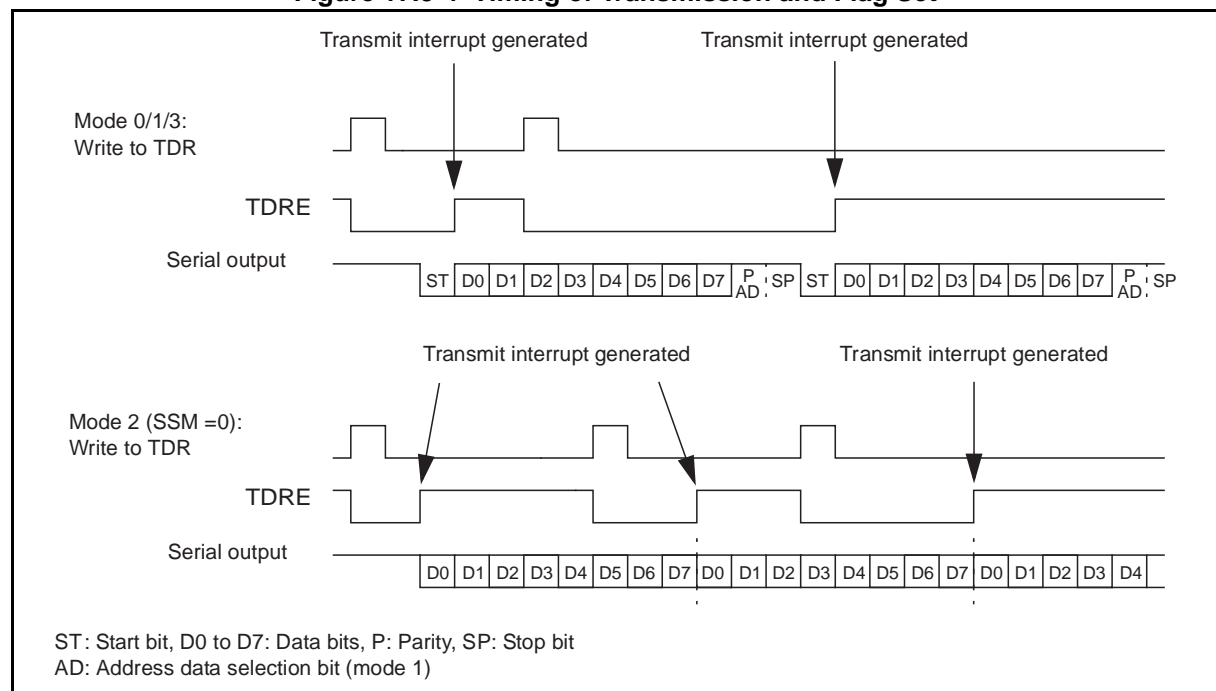
Note:

Figure 17.5-4 does not show all transmission operations in mode 0. It only shows an example of a transmission operation using 8-bit data, a parity bit ("even parity" or "odd parity") and one stop bit.

No parity bit is transmitted in mode 3, or in mode 2 with SSM = 0.

■ **Transmit Interrupt Request Generation Timing**

With the transmit interrupt having been enabled (SSR:TIE = 1), if the TDRE flag is set to "1", a transmit interrupt is generated.

Note:

Since the initial value of the TDRE bit is "1", a transmit interrupt is generated immediately after the transmit interrupt is enabled (SSR:TIE = 1). When deciding the timing of enabling the transmit interrupt, take into consideration that the TDRE bit can be cleared only by writing new data to the LIN-UART transmit data register (TDR).

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

MB95330H Series

17.6 LIN-UART Baud Rate

The input clock (transmit/receive clock source) of the LIN-UART can be selected from one of the following:
- **Input a machine clock to a baud rate generator (reload counter).**
- **Input an external clock to a baud rate generator (reload counter).**
- **Use an external clock (SCK pin input clock) directly.**

■ LIN-UART Baud Rate Selection

The baud rate can be selected from one of following three types. Figure 17.6-1 shows the baud rate selection circuit.

● Baud rate derived from the internal clock divided by the dedicated baud rate generator (reload counter)

There are two internal reload counters, corresponding to the transmit serial clock and the receive serial clock respectively. The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock by the value set in BGR1 and BGR0.

The baud rate is used in asynchronous mode and in synchronous mode (transmit side of the serial clock).

As for clock source settings, select the internal clock and use the baud generator clock (SMR:EXT = 0, OTO = 0).

● Baud rate derived from the external clock divided by the dedicated baud rate generator (reload counter)

The external clock is used as the clock source for the reload counter.

The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator registers 1, 0 (BGR0, BGR1).

The reload counter divides the external clock by the value set in BGR1 and BGR0.

The baud rate is used in asynchronous mode.

As for clock source settings, select the external clock and use the baud generator clock (SMR:EXT = 1, OTO = 0).

● Baud rate by the external clock (one-to-one mode)

The clock input from the clock input pin (SCK) of the LIN-UART is used as the baud rate (slave operation in operating mode 2 (synchronous) (ECCR:MS = 1)).

It is used in synchronous mode (serial clock reception side).

To set the clock source, select the external clock and its direct use (SMR:EXT = 1, OTO = 1).

Figure 17.6-1 LIN-UART Baud Rate Selection Circuit

MB95330H Series
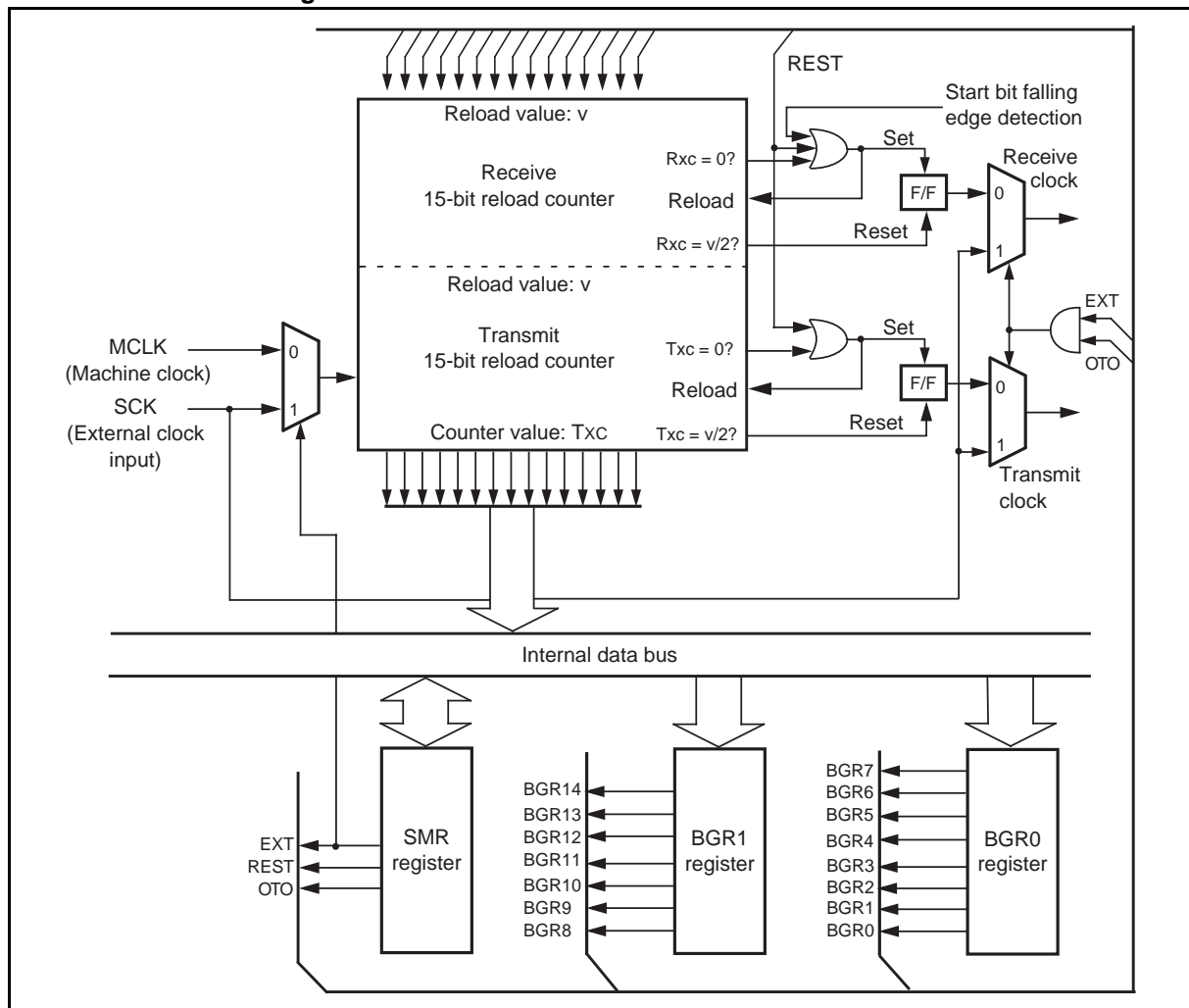
17.6.1 Baud Rate Setting

This section shows baud rate settings and the result of calculating the serial clock frequency.

■ **Baud Rate Calculation**

The two 15-bit reload counters are set by the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0).

The equation for the baud is shown below.

Reload value:

$$v = (\frac{MCLK}{b}) \text{ - } 1$$

v: Reload value, b: Baud rate, MCLK: Machine clock, or external clock frequency

Calculation example

Assuming that the machine clock is 10 MHz, the internal clock is used, and the baud rate is set to 19200 bps:

Reload value:

$$v = (\frac{10 \times 10^6}{19200}) \text{ -}1 = 519.83... \approx 520$$

Thus, the actual baud rate can be calculated as shown below.

$$b = \frac{MCLK}{(v+1)} = \frac{10 \times 10^6}{521} = 19193.8579$$

Note:

The reload counter stops if the reload value is set to "0". Therefore, set the smallest reload value to "1".

For transmission/reception in asynchronous mode, since five times of oversampling have to be done before the reception value is determined, the reload value must be set to at least "4".

■ **Reload Value and Baud Rate of Each Clock Speed**

Table 17.6-1 shows the reload value and baud rate of each clock speed.

Table 17.6-1 Reload Value and Baud Rate

Baud rate	8 MHz (MCLK)		10 MHz (MCLK)		16 MHz (MCLK)		16.25 MHz (MCLK)	
	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation
2M	-	-	4	0	7	0	-	-
1M	7	0	9	0	15	0	-	-
500000	15	0	19	0	31	0	-	-
400800	-	-	-	-	-	-	-	-
250000	31	0	39	0	63	0	64	0
230400	-	-	-	-	68	- 0.64	-	-
153600	51	- 0.16	64	- 0.16	103	- 0.16	105	0.19
125000	63	0	79	0	127	0	129	0
115200	68	- 0.64	86	0.22	138	0.08	140	- 0.04
76800	103	0.16	129	0.16	207	- 0.16	211	0.19
57600	138	0.08	173	0.22	277	0.08	281	- 0.04
38400	207	0.16	259	0.16	416	0.08	422	- 0,04
28800	277	0.08	346	- 0.06	555	0.08	563	- 0.04
19200	416	0.08	520	0.03	832	- 0.04	845	- 0.04
10417	767	< 0.01	959	< 0.01	1535	< 0.01	1559	< 0.01
9600	832	- 0.04	1041	0.03	1666	0.02	1692	0.02
7200	1110	< 0.01	1388	< 0.01	2221	< 0.01	2256	< 0.01
4800	1666	0.02	2082	- 0.02	3332	< 0.01	3384	< 0.01
2400	3332	< 0.01	4166	< 0.01	6666	< 0.01	6770	< 0.01
1200	6666	< 0.01	8334	< 0.01	13332	< 0.01	13541	< 0.01
600	13332	< 0.01	16666	< 0.01	26666	< 0.01	27082	< 0.01
300	26666	< 0.01	-	-	53332	< 0.01	54166	< 0.01

The unit of frequency deviation (dev.) is %. MCLK represents machine clock.

MB95330H Series

■ External Clock

The external clock is selected by writing "1" to the EXT bit in the LIN-UART serial mode register (SMR). In the baud rate generator, the external clock can be used in the same way as the internal clock.

When slave operation is used in operating mode 2 (synchronous), select the one-to-one external clock input mode (SMR:OTO = 1). In this mode, the external clock input to SCK is input directly to the LIN-UART serial clock.

Note:

The external clock signal is synchronized with the internal clock (MCLK: machine clock) in the LIN-UART. Therefore, if the external clock becomes not divisible because its cycle is faster than half the cycle of the internal clock, the external clock signal becomes unstable. For the value of the SCK clock, refer to the data sheet of the MB95330H Series.

■ **Operation of Dedicated Baud Rate Generator (Reload Counter)**

Figure 17.6-2 shows the operation of two reload counters using a reload value "832" as an example.

Figure 17.6-2 Operation of Dedicated Baud Rate Generator (Reload Counter)
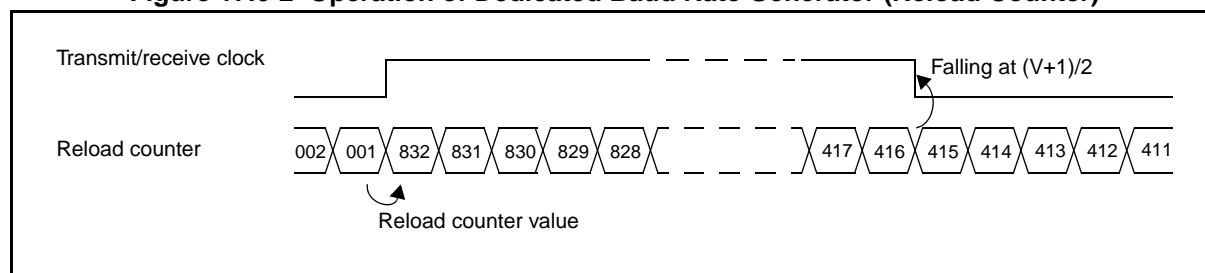
Note:

The falling edge of the serial clock signal is generated after the reload value divided by 2 [(V+1)/2] is counted.

MB95330H Series

17.6.2   Reload Counter

This block is a 15-bit reload counter functioning as a dedicated baud rate generator. It generates the transmit/receive clock from the external clock or internal clock.
The count value in the transmit reload counter can be read from the LIN-UART baud rate generator registers 1, 0 (BGR 1 and BGR 0).

■ Functions of Reload Counter

There are two types of reload counter, the transmit reload counter and the receive reload counter. The reload counter functions as a dedicated baud rate generator. It consists of a 15-bit register for a reload value and generates the transmit/receive clock from the external clock or internal clock. The count value in the transmit reload counter can be read from the LIN-UART baud rate generator registers 1, 0 (BGR 1 and BGR 0).

● Start of counting

Writing a reload value to the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0) causes the reload counter to start counting.

● Restart

The reload counter restarts under the following conditions.

For both transmit/receive reload counters

- LIN-UART programmable reset (SMR:UPCL bit)
- Programmable restart (SMR:REST bit)

For the receive reload counter

- Detection of a start bit falling edge in asynchronous mode

● Simple timer function

If the REST bit in the LIN-UART serial mode register (SMR) is set to "1", the two reload counters restart at the next clock cycle.

This function enables the transmit reload counter to be used as a simple timer.

Figure 17.6-3 shows an example of using this function (when the reload value is 100).

Figure 17.6-3 Example of Using a Simple Timer by Restarting the Reload Timer
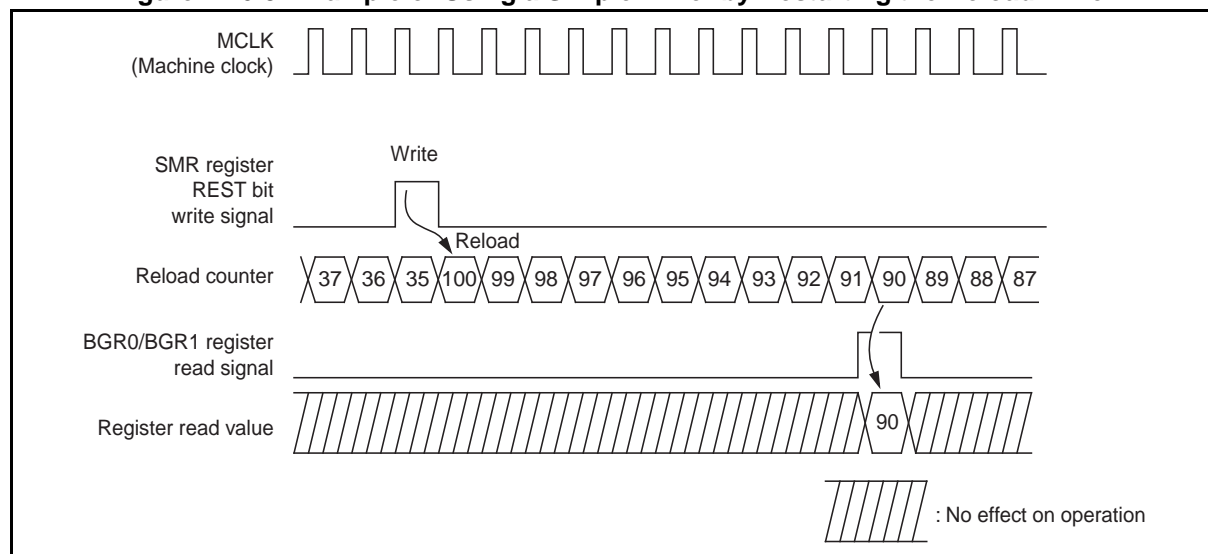
The number of machine clock cycles "cyc" after the restart in this example is obtained by the following equation.

$$cyc = v - c + 1 = 100 - 90 + 1 = 11$$

v: Reload value, c: Reload counter value

Note:

The transmit reload counter restarts also when the LIN-UART is reset by writing "1" to the SMR:UPCL bit.

Automatic restart (receive reload counter only)

The receive reload counter restarts when the start bit falling edge is detected in asynchronous mode. This automatic restart function is to synchronize the receive shift register with the received data.

● Clear counter

When a reset occurs, the reload values in the LIN-UART baud rate generator registers 1, 0 (BGR 1, BGR 0) and the reload counter are cleared to "00_H", and the reload counter stops.

Although the counter value is temporarily cleared to "00_H" by the LIN-UART reset (writing "1" to SMR:UPCL), the reload counter restarts since the reload value is kept.

If the restart setting is used (writing "1" to SMR:REST), the reload counter restarts without the counter value being cleared to "00_H".

MB95330H Series

17.7 Operations of LIN-UART and LIN-UART Setting Procedure Example

The LIN-UART performs bi-directional serial communication in operating mode 0/2, master/slave communication in operating mode 1, LIN master/slave communication in operating mode 3.

■ Operations of LIN-UART

● Operating mode

The LIN-UART has four operating modes (0 to 3), providing different connection methods between CPUs and different data transfer methods as shown in Table 17.7-1.

Table 17.7-1 LIN-UART Operating Modes

Operating mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multiprocessor mode	7 bits or 8 bits +1[*]	-	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	
3	LIN mode	8 bits	-	Asynchronous	1 bit	LSB first

- : Unavailable

[*] : "+1" is the address/data select bit (A/D) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operating modes.

Table 17.7-2 LIN-UART Operating Modes

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

Notes:

- In operating mode 1, a system connecting to a master/slave supports both master operations and slave operations.
- In operating mode 3, the communication format is fixed at "8-bit data, no parity bit, one stop bit, LSB-first".
- If the operating mode is changed, all transmission operations and reception operations are canceled, and the LIN-UART waits for the next transmission/reception.

■ Inter-CPU Connection Method

The external clock one-to-one connection (normal mode) and the master/slave connection (multiprocessor mode) can be selected as an inter-CPU connection method. In either method, CPUs must use the same data length, parity setting, synchronization type, etc. Select their operating modes as follows.

- One-to-one connection: Both CPUs must use the either operating mode 0 or operating mode 2. Select the operating mode 0 for asynchronous method or the operating mode 2 for synchronous method. In addition, in operating mode 2, set one CPU as the transmission side of serial clock and the other as the reception side of serial clock.

- Master/slave connection: Select operating mode 1. Use the CPU as a master/slave system.

■ Asynchronous/Synchronous Method

As for the asynchronous method, the receive clock is synchronized with the receive start bit falling edge. As for the synchronous method, the receive clock can be synchronized with the clock signal of the serial clock transmission side, or with the clock signal of the LIN-UART operating as the transmission side.

■ Signaling

NRZ (Non Return to Zero).

■ Enable Transmission/Reception

The LIN-UART uses the SCR:TXE bit and the SCR:RXE bit to control transmission and reception, respectively. Execute the following operations to disable transmission or reception.

- To disable reception while it is in progress: wait until reception ends, read the receive data register (RDR), then disable reception.

- To disable transmission while it is in progress: wait until transmission ends, then disable transmission.

■ Setting Procedure Example

Below is an example of procedure for setting the LIN-UART.

● Initial settings

1) Set the port input (DDR0).
2) Set the interrupt level (ILR1, ILR2).
3) Set the data format and enable transmission/reception (SCR).
4) Select the operating mode and the baud rate, and enable pin output (SMR).
5) Set the baud rate generators 1, 0 (BGR1,BGR0).

MB95330H Series

17.7.1 Operations in Asynchronous Mode (Operating Mode 0, 1)

When the LIN-UART is used in operating mode 0 (normal mode) or operating mode 1 (multiprocessor mode), the transfer method is asynchronous transfer.

■ Operations in Asynchronous Mode

● Transmit/receive data format

Transmit/receive data always begins with a start bit ("L" level), followed by a specified data bits length, and ends with at least one stop bit ("H" level).

The bit transfer direction (LSB-first or MSB-first) is determined by the BDS bit in the LIN-UART serial status register (SSR). When the parity bit is used, it is always placed between the last data bit and the first stop bit.

In operating mode 0, the data length can be 7 bits or 8 bits. The use of the parity can be selected. The stop bit length can also be selected from one and two.

In operating mode 1, the data length can be 7 bits or 8 bits. No parity is added while an address/data bit is added. The stop bit length can be selected from one and two.

Below is the equation for the bit length of a transmit/receive frame.

Length = 1 + d + p + s

(d = Number of data bits [7 or 8], p = parity [0 or 1],

s = Number of stop bits [1 or 2])

Figure 17.7-1 shows the transmit/receive data format in asynchronous mode (operating mode 0 or operating mode 1).

Figure 17.7-1 Transmit/Receive Data Format (Operating Mode 0, 1)

ST : Start bit
SP : Stop mode
P : Parity bit
AD: Address/data bit
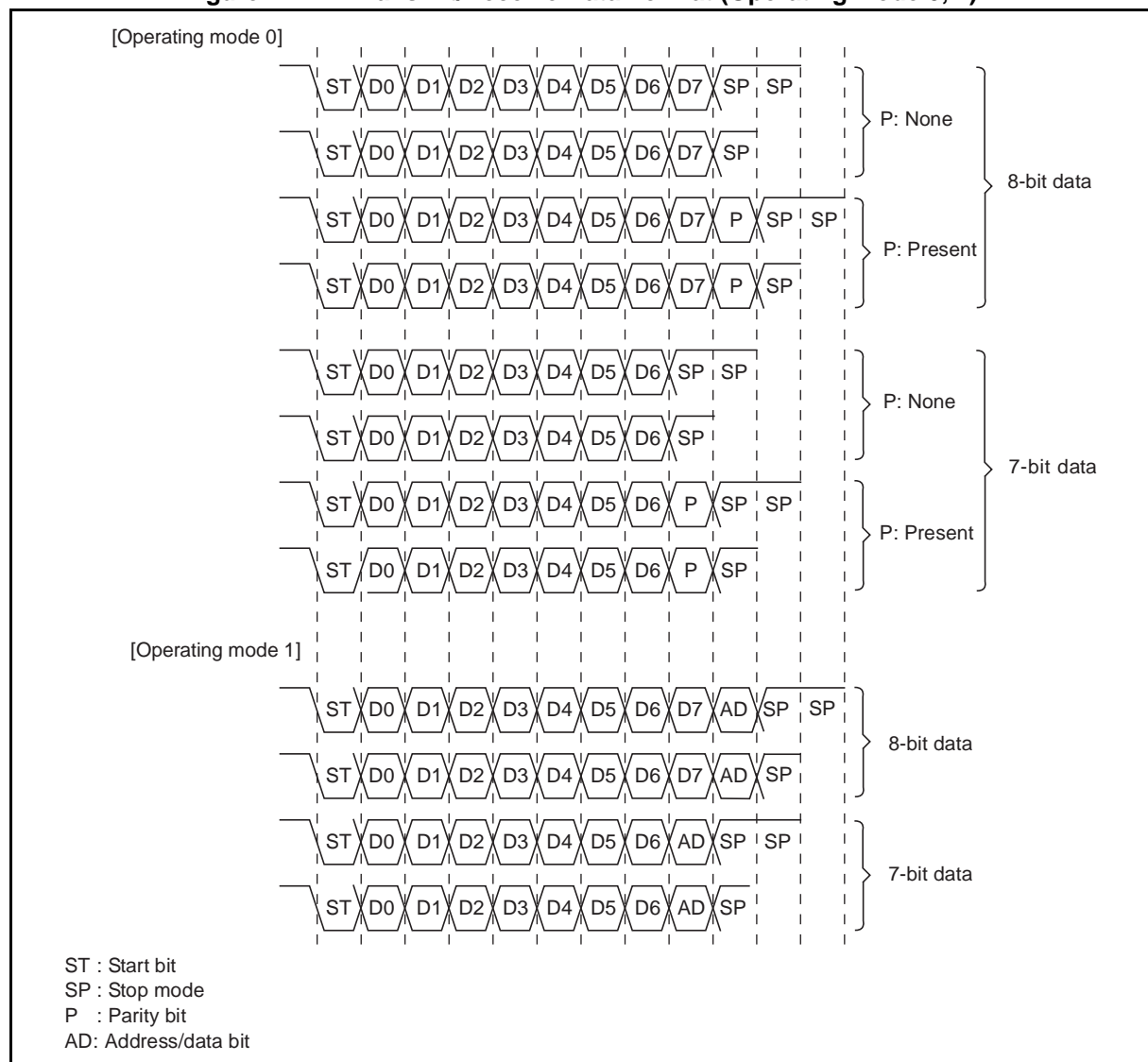
Note:

When the BDS bit in the LIN-UART serial status register (SSR) is set to "1" (MSB-first), the bits are processed in the following order: D7, D6, ... D1, D0 (P).

● Transmission

If the transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is "1", transmit data can be written to the LIN-UART transmit data register (TDR). Writing data sets the TDRE flag to "0". If transmission has been enabled (SCR:TXE = 1) when the TDRE flag is set to "0", the data written to TDR is written to the transmit shift register, and, in the next serial clock cycle, the transmission of the data is started from the start bit.

With the transmit interrupt having been enabled (TIE = 1), if transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmit shift register, the TDRE flag is set to "1" and an interrupt is generated.

When the data length is set to 7 bits (CL = 0), bit 7 in the TDR register becomes an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Since the initial value of the transmit data empty flag bit (SSR:TDRE) is "1", an interrupt is generated immediately when the transmit interrupt is enabled (SSR:TIE =1).

● Reception

The reception is performed when reception is enabled (SCR:RXE =1). When a start bit is detected, one frame data is received according to the data format defined in the LIN-UART serial control register (SCR). If an error occurs, an error flag (SSR:PE, ORE, FRE) is set. After the reception of one frame data ends, the received data is transferred from the receive shift register to the LIN-UART receive data register (RDR), and the receive data register full flag bit (SSR:RDRF) is set to "1". If the receive interrupt request has already been enabled (SSR:RIE = 1) at that time, a receive interrupt request is output.

To read the received data, first check the error flag status to ensure that reception has been executed normally, then read the data from the LIN-UART receive data register (RDR) if the reception is normal. If a reception error has occurred, perform error processing.

When the received data is read, the receive data register full flag bit (SSR:RDRF) is cleared.

When the data length is set to 7 bits (CL = 0), bit 7 in the TDR register becomes an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

Note:

Data in the LIN-UART receive data register (RDR) becomes valid, provided that the receive data register full flag bit (SSR:RDRF) is set to "1" and no error has occurred (SSR:PE, ORE, FRE=0).

● Input clock

Use the internal clock or the external clock. For the baud rate, select the baud rate generator (SMR:EXT = 0 or 1, OTO = 0).

● Stop bit and reception bus idle flag

For transmission, the number of stop bits can be selected from one and two. If two stop bits are selected, both stop bits are detected during reception.

When the first stop bit is detected, the receive data register full flag (SSR:RDRF) is set to "1". When no start bit is detected afterward, the receive bus idle flag (ECCR:RBI) is set to "1", indicating that no reception is executed.

● Error detection

In operating mode 0, the parity error, the overrun error and the frame error can be detected.

In operating mode 1, the overrun error and the frame error can be detected. However, the parity error cannot be detected.

● Parity

The addition (at transmission) of and the detection (during reception) of a parity bit can be set.

The parity enable bit (SCR:PEN) is used to select whether or not to use a parity; the parity select bit (SCR:P) is used to select the odd/even parity.

In operating mode 1, the parity cannot be used.

Figure 17.7-2 Transmission Data when Parity is Enabled
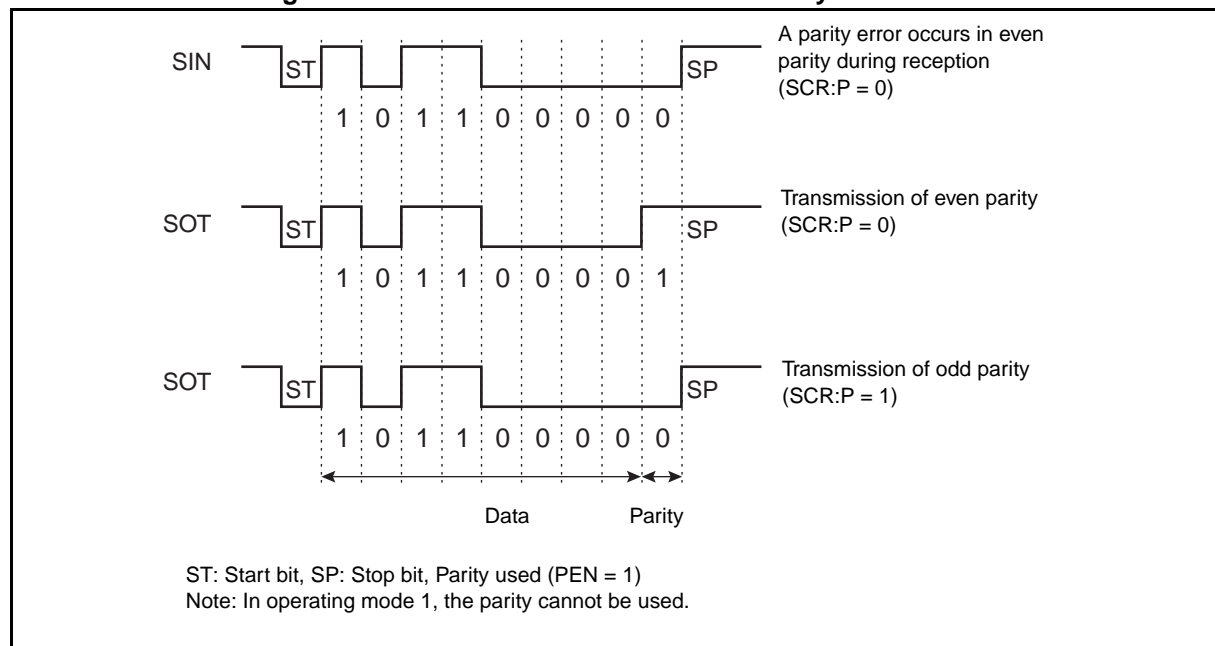
● Data signaling

NRZ data format.

● Data bit transfer method

The data bit transfer method can be LSB-first transfer or MSB-first transfer.

MB95330H Series

17.7.2 Operations in Synchronous Mode (Operating Mode 2)

When the LIN-UART is used in operating mode 2 (normal mode), the transfer method is clock-synchronous transfer.

■ Operations in Synchronous Mode (Operating Mode 2)

● Transmit/receive data format

In synchronous mode, 8-bit data is transmitted and received; the addition of the start bit and of the stop bit can be selected (ECCR:SSM). When the start/stop bits are added to the data format (ECCR:SSM = 1), the addition of the parity bit can also be selected (SCR:PEN).

Figure 17.7-3 shows the data format in synchronous mode (operating mode 2).

Figure 17.7-3 Transmit/Receive Data Format (Operating Mode 2)
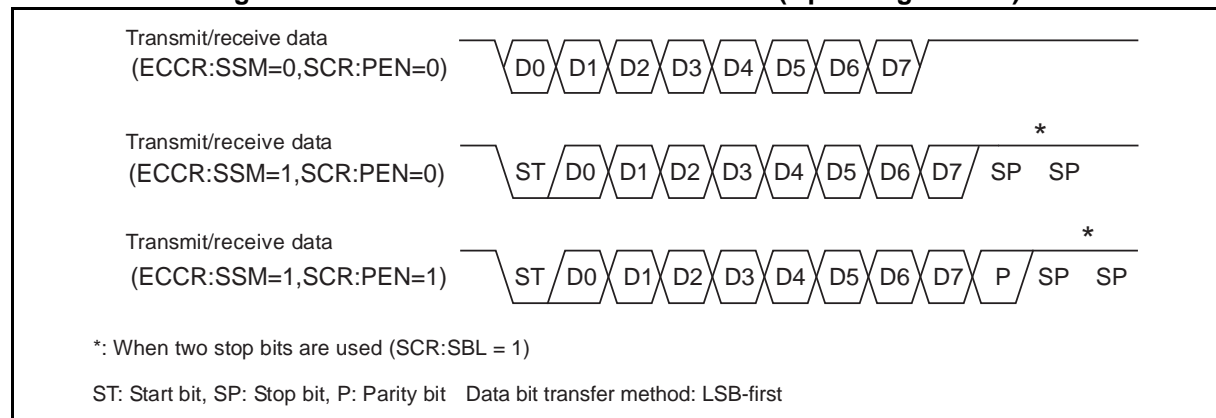
● Clock inversion function

When the SCES bit in the LIN-UART extended status control register (ESCR) is "1", the serial clock is inverted. In the case of serial clock reception side is selected, the LIN-UART samples data at the falling edge of the received serial clock. In the case of serial clock transmission side is selected, the mark level is set to "0" when the SCES bit is "1".

Figure 17.7-4 Transmission Data Format During Clock Inverted
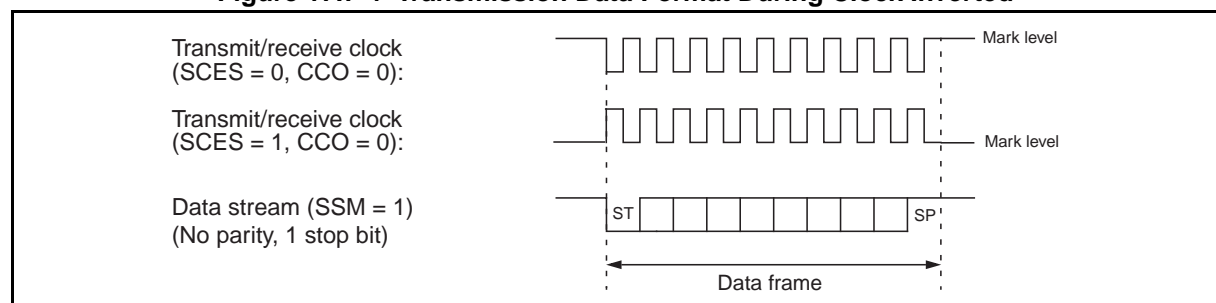
● Start/stop bits

When the SSM bit in the LIN-UART extended communication control register (ECCR) is "1", the start and stop bits are added to the data format as they are in asynchronous mode.

● Clock supply

In clock synchronous mode (normal), the number of transmit/receive data bits must be equal to the number of clock cycles. When the start/stop bits are enabled, the number of clock cycles must be equal to the sum of the transmit/receive data bits and the added start/stop bits.

With the serial clock transmission side having been selected (ECCR:MS = 0), when the serial clock output is enabled (SMR:SCKE = 1), a synchronous clock is automatically output during transmission/reception. When the serial clock reception side (ECCR:MS = 1) is selected or the serial clock output is disabled (SMR:SCKE = 0), clock cycles equal to the number of transmit/receive data bits must be supplied from an external clock pin.

The clock signal must be kept at the mark level ("H") if serial data is not related to transmission/reception.

● Clock delay

When the SCDE bit in the ECCR is set to "1", a delayed transmit clock is output as shown in Figure 17.7-5. This function is required when the device on the reception side samples data at the rising edge or falling edge of the serial clock.

Figure 17.7-5 Transmit Clock Delay (SCDE = 1)
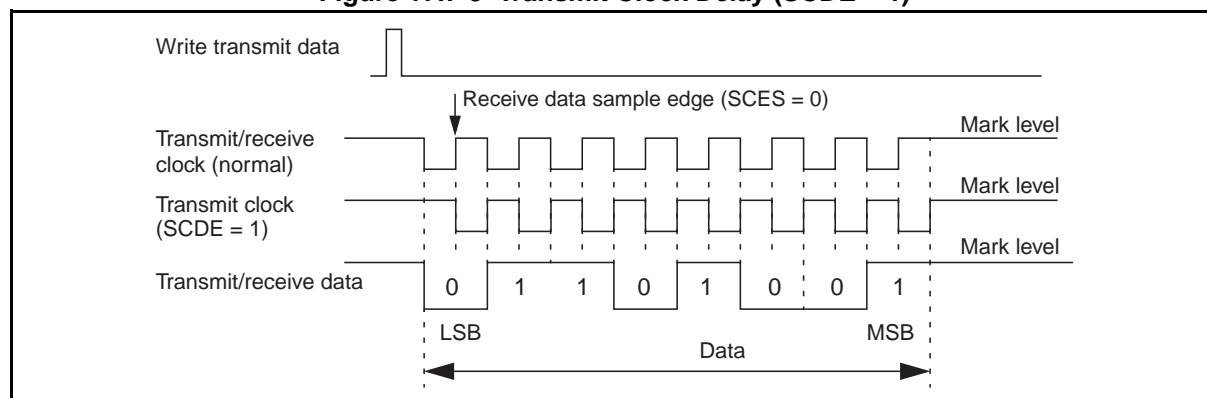
● Clock inversion

When the SCES bit in the LIN-UART extended status register (ESCR) is "1", the LIN-UART clock is inverted, and receive data is sampled at the falling edge of the LIN-UART clock. At that time, the value of the serial data must become valid at the edge of the LIN-UART clock.

MB95330H Series

● Continuous clock supply

When the CCO bit in the ESCR register is "1", the serial clock output from the SCK pin is continuously supplied on the serial clock transmission side. In this case, add the start bit and the stop bit to the data format (SSM = 1) in order to identify the beginning and end of the data frame. Figure 17.7-6 shows the operation of continuous clock supply (operating mode 2).

Figure 17.7-6 Continuous Clock Supply (Operating Mode 2)
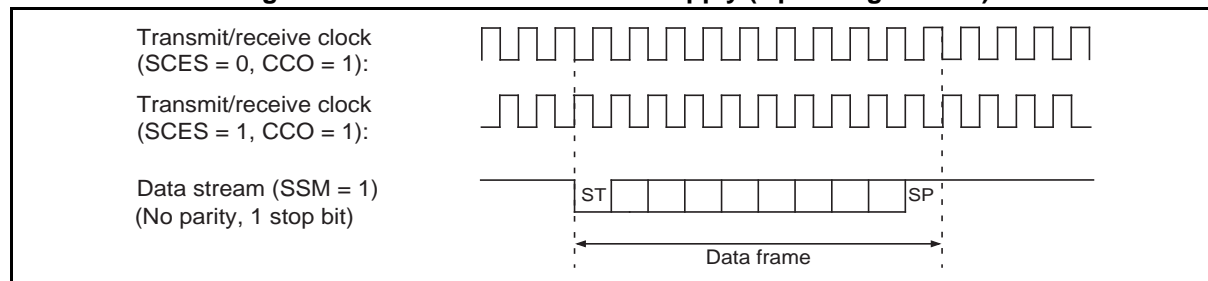
● Error detection

When the start bit and the stop bit are disabled (ECCR:SSM = 0), only overrun errors are to be detected.

● Communication settings for synchronous mode

To perform communications in synchronous mode, the following settings are required.

- LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0)
  Set the dedicated baud rate reload counter to a required value.
- LIN-UART serial mode register (SMR)
  MD1, MD0: "10_B" (Mode 2)

  SCKE : "1"– Uses the dedicated baud rate reload counter

  : "0"– Inputs an external clock

  SOE : "1"– Enables transmission/reception

  : "0"– Enables only reception

- LIN-UART serial control register (SCR)
  RXE, TXE: Set either bit to "1".

  AD : Since the address/data format selection function is not used, the value of this bit has no effect on operation.

  CL : Since the bit length is automatically set to 8 bits, the value of this bit has no effect on operation.

  CRE : "1" – Clears the error flag.

  - For SSM = 0:

    PEN, P, SBL: Since neither the parity bit nor the stop bit is used, the values of these three bits have no effect on operation.

  - For SSM = 1:

    PEN : "1": Adds/detects parity bit, "0": Not use parity bit

    P : "1": Odd parity, "0": Even parity

    SBL : "1": Stop bit length 2, "0": Stop bit length 1

- LIN-UART serial status register (SSR)

  BDS : "0"– LSB-first, "1"– MSB-first

  RIE : "1"– Enables receive interrupts, "0"– Disables receive interrupts

  TIE : "1"– Enables transmit interrupts, "0"– Disables transmit interrupts

- LIN-UART extended communication control register (ECCR)

  SSM : "0"– Not use start/stop bits (normal),

  "1"– Uses start/stop bits (extended function),

  MS : "0"– Serial clock transmission side (serial clock output),

  "1"– Serial clock reception side (inputs serial clock from the device on the serial clock transmission side)

Note:

To start communication, write data to the LIN-UART transmit data register (TDR).

To receive data only, disable the serial output (SMR:SOE = 0), and then write dummy data to the TDR register.

Enabling continuous clock output and the start/stop bits enables bi-directional communication as that in asynchronous mode.

MB95330H Series

17.7.3   Operations of LIN function (Operating Mode 3)

In operating mode 3, the LIN-UART works as the LIN master and the LIN slave. In operating mode 3, the communication format is set to 8-bit data, no parity, stop bit 1, LSB first.

■ Asynchronous LIN Mode Operation

● Operation as LIN master

In LIN mode, the master determines the baud rate for the entire bus, and the slave synchronizes with the master.

Writing "1" to the LBR bit in the LIN-UART extended communication control register (ECCR) outputs 13 bits to 16 bits at the "L" level from the SOT pin. These bits are the LIN synch break indicating the beginning of a LIN message.

The TDRE flag bit in the LIN-UART serial status register (SSR) is then set to "0". After the LIN synch break, the TDRE flag bit is set to "1" (initial value). If the TIE bit in SSR is "1" at this time, a transmit interrupt is output.

The length of the LIN synch break transmitted is set by the LBL 0/LBL1 bits in ESCR as shown in the following table.

Table 17.7-3 LIN Synch Break Length

LBL0	LBL1	Synch break length
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

A LIN synch field is transmitted as byte data 55_H following a LIN synch break. To prevent the generation of a transmit interrupt, 55_H can be written to the TDR after the LBR bit in ECCR is set to "1" even if the TDRE flag bit is "0".

● Operation as LIN slave

In LIN slave mode, the LIN-UART must synchronize with the baud rate of the master. The LIN-UART generates a receive interrupt when LIN break interrupt is enabled (LBIE = 1) even though reception has been disabled (RXE = 0). The LBD bit in ESCR is set to "1" as a receive interrupt is generated.

Writing "0" to the LBD bit clears the receive interrupt request flag.

The calculation of baud rate is illustrated below using the operation of the LIN-UART as an example. When the LIN-UART detects the first falling edge of the synch field, set the internal signal to be input to the 8/16-bit composite timer to "H", and then start the 8/16-bit composite timer. The internal signal becomes "L" at the fifth falling edge. The 8/16-bit composite timer must be set to the input capture mode. In addition, the 8/16-bit composite timer interrupt must be enabled and the 8/16-bit composite timer must be set to detect both edges. The time at which the input signal input to the 8/16-bit composite timer is eight times the baud rate.

The baud rate setting can be found by the following equations.

When the counter of the 8/16-bit composite timer does not overflow

$$\text{BGR value} = (b - a) / 8 - 1$$

When the counter of the 8/16-bit composite timer has overflowed

$$\text{BGR value} = (max + b - a) / 8 - 1$$

max: Maximum value of free-run timer

a: TII0 data register value after the first interrupt

b: TII0 data register value after the second interrupt

Note:

If the BGR value newly calculated based on the synch field in LIN slave mode as explained above has an error of ±15% or more, do not set the baud rate.

For the operations of the input capture function of the 8/16-bit composite timer, see Section 14.13 "Operation of Input Capture Function".

● LIN synch break detection interrupt and flag

The LIN break detection (LBD) flag in ESCR is set to "1" when the LIN synch break is detected in slave mode. When the LIN break interrupt is enabled (LBIE = 1), an interrupt is generated.

Figure 17.7-7 Timing of LIN Synch Break Detection and Flag Set
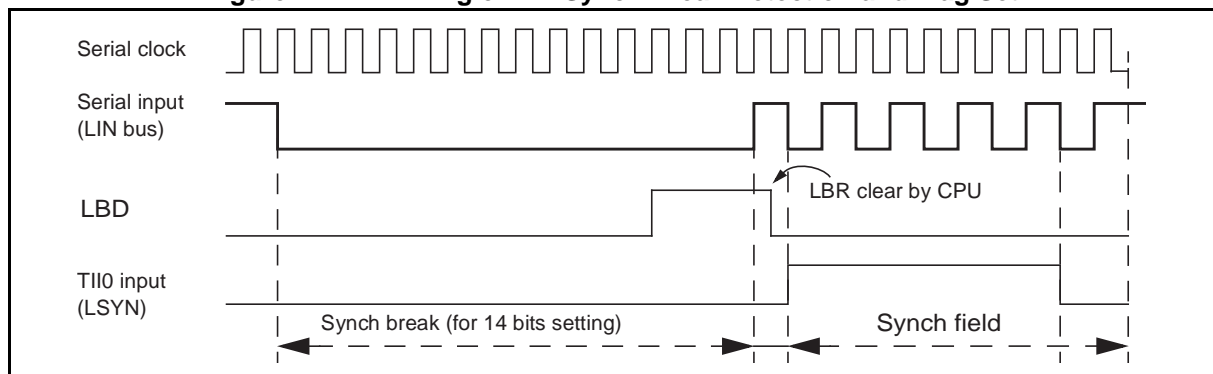
The above diagram shows the timing of the LIN synch break detection and flag.

Since the data framing error (FRE) flag bit in SSR generates a receive interrupt two bits earlier than a LIN break interrupt (if the following communication format is used: 8-bit data, no parity, one stop bit.), set the RXE to "0" when using the LIN break.

The LIN synch break detection functions only in operating mode 3.

Figure 17.7-8 shows the LIN-UART operation in LIN slave mode.

MB95330H Series

Figure 17.7-8 LIN-UART Operation in LIN Slave Mode
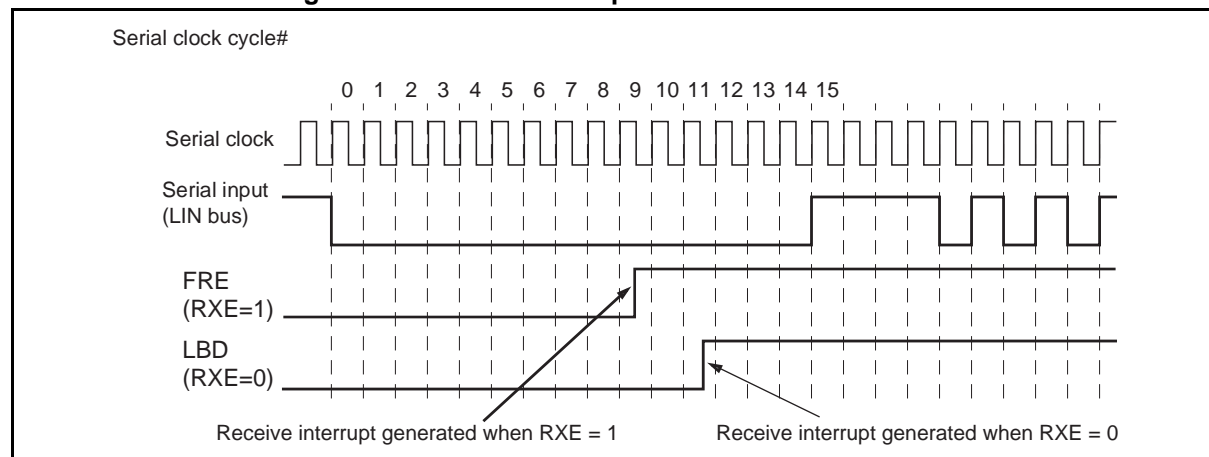
● LIN bus timing

Figure 17.7-9 LIN Bus Timing and LIN-UART Signals
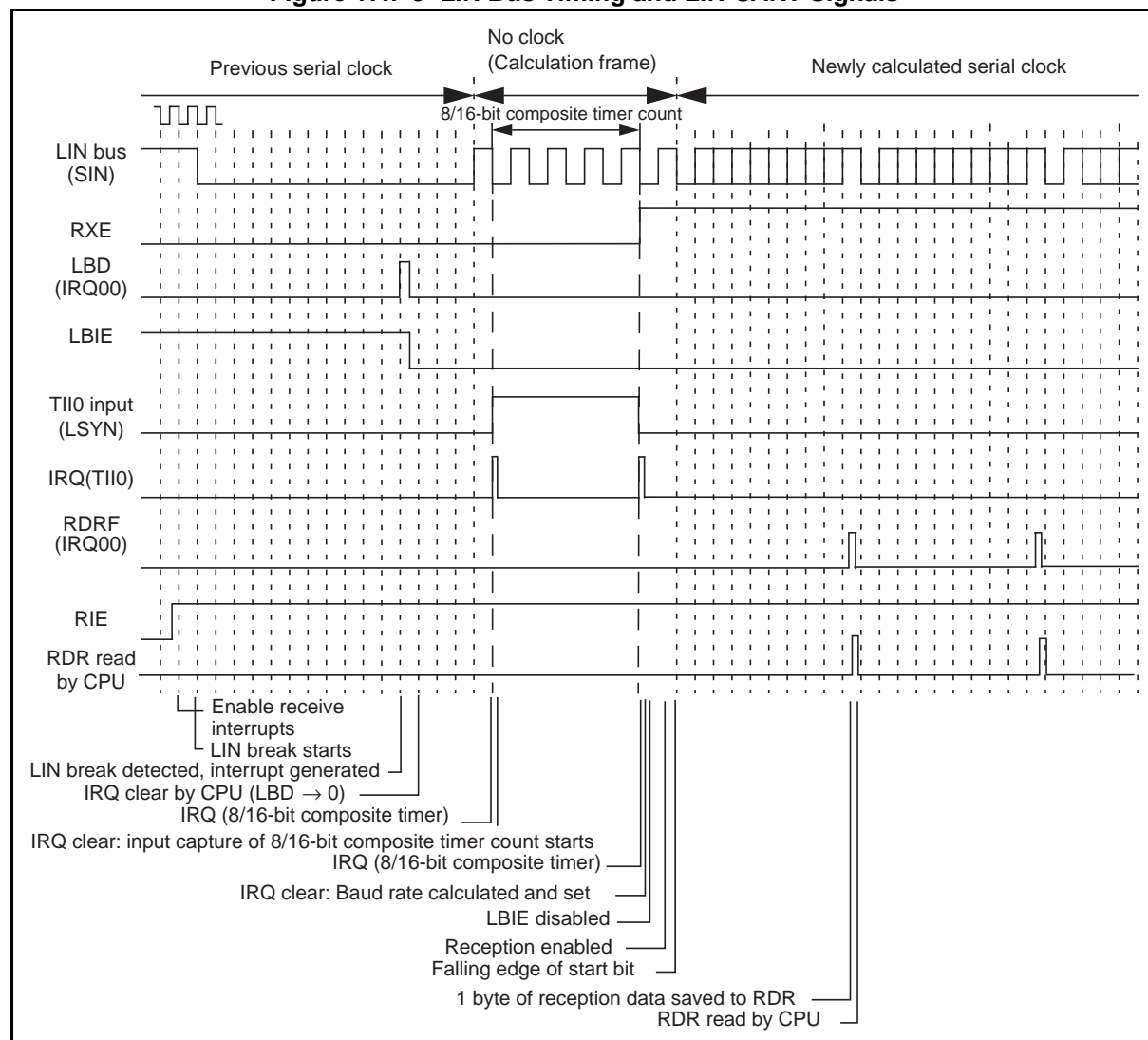
17.7.4 Serial Pin Direct Access

The transmit pin (SOT) and the receive pin (SIN) can be accessed directly.

■ LIN-UART Pin Direct Access

The LIN-UART allows the programmer to directly access the serial I/O pins.

The status of the serial input pin (SIN) can be read by using the serial I/O pin direct access bit (ESCR:SIOP).

To freely set the value of the serial output pin (SOT), enable the direct write access to the serial output pin (SOT) (ESCR:SOPE = 1), write "0" or "1" to the serial I/O pin direct access bit (ESCR:SIOP), and then enable serial output (SMR:SOE = 1).

In LIN mode, this feature is used for reading transmitted data and for error handling when there is a physical LIN bus line signal error.

Note:

Direct access is allowed only when transmission is not in progress (the transmit shift register is empty).

Before enabling transmission (SMR:SOE = 1), write a value to the serial output pin direct access bit (ESCR:SIOP). This prevents a signal of an unexpected level from being output since the SIOP bit holds a previous value.

While the value of the SIN pin is read by normal read, the value of the SOT pin is read from the SIOP bit by the read-modify-write (RMW) type of instruction.

MB95330H Series

17.7.5 Bidirectional Communication Function (Normal Mode)

Normal serial bidirectional communication can be performed in operating mode 0 or 2. Asynchronous mode can be selected in operating mode 0 and synchronous mode in operating mode 2.

■ Bidirectional Communication Function

To operate the LIN-UART in normal mode (operating mode 0 or 2), the settings shown in Figure 17.7-10 are required.

Figure 17.7-10 Settings of LIN-UART Operating Modes 0 and 2

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 0 →	◎	◎	◎	◎	×	0	◎	◎	0	0	0	◎	0	0	◎	◎
Mode 2 →	▣	▣	▣	+	×	0	◎	◎	1	0	◎	◎	0	0	◎	◎

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8								
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)							
Mode 0 →	◎	◎	◎	◎	◎	◎	◎	◎								
Mode 2 →	▣	◎	▣	◎	◎	◎	◎	◎								

ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 0 →	×	×	×	×	◎	◎	0	0	0	0	×	×	×	0	◎	◎
Mode 2 →	×	×	×	×	◎	◎	▣	◎	0	×	◎	◎	◎	0	▣	▣

◎ : Bit to be used
× : Unused bit
1 : Set to "1"
0 : Set to "0"
▣ : Used when SSM = 1 (Synchronous star/stop bit mode)
+ : Bit correctly set automatically

● Inter-CPU connection

When using bidirectional communication, connect two CPUs as shown in Figure 17.7-11.

Figure 17.7-11 Example of Connection for Bidirectional Communication in LIN-UART Mode 2

CPU1 (Serial clock transmit side) — SOT, SIN, SCK, Output
CPU2 (Serial clock receive side) — SOT, SIN, SCK, Input
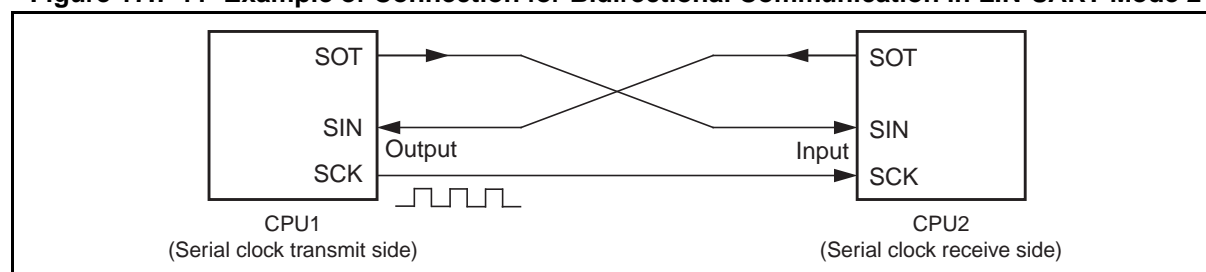
● Communication procedure example

The communication starts from the transmit side at any time after transmit data is ready. The receive side returns ANS (per one byte in this example) regularly after receiving transmit data. Figure 17.7-12 is an example of bidirectional communication flow chart.

Figure 17.7-12 Example of Bidirectional Communication Flow Chart

MB95330H Series
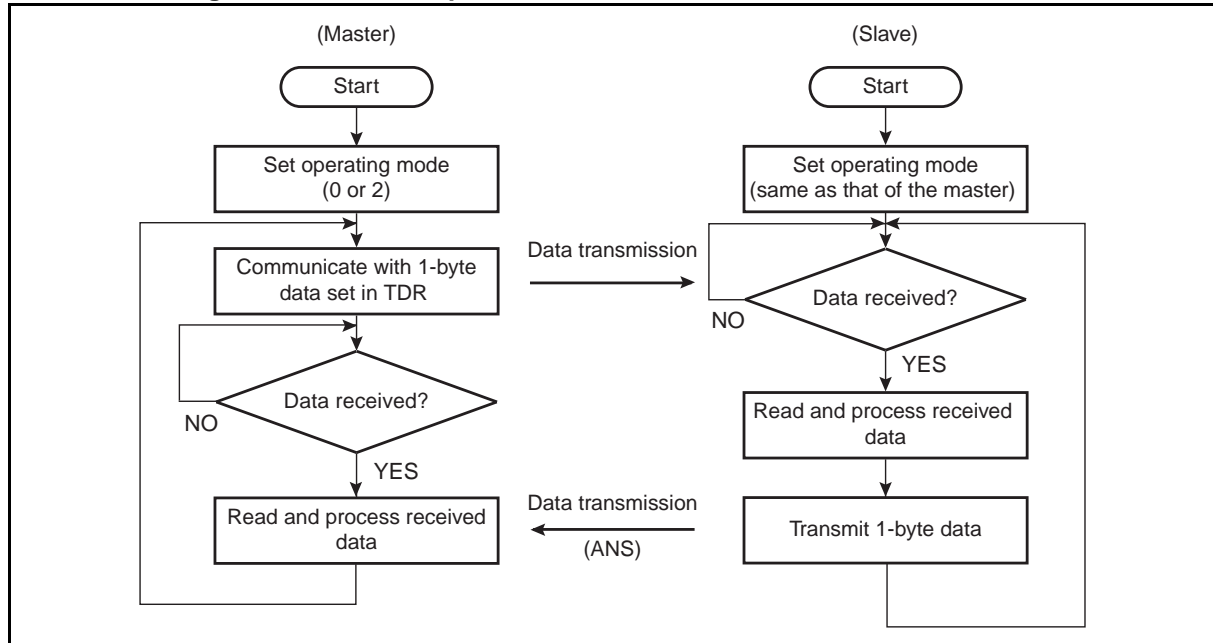
17.7.6 Master/Slave Mode Communication Function (Multiprocessor Mode)

Operating mode 1 allows communication among multiple CPUs connected in master/slave mode. The LIN-UART can be used as a master or a slave.

■ Master/Slave Mode Communication Function

To operate the LIN-UART in multiprocessor mode (operating mode 1), the settings shown in Figure 17.7-13 are required.

Figure 17.7-13 Settings of LIN-UART Operating Mode 1

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 1 →	+	×	◎	◎	◎	0	◎	◎	0	1	0	◎	0	0	◎	◎

SSR, RDR1/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set compare data (during writing)							
									Retain receive data (during reading)							
Mode 1 →	×	◎	◎	◎	◎	◎	◎	◎								

ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 1 →	×	×	×	×	◎	◎	0	0	0	×	×	×	×	0	◎	◎

◎ : Bit to be used
× : Unused bit
1 : Set to "1"
0 : Set to "0"
+ : Bit correctly set automatically

● Inter-CPU connection

For master/slave mode communication, a communication system consists of two common communication lines connecting between one master CPU and multiple slave CPUs as shown in Figure 17.7-14. The LIN-UART can be used as a master or a slave.

Figure 17.7-14 Connection Example of LIN-UART Master/Slave Mode Communication
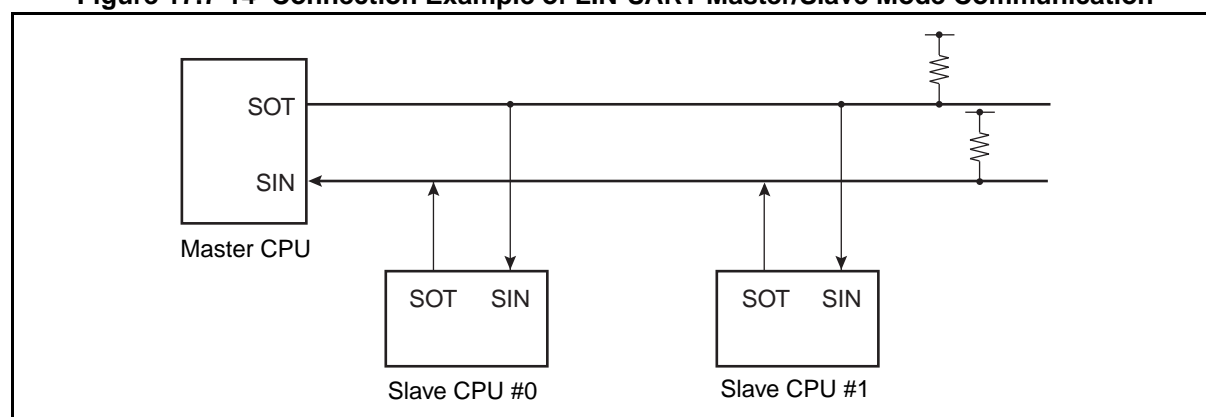
● Function selection

In master/slave mode communication, select the operating mode and the data transfer method as shown in Table 17.7-4.

Table 17.7-4 Selection of Master/Slave Mode Communication Functions

	Operating mode		Data	Parity	Synchronous method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission/ reception	Mode 1 (Transmit/ receive AD bit)	Mode 1 (Transmit/ receive AD bit)	AD = 1 + 7-bit or 8-bit address	None	Asynchronous	1 bit or 2 bits	LSB first or MSB first
Data transmission/ reception			AD = 0 + 7-bit or 8-bit data				

● Communication procedure

Master/slave mode communication starts as the master CPU transmits address data. The address data, which is the data chosen when the AD bit is set to "1", determines the slave CPU that is to be the destination of the communication. A slave CPU uses a program to check address data, and communicates with the master CPU when the address data matches the address assigned to that slave CPU.

Figure 17.7-15 is a flow chart showing master/slave mode communication (multiprocessor mode).

MB95330H Series

Figure 17.7-15 Master/Slave Mode Communication Flow Chart
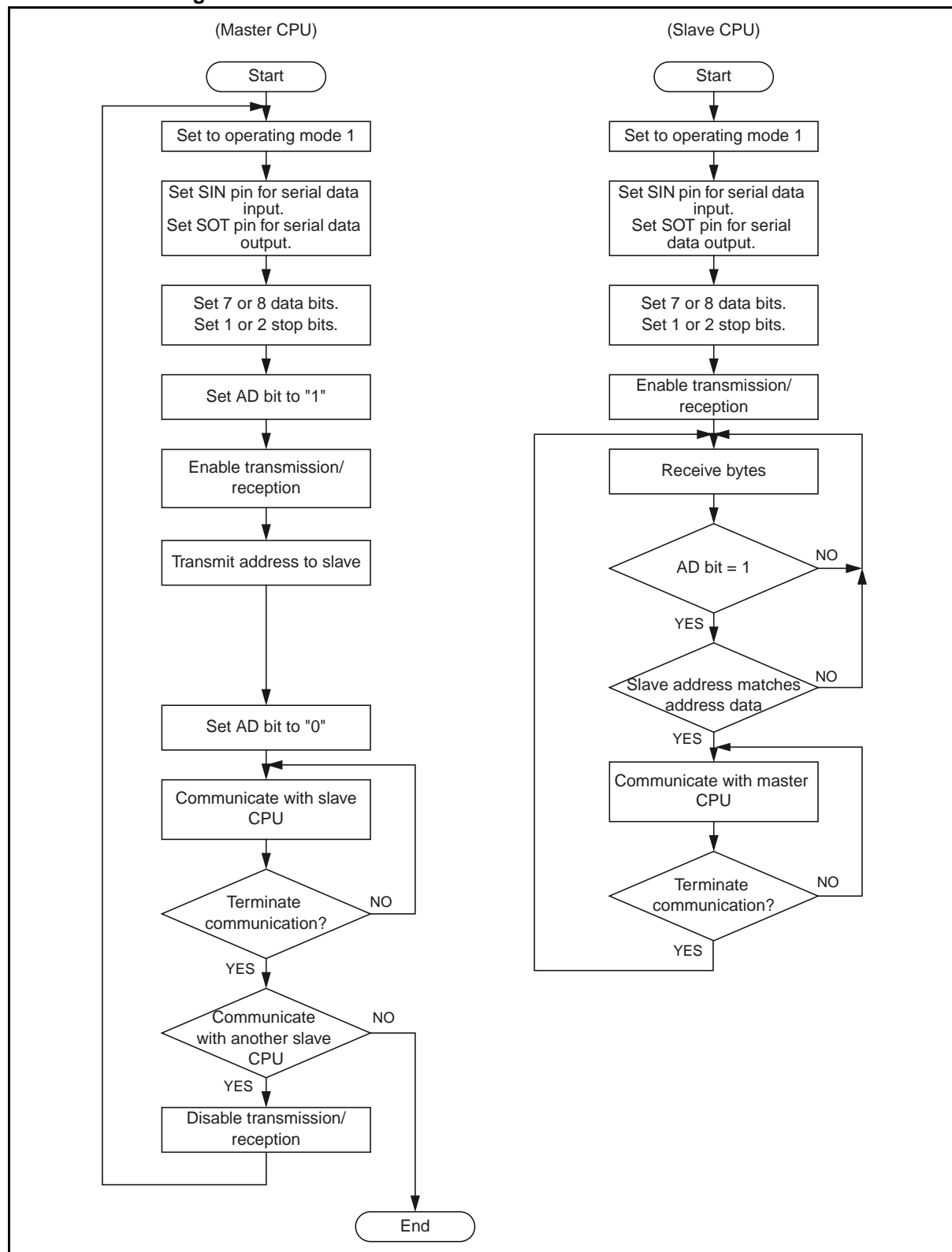
17.7.7 LIN Communication Function

In LIN-UART communication, a LIN device can be used in a LIN master system or a LIN slave system.

■ LIN Master/Slave Mode Communication Function

Figure 17.7-16 shows the required settings for the LIN communication mode (operating mode 3) of the LIN-UART.

Figure 17.7-16 Settings of LIN-UART Operating Mode 3 (LIN)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 3 →	+	×	+	+	×	0	◎	◎	1	1	0	◎	0	0	◎	◎

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8				
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing)			
									Retain reception data (during reading)			
Mode 3 →	×	◎	◎	◎	◎	+	◎	◎				

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 3 →	◎	◎	◎	◎	◎	◎	0	0	0	◎	×	×	×	0	◎	◎

◎ : Bit to be used
× : Unused bit
1 : Set to "1"
0 : Set to "0"
+ : Bit correctly set automatically

● LIN device connection

Figure 17.7-17 shows an example of communication in a LIN bus system.

The LIN-UART can operate as a LIN master or a LIN slave.

Figure 17.7-17 Example of LIN Bus System Communication

MB95330H Series
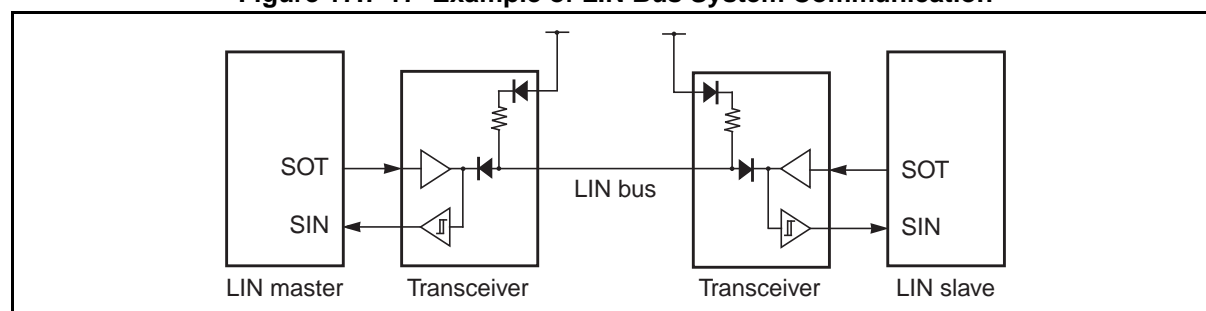
17.7.8 Examples of LIN-UART LIN Communication Flow Chart (Operating Mode 3)

This section shows examples of LIN-UART LIN communication flow charts.

■ LIN Master Device

Figure 17.7-18 LIN Master Flow Chart

* 1: If an error occurs, proceed to process the error.
* 2: - If the FRE or ORE flag is set to "1", write "1" to the SCR:CRE bit to clear the error flag.
    - If the ESCR:LBD bit is set to "1", execute the LIN-UART reset.
Note: Deal properly with any error detected in a process.

■ LIN Slave Device
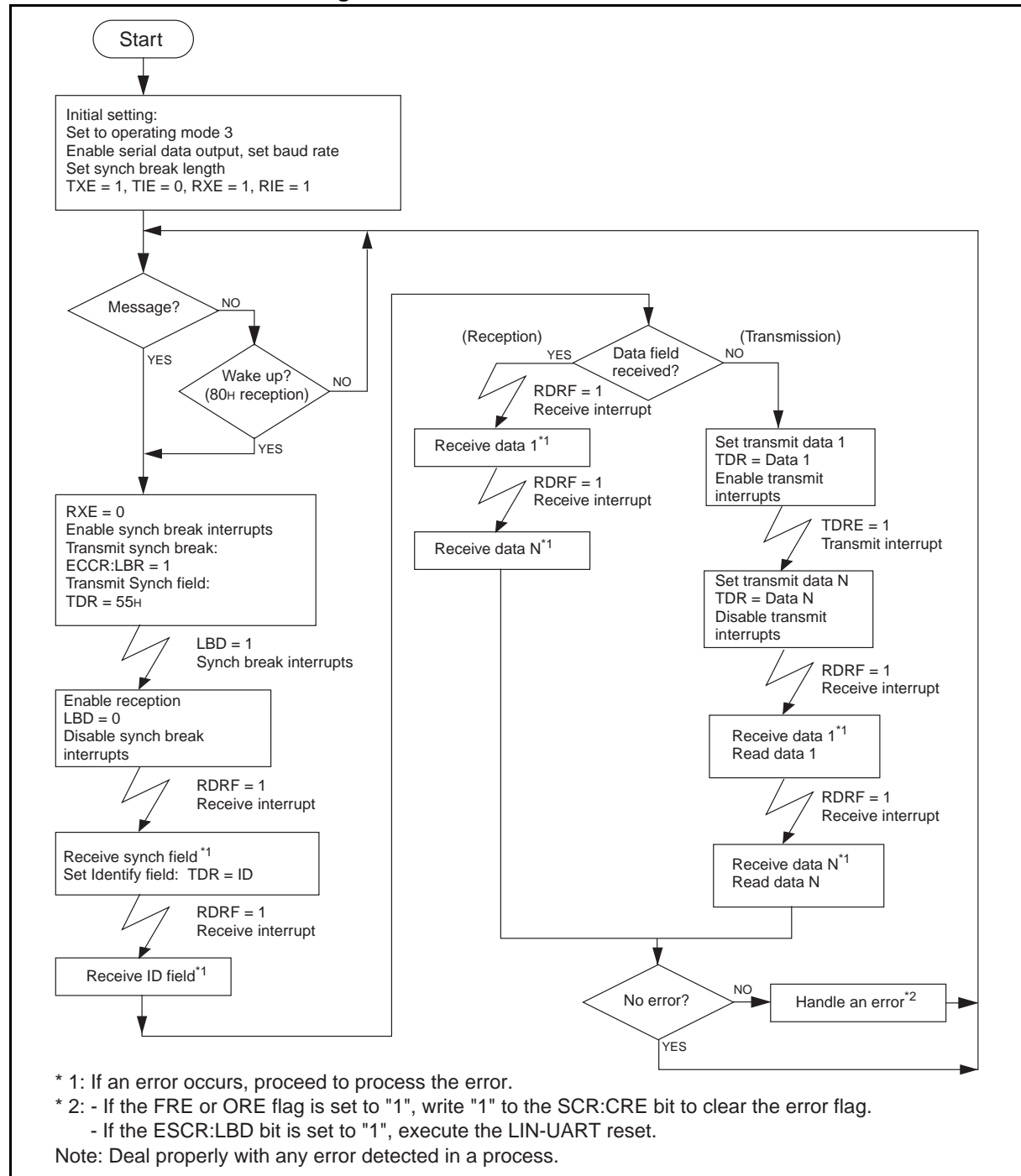
Figure 17.7-19 LIN Slave Flow Chart

* 1: If an error occurs, proceed to process the error.
* 2: - If the FRE or ORE flag is set to "1", write "1" to the SCR:CRE bit to clear the error flag.
   - If the ESCR:LBD bit is set to "1", execute the LIN-UART reset.
Note: Deal properly with any error detected in a process.

MB95330H Series
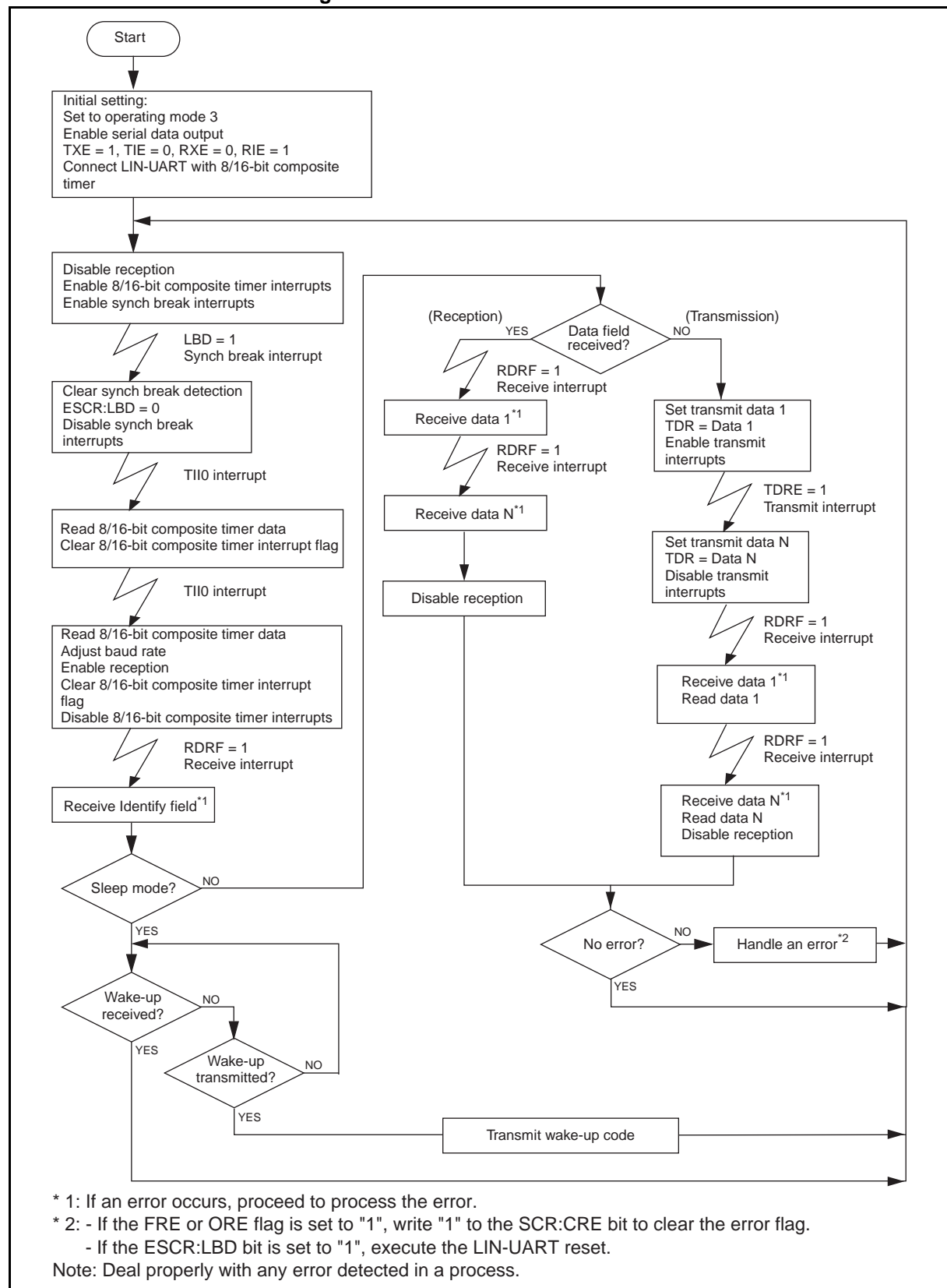
17.8 Notes on Using LIN-UART

This section provides notes on using the LIN-UART.

■ Notes on Using LIN-UART

● Enabling operation

The LIN-UART has the TXE bit and the RXE bit in the LIN-UART serial control register (SCR) to enable transmission and reception respectively. Since both transmission and reception are disabled by default (initial values), they must be enabled before the transfer starts. Transmission and reception can be disabled to stop transfer if necessary.

● Setting communication mode

The communication mode should be set while the LIN-UART stops operating. If the communication mode is set while transmission or reception is in progress, the integrity of data being transmitted or received at the setting of the mode is not guaranteed.

● Timing of enabling transmit interrupts

Since the default (initial) value of the transmit data empty flag bit (SSR:TDRE) is "1" (no transmit data, transmit data write enabled), a transmit interrupt request is made immediately after the transmit interrupt request is enabled (SSR:TIE = 1). To prevent any transmit interrupt request from being made, always set the TIE flag bit to "1" after setting transmit data.

● Modifying operation settings

After modifying operation settings such as the addition of start/stop and changing the data format, reset the LIN-UART.

Even though the setting of the LIN-UART serial mode register (SMR) and the resetting of the LIN-UART (SMR:UPCL = 1) are executed simultaneously, that does not ensure that the operation settings are correct. Therefore, after setting the LIN-UART serial mode register (SMR), reset the LIN-UART again.

● Using LIN functions

The LIN functions are available in operating mode 3. In the same mode, the communication format is predefined (8-bit data, no parity, one stop bit, LSB first).

While the length of the LIN synch break transmit bit is variable, in detection, the bit length is fixed at 11 bits.

● LIN slave settings

Before the LIN-UART starts operating as a slave, the baud rate must be set before the first LIN synch break is received to ensure that a LIN synch break whose length is a minimum of 13 bits is successfully detected.

● Bus idle function

The bus idle function is not available in synchronous mode (operating mode 2).

● AD bit (LIN-UART serial control register (SCR): Address/data format select bit)

Pay attention to the following issues when using the AD bit.

The AD bit is used to select the address/data for transmission by writing a value to it. When the AD bit is read, it returns the value of the AD bit received last. Inside the microcontroller, the AD bit value received and the one transmitted are saved in separate registers.

The AD bit value transmitted is read when the read-modify-write (RMW) type of instruction is used. Therefore, if another bit in the SCR register is accessed by bit access, an incorrect value may be written to the AD bit.

For the above reason, the AD bit must be set by the last access to the SCR register before transmission. The above problem can also be prevented by always using byte access to write values to the SCR register.

● LIN-UART software reset

Execute the LIN-UART software reset (SMR:UPCL = 1) when the TXE bit in the LIN-UART serial control register (SCR) is "0".

● Synch break detection

In operating mode 3 (LIN mode), when serial input is 11 bits or more in width and becomes "L", the LBD bit in the extended status control register (ESCR) is set to "1" (synch break detected) and the LIN-UART waits for the synch field. Therefore, when serial input has more than 11 bits of "0" not at the time of a synch break, the LIN-UART recognizes that a synch break has been input (LBD = 1) and then waits for the synch field.

In this case, execute the LIN-UART reset (SMR: UPCL = 1).

MB95330H Series

17.9   Sample Settings for LIN-UART

This section provides sample settings for the LIN-UART.

■ Sample Settings

● Method of selecting an operating mode

Use the operating mode select bits (SMR:MD[1:0]).

Operating mode		Operating mode select bits (MD[1:0])
Mode 0	Asynchronous (Normal mode)	Set the bits to "00_B".
Mode 1	Asynchronous (Multiprocessor mode)	Set the bits to "01_B".
Mode 2	Synchronous (Normal mode)	Set the bits to "10_B".
Mode 3	Asynchronous (LIN mode)	Set the bits to "11_B".

● Types of operating clock and method of selecting an operating clock

Use the external clock select bit (SMR:EXT).

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

● Method of controlling the SCK, SIN, and SOT pins

Use the following settings.

	LIN-UART
To set the SCK pin as an input pin	DDR0:P02 = 0 SMR:SCKE = 0
To set the SCK pin as an output pin	SMR:SCKE = 1
To use the SIN pin	DDR0:P04 = 0
To use the SOT pin	SMR:SOE = 1

● Method of enabling/disabling the LIN-UART operation

Use the receive operation enable bit (SCR:RXE).

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

Use the transmit operation control bit (SCR:TXE).

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

● Method of using an external clock as the serial clock of the LIN-UART

Use the one-to-one external clock input enable bit (SMR:OTO).

Operation	One-to-one external clock input enable bit (OTO)
To enable external clock	Set the bit to "1".

● Method of restarting the reload counter

Use the reload counter restart bit (SMR:REST).

Operation	Reload counter restart bit (REST)
To restart the reload counter	Set the bit to "1".

● Method of resetting the LIN-UART

Use the LIN-UART programmable clear bit (SMR:UPCL).

Operation	LIN-UART programmable clear bit (UPCL)
To reset the LIN-UART with software reset	Set the bit to "1".

● Method of setting the parity

Use the parity enable bit (SCR:PEN) and the parity select bit (SCR:P).

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

MB95330H Series

● Method of setting the data length

Use the data length select bit (SCR:CL).

Operation	Data length select bit (CL)
To set the bit length to 7 bits	Set the bit to "0".
To set the bit length to 8 bits	Set the bit to "1".

● Method of selecting the stop bit length

Use the stop bit length select bit (SCR:SBL).

Operation	Stop bit length select bit (SBL)
To set the stop bit length to 1	Set the bit to "0".
To set the stop bit length to 2	Set the bit to "1".

● Method of clearing the error flag

Use the receive error flag clear bit (SCR:CRE).

Operation	Receive error flag clear bit (CRE)
To clear the error flag (PE, ORE, FRE)	Set the bit to "1".

● Method of setting the transfer direction

Use the transfer direction select bit (SSR:BDS).

In all operating modes, the transfer direction can be selected from LSB-first and MSB-first.

Operation	Transfer direction select bit (BDS)
To select the LSB-first (from the least significant bit)	Set the bit to "0".
To select the MSB-first (from the most significant bit)	Set to the bit "1".

● Method of clearing the receive completion flag

Use the following method.

Operation	Method
To clear the receive completion flag	Read the RDR register.

Reception starts at the first time the RDR register is read.

● Method of clearing the transmit buffer empty flag

Use the following method.

Operation	Method
To clear the transmit buffer empty flag	Write data to the TDR register.

Transmission starts at the first time data is written to the TDR register.

● Method of selecting the data format (address/data) (only in mode 1)

Use the address/data format select bit (SCR:AD).

Operation	Address/data format select bit (AD)
To select the data frame	Set the bit to "0".
To select the address frame	Set the bit to "1".

The setting is effective only in transmission. The AD bit is ignored in reception.

● Method of setting the baud rate

See Section 17.6 "LIN-UART Baud Rate".

● Interrupt-related registers

Interrupt level is set by interrupt level setting registers as shown in the following table.

	Interrupt level setting register	Interrupt vector
Reception	Interrupt level register (ILR1) Address: $0007A_H$	#7 Address: $0FFEC_H$
Transmission	Interrupt level register (ILR2) Address: $0007B_H$	#8 Address: $0FFEA_H$

MB95330H Series

● Method of enabling/disabling/clearing interrupts

Interrupt request enable flag, interrupt request flag

Use the interrupt request enable bits (SSR:RIE), (SSR:TIE) enable respective interrupts.

	UART reception	UART transmission
	Receive interrupt enable bit (RIE)	Transmit interrupt enable bit (TIE)
To disable interrupt requests	Set the bit to "0".	
To enable interrupt requests	Set the bit to "1".	

Use the following setting to clear interrupt requests.

	UART reception	UART transmission
To clear interrupt requests	The receive data register full flag bit (RDRF) is cleared by reading the LIN-UART serial input register (RDR).	The transmit data register empty flag bit (TDRE) is set to "0" by writing data to the LIN-UART serial output data register (TDR).
	The error flag (PE, ORE or FRE) is set to "0" by writing "1" to the error flag clear bit (CRE).	

CHAPTER 18
8/10-BIT A/D CONVERTER

This chapter describes the functions and operations of the 8/10-bit A/D converter.

18.1 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 AD bit in the ADC1 register
- Continuous activation using the 8/16-bit composite timer output TO00

MB95330H Series

18.2　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 18.2-1 is the block diagram of the 8/10-bit A/D converter.

Figure 18.2-1　Block Diagram of 8/10-bit A/D Converter
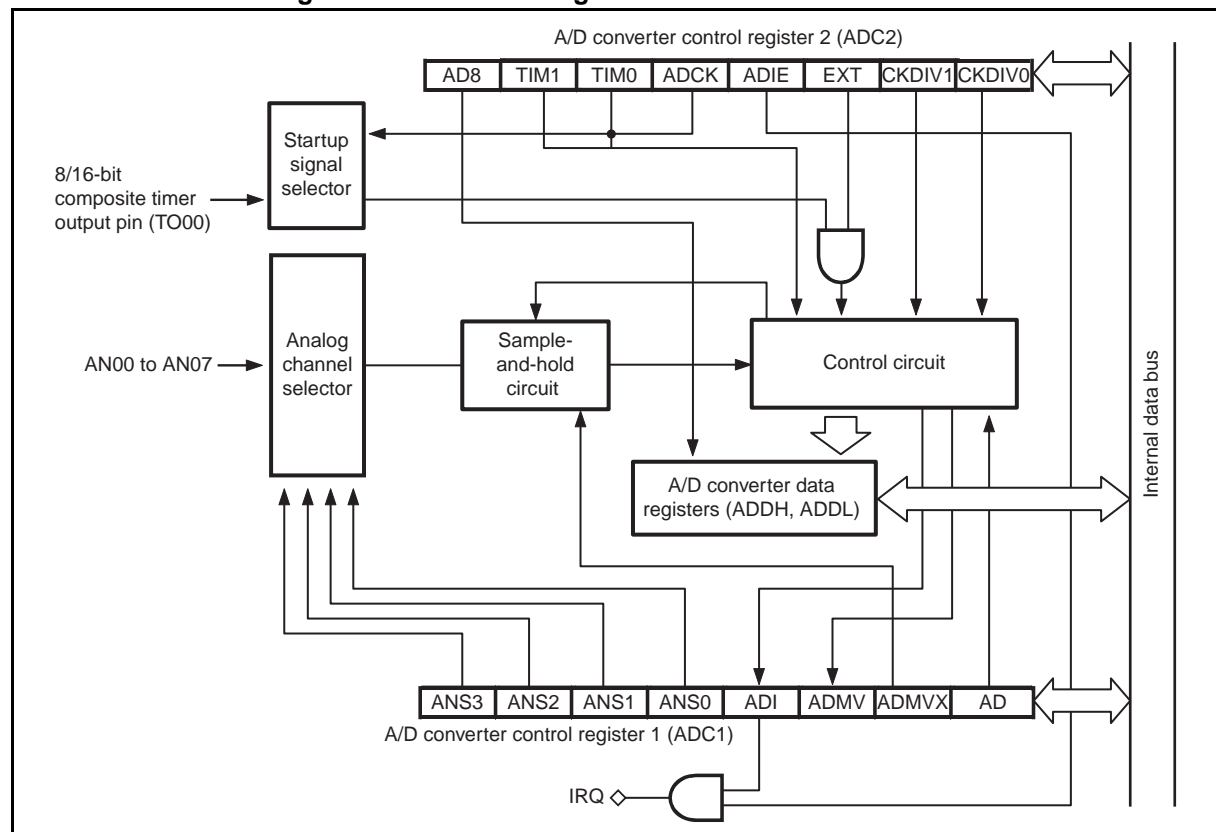
● Clock selector

This selects the A/D conversion clock with continuous activation having been enabled (ADC2:EXT = 1).

● Analog channel selector

This is the circuit selecting an input channel from several analog input pins.

● Sample-and-hold circuit

This circuit holds input voltage selected by the analog channel selector. By sampling the input voltage and holding it immediately after A/D conversion starts, this circuit prevents A/D conversion from being affected by the fluctuation in input voltage during the conversion (comparison).

● Control circuit

The A/D conversion function determines the values in the 10-bit A/D data register sequentially from MSB to LSB based on the voltage compare signal from the comparator. When A/D conversion is completed, the A/D conversion function sets the interrupt request flag bit (ADC1: ADI) to "1".

● A/D converter data registers (ADDH/ADDL)

The upper two bits of 10-bit A/D data are stored in the ADDH register; the lower eight bits in the ADDL register.

If the A/D conversion precision bit (ADC2:AD8) is set to "1", the A/D conversion precision becomes 8-bit precision, and the upper eight bits of 10-bit A/D data can be obtained by reading 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).

MB95330H Series

18.3 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 MB95330H Series has eight channels of analog input pin.

The analog input pins are also used as general-purpose I/O ports.

● AN07 pin to AN00 pin

AN07 to AN00: When using the A/D conversion function, input to one of these pins the analog voltage to be converted. A pin of AN07 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.

■ Block Diagrams of Pins of 8/10-bit A/D Converter

Figure 18.3-1 Block Diagram of Pins AN00 and AN01 (P00/INT00/AN00 and P01/INT01/AN01) of 8/10-bit A/D Converter
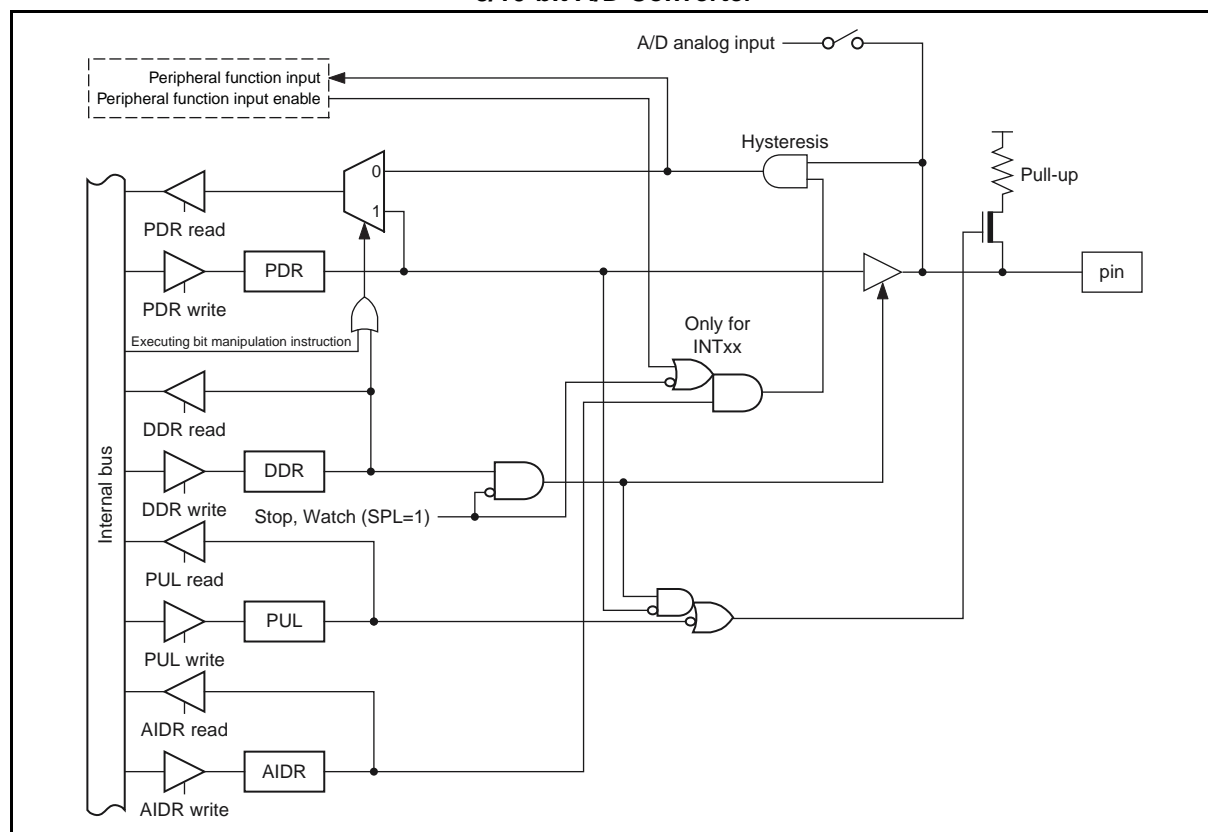
Figure 18.3-2 Block Diagram of Pins AN02, AN03 and AN05 (P02/INT02/AN02/SCK, P03/INT03/AN03/SOT and P05/INT05/AN05/TO00/HCLK2) of 8/10-bit A/D Converter
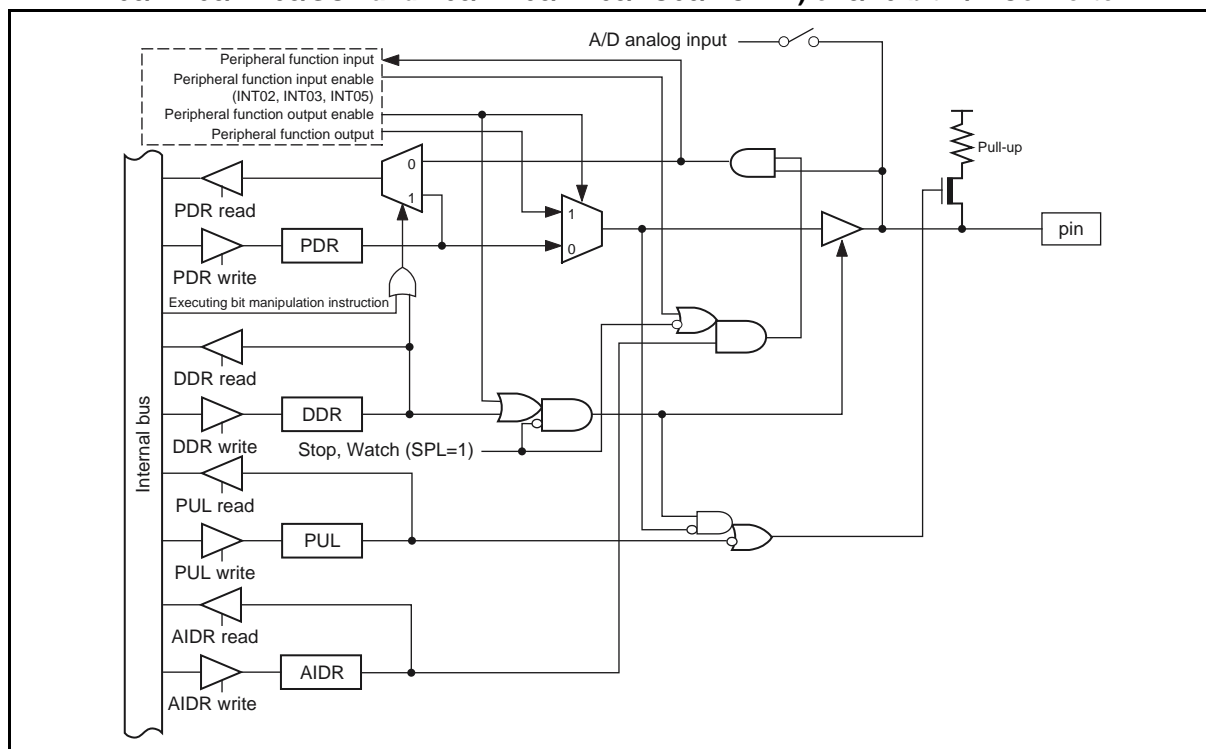
Figure 18.3-3 Block Diagram of Pin AN04 (P04/INT04/AN04/SIN/HCLK1/EC0) of 8/10-bit A/D Converter

Figure 18.3-4 Block Diagram of Pin AN06 (P06/INT06/AN06/TO01) of 8/10-bit A/D Converter

Figure 18.3-5 Block Diagram of Pin AN07 (P07/INT07/AN07) of 8/10-bit A/D Converter

MB95330H Series

18.4 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 18.4-1 lists the registers of the 8/10-bit A/D converter.

Figure 18.4-1 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

18.4.1 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 18.4-2 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 bits
0	0	0	0	AN00 pin
0	0	0	1	AN01 pin
0	0	1	0	AN02 pin
0	0	1	1	AN03 pin
0	1	0	0	AN04 pin
0	1	0	1	AN05 pin
0	1	1	0	AN06 pin
0	1	1	1	AN07 pin

R/W   : Readable/writable (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

Table 18.4-1 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 AN07. 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: Current cutoff analog switch 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.

18.4.2 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 18.4-3 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	No external start signal is used.
1	Starts by 8/16-bit composite timer output pin (TO00).

TIM1	TIM0	Sampling time select bits
0	0	CKIN × 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 18.4-2 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.

18.4.3 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 18.4-4 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 data 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 data.

These two registers are read-only registers. Writing data to them has no effect on operation.

In A/D conversion in which 8-bit precision is selected, SAR8 and SAR9 in the ADDH register become "0".

● A/D conversion function

When A/D conversion is started, the results of conversion are finalized and stored in the ADDH and ADDL registers after the conversion time according to the register settings elapses. After A/D conversion is completed and before the next A/D conversion is completed, read A/D data registers (conversion results), and clear the ADI flag bit (bit 3) in the ADC1 register. During A/D conversion, the values of the ADDH and ADDL registers are results of the last A/D conversion.

18.5 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 18.5-1 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.

18.6 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 in the ADC2 register.

■ Operations of 8/10-bit A/D Converter Conversion Function

● Software activation

The settings shown in Figure 18.6-1 are required for activating the A/D conversion function with the software.

Figure 18.6-1 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.

MB95330H Series

● Continuous activation

The settings shown in Figure 18.6-2 are required for continuous activation of the A/D conversion function.

Figure 18.6-2 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).

Notes:

- The contents of the ADDH and ADDL registers are retained until the end of A/D conversion. Therefore, during A/D conversion, the values resulting from last conversion will be returned if the two registers are read.

- Do not change the analog input pin (ADC1: ANS3 to ANS0) while AD conversion function is being used. During continuous activation in particular, disable continuous activation (ADC2: EXT = 0) before changing the analog input pin.

- The start of the reset mode, the stop mode or the watch mode causes the A/D converter to stop and the ADMV bit to be cleared to "0".

■ Setting Procedure Example

Below is an example of procedure for setting the 8/10-bit A/D converter:

● Initial settings

1) Set the input port (DDR0).

2) Set the interrupt level (ILR4).

3) Enable A/D input (ADC1:ANS0 to ANS3).

4) Set the sampling time (ADC2:TIM1, TIM0).

5) Select the clock (ADC2:CKDIV1, CKDIV0).

6) Set A/D conversion 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).

MB95330H Series

18.7   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 AN07) 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 AN07), or after disconnecting analog input (AN00 to AN07).

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.

FUJITSU SEMICONDUCTOR LIMITED

- 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 MB95330H Series.

MB95330H Series

18.8 Sample Settings for 8/10-bit A/D Converter

This section provides sample settings for the 8/10-bit A/D Converter.

■ Sample Settings

● 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.

Operation	Analog switch for current cutoff control bit (ADMVX)
To switch off internal reference power supply	Set the bit to "0".
To switch on internal reference power supply	Set the bit to "1".

● Method of selecting the method of activating the 8/10-bit A/D conversion function

Use the continuous activation enable bit (ADC2:EXT) to select an activation trigger.

A/D conversion activation source	Continuous activation enable bit (EXT)
To select the software trigger	Set the bit to "0".
To select the input clock rising signal	Set the bit to "1".

• Method of generating a software trigger

Use the A/D conversion start bit (ADC1:AD) to generate a software trigger.

Operation	A/D conversion start bit (AD)
To generate a software trigger	Set the bit to "1".

- Method of activating the A/D conversion function using the input clock

   An activation trigger is generated at the rising edge of the input clock.

To select the input clock, use external start signal select bit (ADC2:ADCK).

Input clock	External start signal select bit (ADCK)
Do not use any external start signal	Set the bit to "0".
To select the 8/16-bit composite timer output pin (TO00)	Set the bit to "1".

● Method of selecting A/D conversion precision

   Use the precision select bit (ADC2:AD8) to select the precision of conversion results.

Operating mode	Precision select bit (AD8)
To select 10-bit precision	Set the bit to "0".
To select 8-bit precision	Set the bit to "1".

● Method of using analog input pins

   Use the analog input pin select bits (ADC1:ANS3 to ANS0) to select an analog input pin.

Operation	Analog input pin select bits (ANS3 to ANS0)
To use the 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".

● 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.

Operation	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.

Operation	Interrupt request bit (ADI)
To clear an interrupt request	Set the bit to "1" or activate the A/D conversion function.

CHAPTER 19

LOW-VOLTAGE DETECTION RESET CIRCUIT

This chapter describes the function and operation of the low-voltage detection reset circuit. (The low-voltage detection reset circuit is available in MB95F332K/F333K/F334K only.)

19.1 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 MB95F332K/F333K/F334K 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 MB95F332K/F333K/F334K only. Refer to the data sheet of the MB95330H Series for details of the electrical characteristics.

MB95330H Series

19.2 Configuration of Low-voltage Detection Reset Circuit

Figure 19.2-1 is the block diagram of the low-voltage detection reset circuit.

■ **Block Diagram of Low-voltage Detection Reset Circuit**

Figure 19.2-1 Block Diagram of Low-voltage Detection Reset Circuit
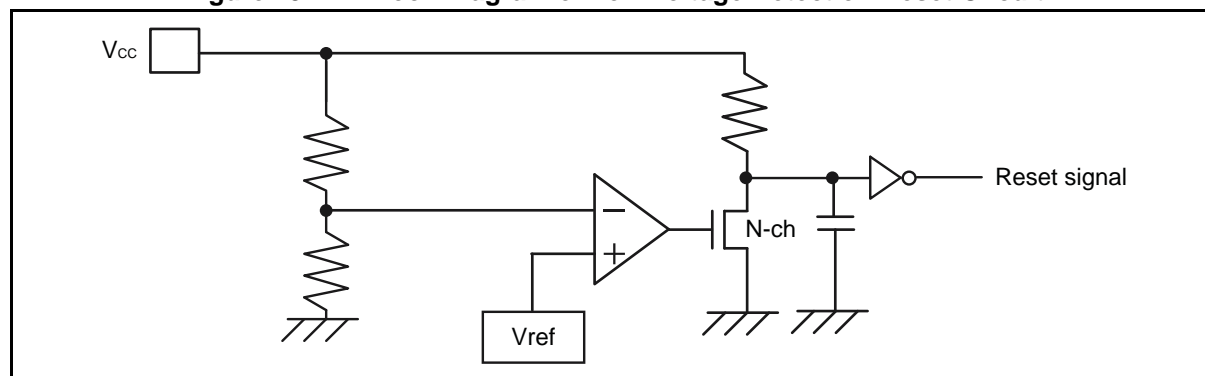
19.3 Pins of Low-voltage Detection Reset Circuit

This section describes the pins of the low-voltage detection reset circuit.

■ Pins of Low-voltage Detection Reset Circuit

● V_{CC} pin

The low-voltage detection reset circuit monitors the voltage of this pin.

● V_{SS} pin

This is the GND pin serving as the reference for voltage detection.

● $\overline{RST}$ pin

The low-voltage detection reset signal is output inside the microcontroller and to this pin.

MB95330H Series

19.4 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 MB95330H Series.

Figure 19.4-1 Operation of Low-voltage Detection Reset Circuit
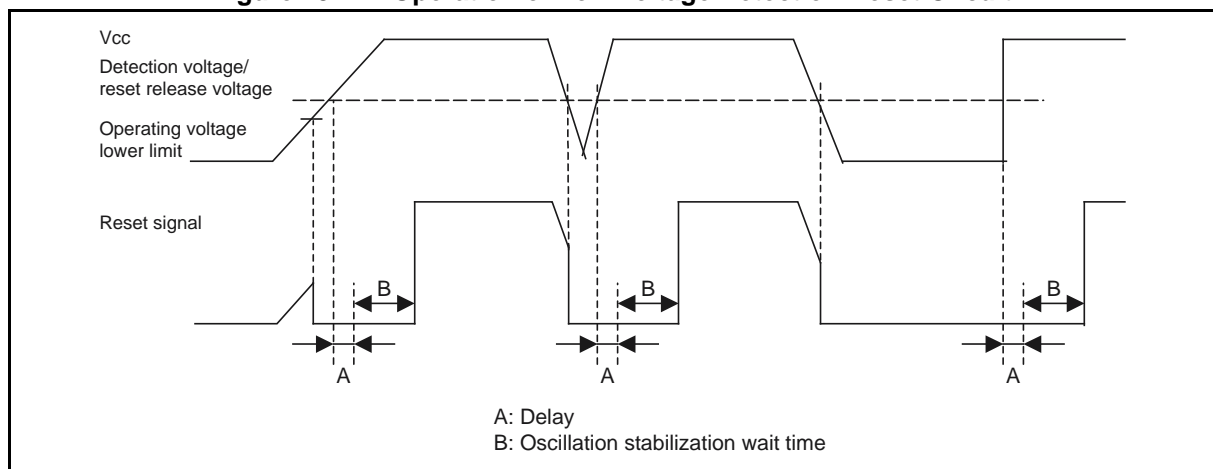
■ Operation in Standby Mode

The low-voltage detection reset circuit keeps operating even in standby mode (stop mode, sleep mode, subclock mode and watch mode).

CHAPTER 20
CLOCK SUPERVISOR COUNTER

This chapter describes the functions and operations of the clock supervisor counter.

20.1 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 counts up either according to a time-base timer interval selected from eight options or according to 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).

MB95330H Series

20.2 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 20.2-1 is the block diagram of the clock supervisor counter.

Figure 20.2-1 Block Diagram of Clock Supervisor Counter
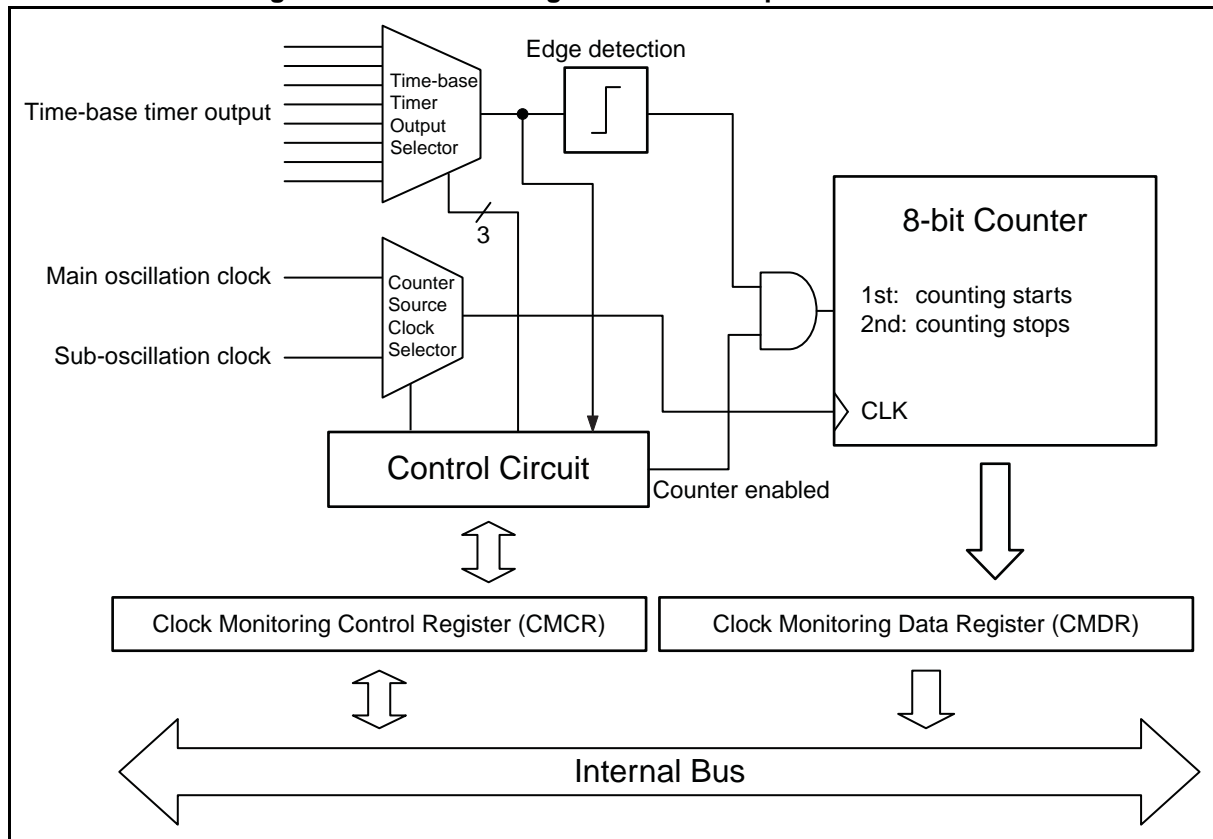
● Control circuit

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.

MB95330H Series

20.3　Registers of Clock Supervisor Counter

This section describes the registers of the clock supervisor counter.

■ Registers of Clock Supervisor Counter

Figure 20.3-1 shows the registers of the clock supervisor counter.

Figure 20.3-1 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	R0/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.)
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

20.3.1 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 content of this register.

■ Clock Monitoring Data Register (CMDR)

Figure 20.3-2 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 20.3-1 Functions of Bits in Clock Monitoring Data Register (CMDR)

Bit name		Function
bit7 to bit0	CMDR7 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 20.5-2 "Clock Supervisor Counter Operation 2").

Note:

The value of this register is "0" as long as the counter is operating (CMCEN = 1).

MB95330H Series

20.3.2   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 20.3-3 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	R0/W0	R/W	R/W	R/W	R/W	R/W	

CMCEN	Counter enable bit
0	Disables the counter.
1	Enables the counter.

TBTSEL2	TBTSEL2	TBTSEL2	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.)
R0/WX   : The read value is "0". Writing a value to it has no effect on operation.
R0/W0   : The write value is "0" and the write value "0".
-          : Undefined bit
          : Initial value

Table 20.3-2 Functions of Bits in Clock Monitoring Control Register (CMCR)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5	Reserved bit	This bit is a reserved bit. Always set this bit to "0". The read value is always "0".
bit4	CMCSEL: Counter clock select bit	This bit selects the counter clock source. **Writing "0":** selects the external main oscillation clock as the source clock of the counter. **Writing "1":** selects the external sub-oscillation clock as the source clock of the counter.
bit3 to bit1	TBTSEL2, TBTSEL1, TBTSEL0: Time-base timer counter output select bits	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. See table below
bit0	CMCEN: Counter enable bit	This bit enables and disables the clock supervisor counter. **Writing "0":** stops the counter and clears the CMDR register. **Writing "1":** enables the counter. The counter starts counting when detecting the rising edge of the 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.

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	$2^{17} \times 1/F_{CRH}$

Notes:
- Do not modify the CMCSEL bit when CMCEN = 1.
- Do not modify the TBTSEL[2:0] bits when CMCEN = 1.

MB95330H Series

20.4 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 20.4-1 Clock Supervisor Counter Operation 1
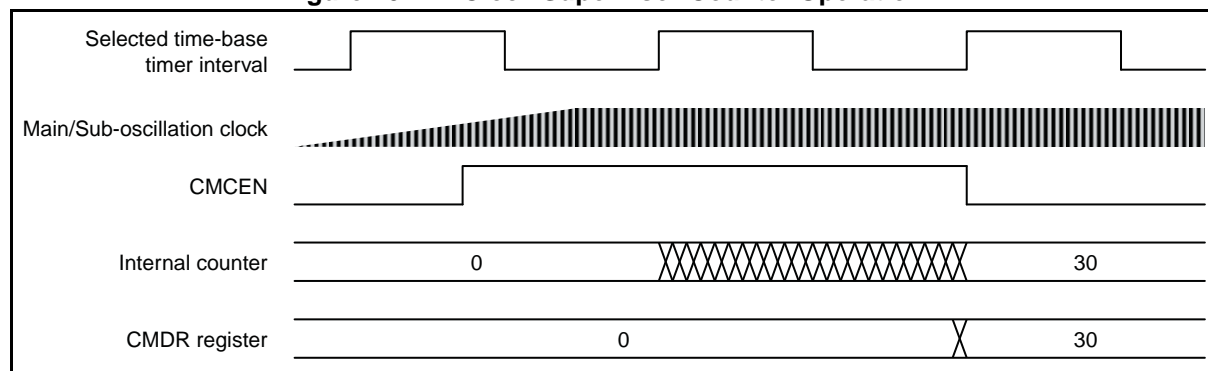
● Clock Supervisor Counter Operation 2

The CMDR register is cleared when the CMCEN bit changes from "0" to "1".

Figure 20.4-2 Clock Supervisor Counter Operation 2
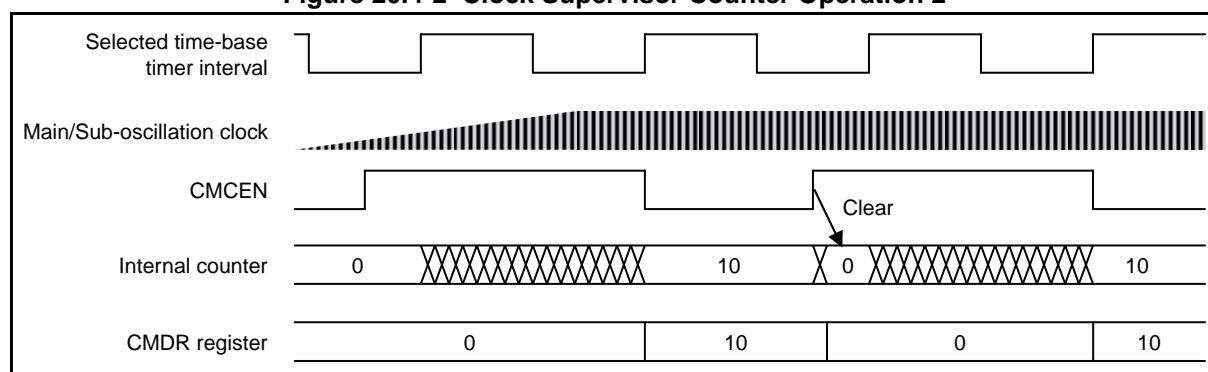
● Clock Supervisor Counter Operation 3

The counter stops counting if it reaches "255". It cannot count further than "255".

Figure 20.4-3 Clock Supervisor Counter Operation 3
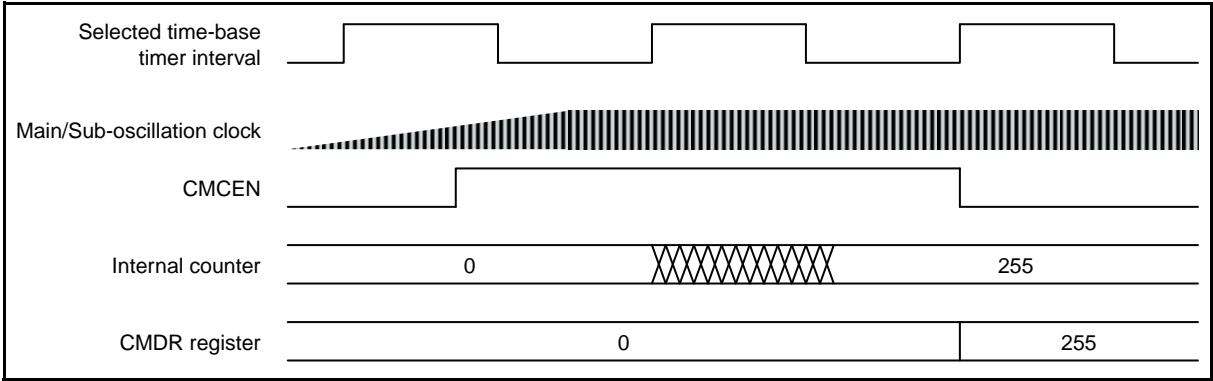
● Clock Supervisor Counter Operation 4

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 20.4-4 Clock Supervisor Counter Operation 4
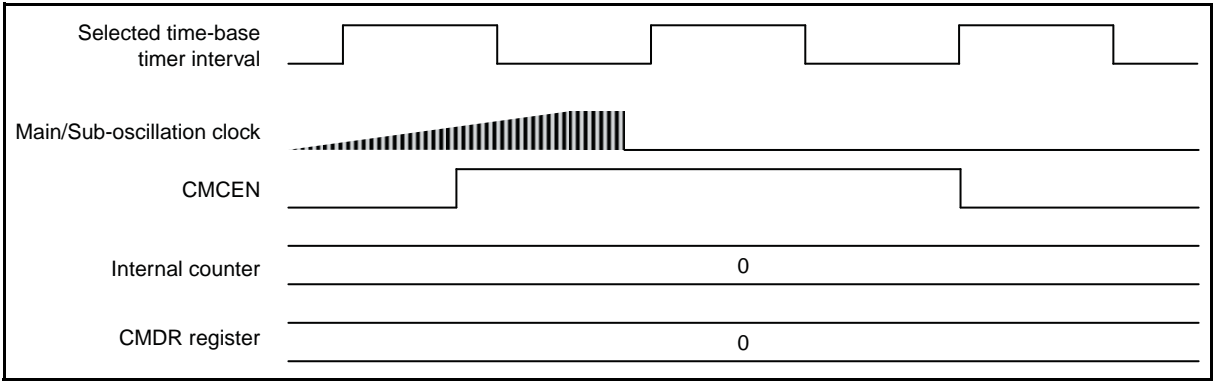
● Clock Supervisor Counter Operation 5

The counter is cleared to "0" by the software if the CMCEN is set to "0" while the counter is operating.

Figure 20.4-5 Clock Supervisor Counter Operation 5

MB95330H Series
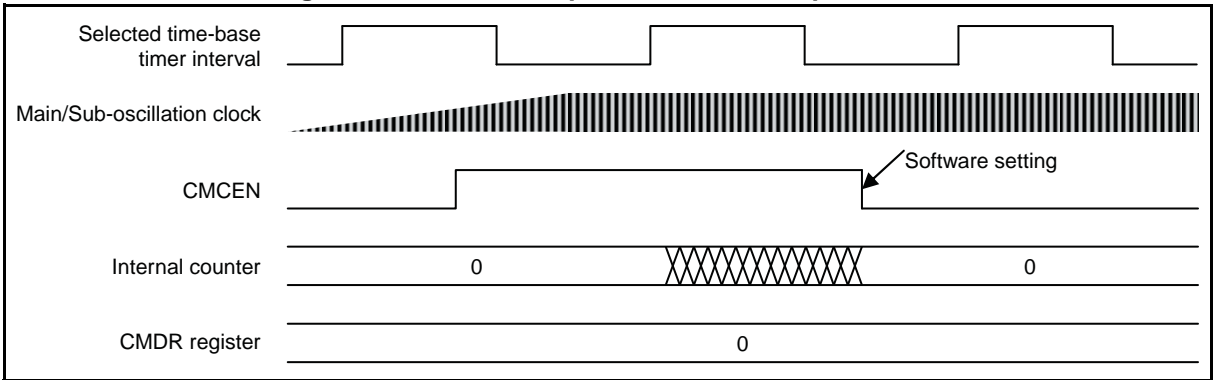
■ Table of Time-base Timer Intervals & Clock Supervisor Counter Values

Table 20.4-1 shows time-base timer intervals suitable for using different main CR clock frequency to measure different external clocks.

Table 20.4-1 Table of Counter Values in Relation to TBTSEL Settings (1 / 2)

Main CR (F_{CRH}) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measurement error	TBTSEL2 - TBTSEL0							
				000_B	001_B	010_B	011_B	100_B	101_B	110_B	111_B
				$(2^3 \times 1/F_{CRH})$	$(2^5 \times 1/F_{CRH})$	$(2^7 \times 1/F_{CRH})$	$(2^9 \times 1/F_{CRH})$	$(2^{11} \times 1/F_{CRH})$	$(2^{13} \times 1/F_{CRH})$	$(2^{15} \times 1/F_{CRH})$	$(2^{17} \times 1/F_{CRH})$
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 20.4-1 Table of Counter Values in Relation to TBTSEL Settings (2 / 2)

Main CR (F_CRH) [MHz]	Main/Sub-crystal oscillation [MHz]	Main CR error	Measurement error	TBTSEL2 - TBTSEL0							
				000_B ($2^3 \times 1/F_{CRH}$)	001_B ($2^5 \times 1/F_{CRH}$)	010_B ($2^7 \times 1/F_{CRH}$)	011_B ($2^9 \times 1/F_{CRH}$)	100_B ($2^{11} \times 1/F_{CRH}$)	101_B ($2^{13} \times 1/F_{CRH}$)	110_B ($2^{15} \times 1/F_{CRH}$)	111_B ($2^{17} \times 1/F_{CRH}$)
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	33113
	10	+5%	-1	2	11	47	194	779	3119	12482	49931
		-5%	+1	4	14	54	216	863	3450	13798	55189
	20	+5%	-1	5	23	96	389	1559	6240	24965	99863
		-5%	+1	7	27	108	432	1725	6899	27595	110377
	32.5	+5%	-1	8	38	157	632	2534	10141	40568	162278
		-5%	+1	11	44	176	701	2803	11211	44841	179362

    : Recommended setting

    : The counter value becomes "0" or "255".

MB95330H Series

Table 20.4-1 is calculated by the following equation:

$$\text{Counter value} = \frac{\begin{pmatrix} 2^3 \times 1/F_{CRH}(\text{TBTSEL=000}) \\ 2^5 \times 1/F_{CRH}(\text{TBTSEL=001}) \\ 2^7 \times 1/F_{CRH}(\text{TBTSEL=010}) \\ 2^9 \times 1/F_{CRH}(\text{TBTSEL=011}) \\ 2^{11} \times 1/F_{CRH}(\text{TBTSEL=100}) \\ 2^{13} \times 1/F_{CRH}(\text{TBTSEL=101}) \\ 2^{15} \times 1/F_{CRH}(\text{TBTSEL=110}) \\ 2^{17} \times 1/F_{CRH}(\text{TBTSEL=111}) \end{pmatrix} \times \text{Main/Sub-Oscillation Clock Frequency}}{2} \pm 1 \text{ (Measurement error)}$$

*Omit the decimal places of "Value".

Selected time-base timer interval

Within this period, the "Value" in the above equation is counted by the main/sub oscillation clock.

■ **Sample Operation Flow Chart of Clock Supervisor**

Figure 20.4-6 Sample Operation Flow Chart of Clock Supervisor

Clock supervision starts

Oscillation stabilization wait time elapses — NO

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 $\times$ 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 ($2^{13} \times 1/F_{CRH}$)

Notes:

- See Section 10.1 "Overview of Time-base Timer" for time-base timer interval settings.
- See Section 6.4 "Oscillation Stabilization Wait Time Setting Register (WATR)" for main/sub-oscillation stabilization time settings.

20.5 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 20.4-1 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 20.5-1 below shows the appropriate clock gear setting for each TBTSEL setting.

Table 20.5-1 Appropriate Clock Gear Setting for Respective TBTSEL Settings

DIV (clock gear setting)	TBTSEL2 to TBTSEL0		
	000_B	001_B	010_B to 111_B
	$2^3 \times 1/F_{CRH}$	$2^5 \times 1/F_{CRH}$	$2^7 \times 1/F_{CRH}$ to $2^{17} \times 1/F_{CRH}$
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

MB95330H Series

● 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 20.5-1 Clock Supervisor Counter Operation 1
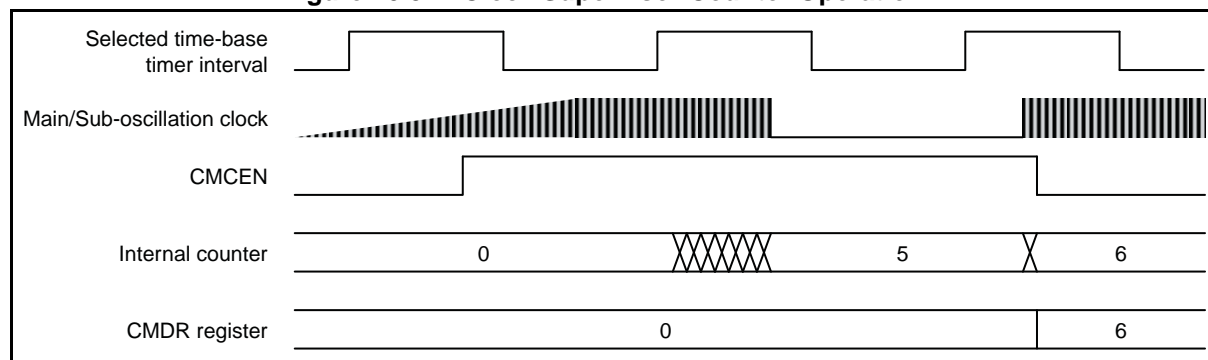
● With the clock supervisor counter running, if the external clock stops, CMCEN is set to "0" when a falling edge of the time-base timer interval selected is detected after the second rising edge of the same interval. The counter is cleared at the same falling edge.

Figure 20.5-2 Clock Supervisor Counter Operation 2
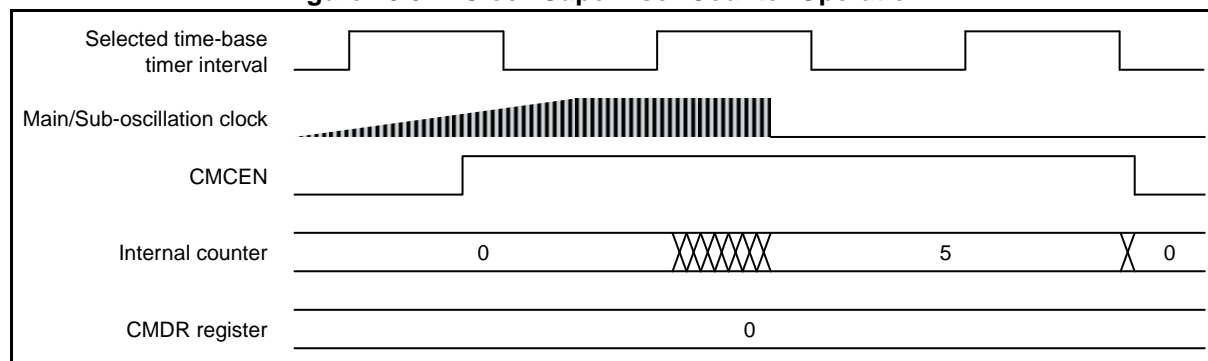
CHAPTER 21
8/16-BIT PPG

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

21.1   Overview of 8/16-bit PPG

The 8/16-bit PPG is an 8-bit reload timer module that uses pulse output control based on timer operation to perform PPG output. The 8/16-bit PPG also operates in cascade (8 bits + 8 bits) as 16-bit PPG.

■ Overview of 8/16-bit PPG

The following section summarizes the 8/16-bit PPG functions.

● 8-bit PPG output independent operation mode

In this mode, the unit can operate as two 8-bit PPG (PPG timer 00 and PPG timer 01).

● 8-bit prescaler + 8-bit PPG output operation mode

The rising and falling edge detection pulses from the PPG timer 01 output can be input to the down-counter of the PPG timer 00 to enable variable-cycle 8-bit PPG output.

● 16-bit PPG output operation mode

The unit can also operate in cascade (PPG timer 01 (upper 8 bits) + PPG timer 00 (lower 8 bits)) as 16-bit PPG output.

● PPG output operation

In this operation, a variable-cycle pulse waveform is output in any duty ratio.

The unit can also be used as a D/A converter in conjunction with an external circuit.

● Output inversion mode

This mode can invert the PPG output value.

MB95330H Series

21.2 Configuration of 8/16-bit PPG

This section shows the block diagram of the 8/16-bit PPG.

■ Block Diagram of 8/16-bit PPG

Figure 21.2-1 shows the block diagram of the 8/16-bit PPG.

Figure 21.2-1 Block Diagram of 8/16-bit PPG
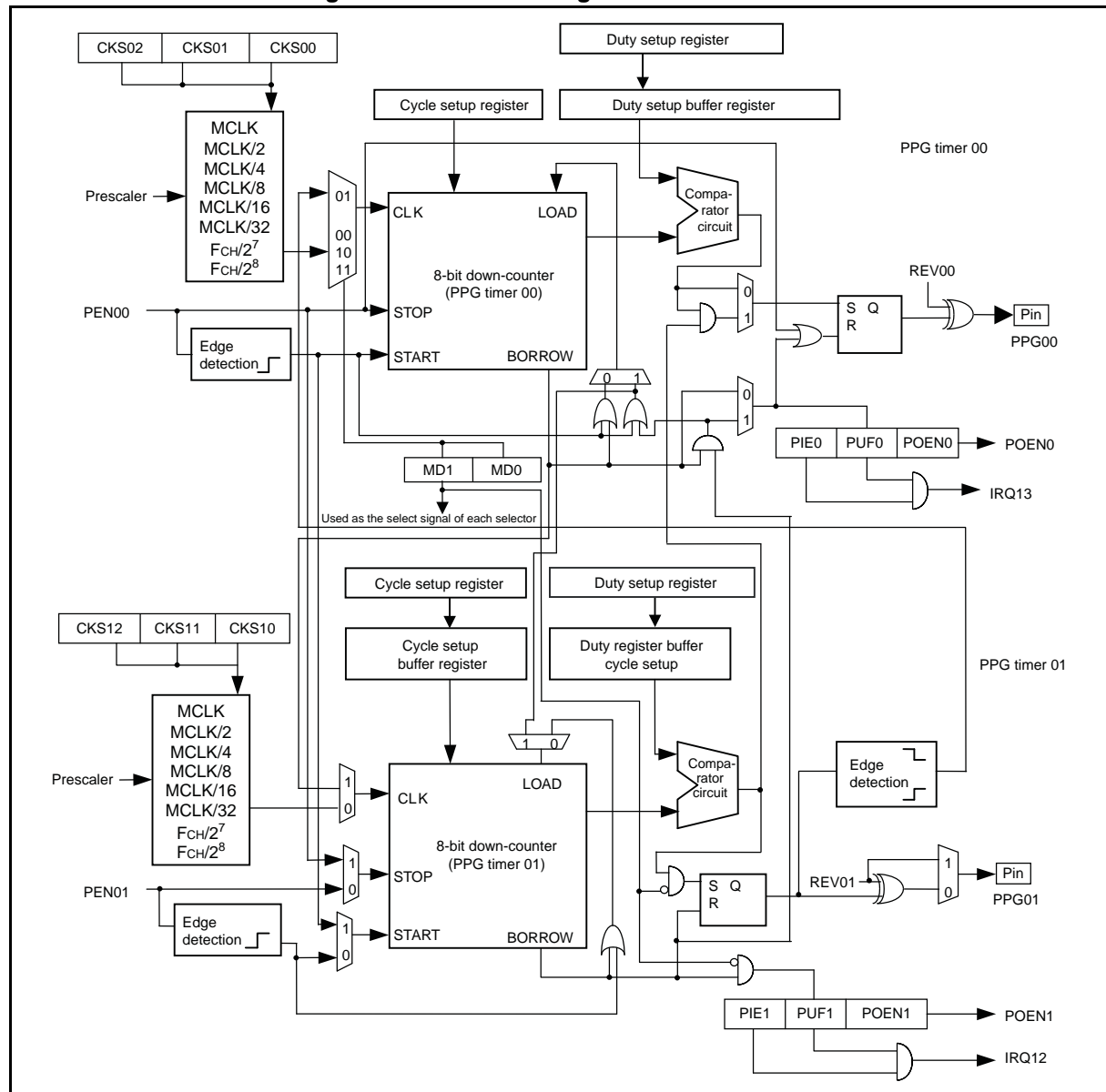
● Counter clock selector

The clock for the countdown of 8-bit down counter is selected from eight types of internal count clocks.

● 8-bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" level until the value of 8-bit down counter is corresponding to the value of 8/16-bit PPG duty setup buffer register from the value of 8/16-bit set buffer register of PPG cycle.

Afterwards, after keep "L" level the output until the counter value is corresponding to "1", it keeps counting 8-bit down counter from the value of 8/16-bit PPG cycle setup buffer register.

● 8/16-bit PPG timer 01 control register (PC01)

The operation condition on the PPG timer 01 side of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 00 control register (PC00)

The operation mode of 8/16-bit PPG timer and the operation condition on the PPG timer 00 side are set.

● 8/16-bit PPG timer 01/00 cycle setup buffer register ch.0 (PPS01), ch.0(PPS00)

The compare value for the cycle of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 01/00 duty setup buffer register ch.0 (PDS01), ch.0(PDS00)

The compare value for "H" width of 8/16-bit PPG timer is set.

● 8/16-bit PPG start register

The start or the stop of 8/16-bit PPG timer is set.

● 8/16-bit PPG output inversion register

An initial level also includes the output of 8/16-bit PPG timer and it is reversed.

■ Input Clock

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

MB95330H Series

21.3 Channels of 8/16-bit PPG

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

■ Channels of 8/16-bit PPG

The 8/16-bit PPG of the MB95330H Series has three channels, each of which consists of 8-bit PPG timer 00 and 8-bit PPG timer 01. They can be used respectively as two 8-bit PPGs or as a single 16-bit PPG.

Table 21.3-1 shows the pins of each channel and Table 21.3-2 the registers of each channel.

Table 21.3-1 Pins of 8/16-bit PPG

Channel	Pin name	Pin function
0	PPG00	PPG timer 00 (8-bit PPG (00), 16-bit PPG)
	PPG01	PPG timer 01 (8-bit PPG (01), 8-bit prescaler)
1	PPG10	PPG timer 00 (8-bit PPG (10), 16-bit PPG)
	PPG11	PPG timer 01 (8-bit PPG (11), 8-bit prescaler)
2	PPG20	PPG timer 00 (8-bit PPG (20), 16-bit PPG)
	PPG21	PPG timer 01 (8-bit PPG (21), 8-bit prescaler)

Table 21.3-2 Registers of 8/16-bit PPG

Channel	Register abbreviation	Corresponding register (Name in this manual)
0	PC01	8/16-bit PPG timer 01 control register
	PC00	8/16-bit PPG timer 00 control register
	PPS01	8/16-bit PPG timer 01 cycle setup buffer register
	PPS00	8/16-bit PPG timer 00 cycle setup buffer register
	PDS01	8/16-bit PPG timer 01 duty setup buffer register
	PDS00	8/16-bit PPG timer 00 duty setup buffer register
1	PC11	8/16-bit PPG timer 01 control register
	PC10	8/16-bit PPG timer 00 control register
	PPS11	8/16-bit PPG timer 01 cycle setup buffer register
	PPS10	8/16-bit PPG timer 00 cycle setup buffer register
	PDS11	8/16-bit PPG timer 01 duty setup buffer register
	PDS10	8/16-bit PPG timer 00 duty setup buffer register
2	PC21	8/16-bit PPG timer 01 control register
	PC20	8/16-bit PPG timer 00 control register
	PPS21	8/16-bit PPG timer 01 cycle setup buffer register
	PPS20	8/16-bit PPG timer 00 cycle setup buffer register
	PDS21	8/16-bit PPG timer 01 duty setup buffer register
	PDS20	8/16-bit PPG timer 00 duty setup buffer register
Both channels	PPGS	8/16-bit PPG start register
	REVC	8/16-bit PPG output inversion register

The following sections of this chapter provide only details of ch. 0 of the 8/16-bit PPG.

21.4　Pins of 8/16-bit PPG

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

■ Pins of 8/16-bit PPG

● PPG00 pin and PPG01 pin

These pins function both as general-purpose I/O ports and 8/16-bit PPG outputs.

PPG00, PPG01:　A PPG waveform is output to these pins. The PPG waveform can be output by enabling the output by the 8/16-bit PPG timer 01/00 control registers (PC00: POEN0 = 1, PC01: POEN1 = 1).

MB95330H Series

■ Block Diagrams of Pins of 8/16-bit PPG

Figure 21.4-1 Block Diagram of Pins PPG00, PPG10, PPG11 and PPG20 (PPG00/P13, PPG10/P10, PPG11/P11 and PPG20/P15) of 8/16-bit PPG
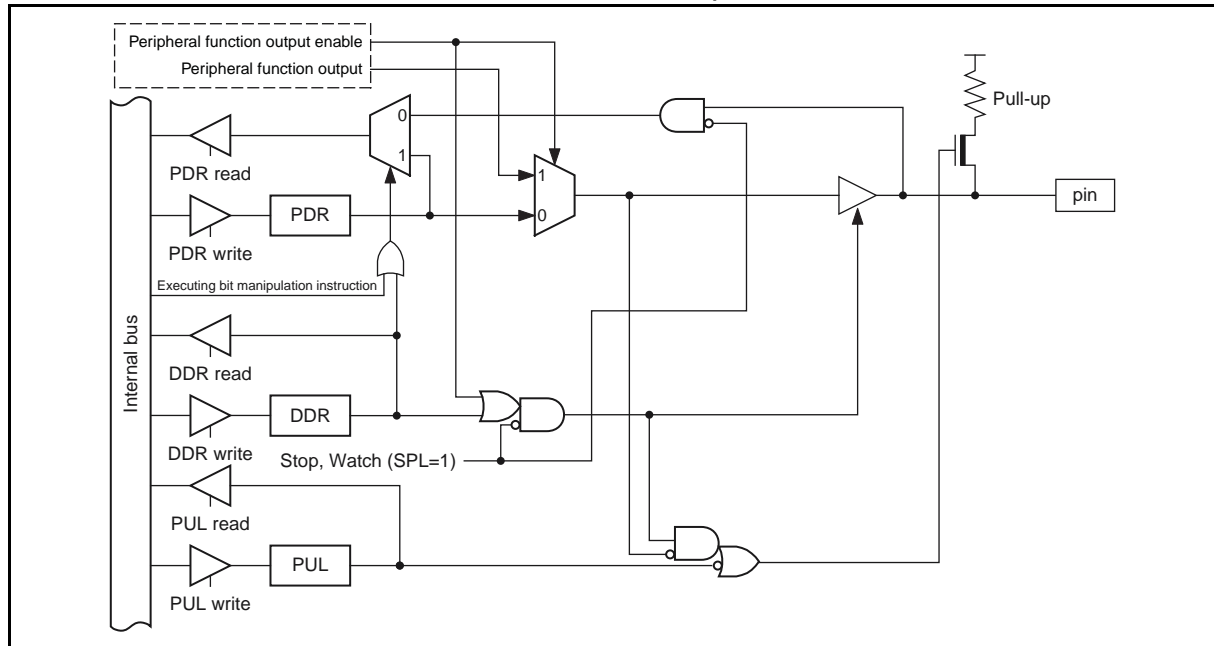
Figure 21.4-2 Block Diagram of Pin PPG01 (PPG01/P14) of 8/16-bit PPG
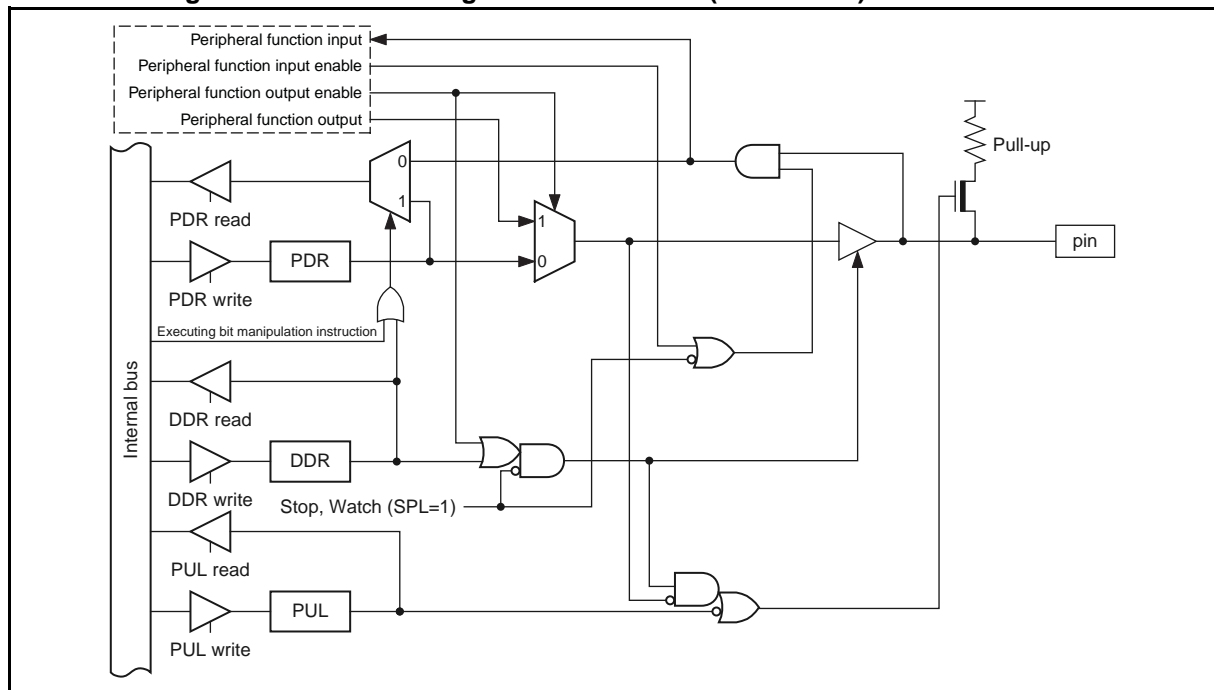
Figure 21.4-3 Block Diagram of Pin PPG21 (PPG21/P16) of 8/16-bit PPG

MB95330H Series
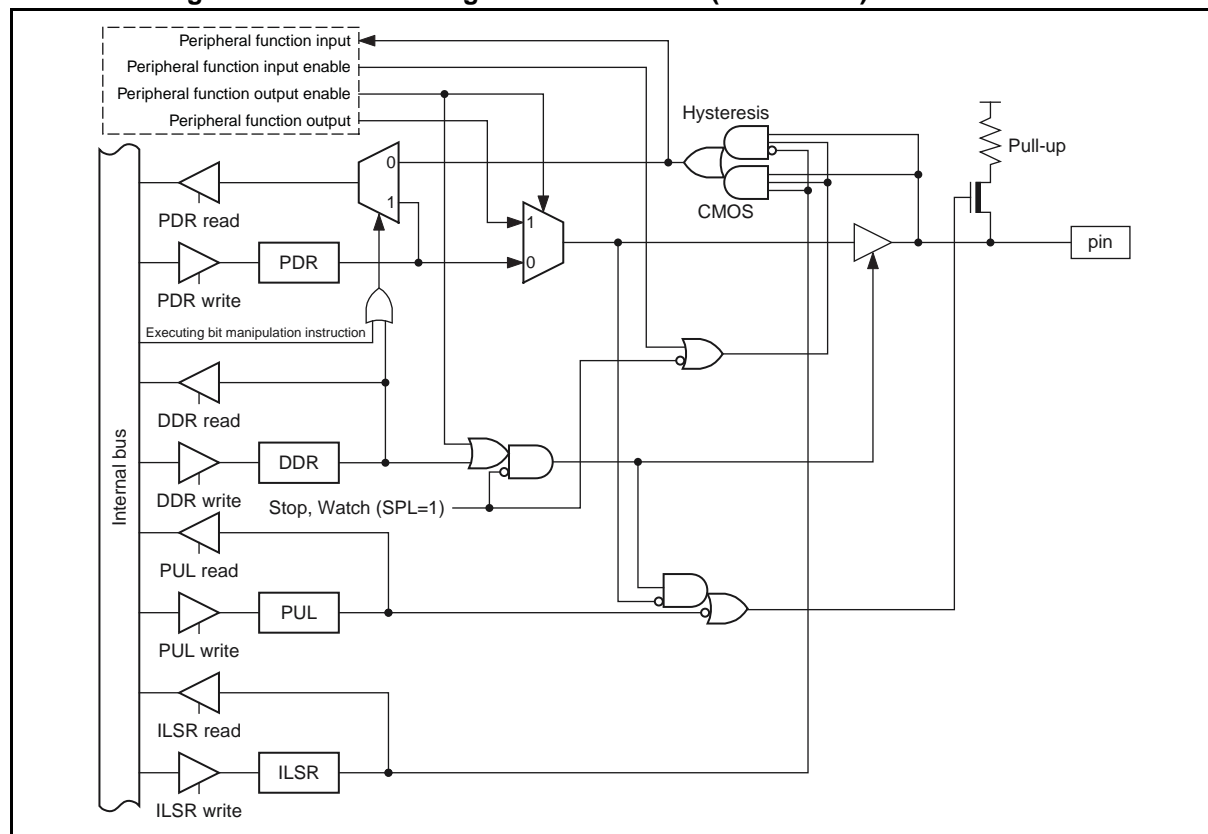
21.5 Registers of 8/16-bit PPG (ch. 0)

This section describes the registers of the 8/16-bit PPG (ch. 0).

■ Registers of 8/16-bit PPG

Figure 21.5-1 shows the registers of the 8/16-bit PPG.

Figure 21.5-1 Registers of 8/16-bit PPG

8/16-bit PPG timer 01 control register (PC01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
003A$_H$	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10	00000000$_B$
	R0/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 control register (PC00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
003B$_H$	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00	00000000$_B$
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 01 cycle setup buffer register (PPS01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F9C$_H$	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 cycle setup buffer register (PPS00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F9D$_H$	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 01 duty setup buffer register (PDS01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F9E$_H$	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG timer 00 duty setup buffer register (PDS00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F9F$_H$	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0	11111111$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG start register (PPGS)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA4$_H$	-	-	PEN21	PEN20	PEN11	PEN10	PEN01	PEN00	00000000$_B$
	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit PPG output inversion register (REVC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA5$_H$	-	-	REV21	REV20	REV11	REV10	REV01	REV00	00000000$_B$
	R0/WX	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 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.
- : Undefined bit

21.5.1 8/16-bit PPG Timer 01 Control Register ch. 0 (PC01)

The 8/16-bit PPG timer 01 control register ch. 0 (PC01) sets the operating conditions for PPG timer 01.

■ 8/16-bit PPG Timer 01 Control Register ch. 0 (PC01)

Figure 21.5-2 8/16-bit PPG Timer 01 Control Register ch. 0 (PC01)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC01 003A$_H$ PC11 003C$_H$ PC21 003E$_H$	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10	00000000$_B$
	R0/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

CKS12	CKS11	CKS10	Operating clock select bits
0	0	0	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	$F_{CH}/2^7$
1	1	1	$F_{CH}/2^8$

POEN1	Output enable bit
0	Output disabled (general-purpose port)
1	Output enabled

PUF1	Counter borrow detection flag bit for PPG cycle down-counter	
	Read	Write
0	Counter borrow not detected	Flag cleared
1	Counter borrow detected	No effect on operation

PIE1	Interrupt request enable bit
0	Interrupt disabled
1	Interrupt enabled

MCLK   : Machine clock frequency
F$_{CH}$: Machine clock oscillation frequency
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.
-               : Undefined bit
░░░   : Initial value

Table 21.5-1 8/16-bit PPG Timer 01 Control Register (PC01)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". Writing a value to it has no effect on operation.
bit5	PIE1: Interrupt request enable bit	This bit controls interrupts of PPG timer 01. **Writing "0"**: disables interrupts of PPG timer 01. **Writing "1"**: enables interrupts of PPG timer 01. The bit outputs an interrupt request (IRQ) when the counter borrow detection bit (PUF1) and the PIE1 bit are both set to "1".
bit4	PUF1: Counter borrow detection flag bit for PPG cycle down-counter	This bit serves as the counter borrow detection flag for the PPG cycle down-counter of the PPG timer 01. • This bit is set to "1" when a counter borrow occurs during 8-bit PPG mode or 8-bit prescaler mode. • In 16-bit PPG mode, this bit is not set to "1" even when a counter borrow occurs. • Writing "1" to the bit is meaningless. • Writing "0" clears the bit. • "1" is read in read-modify-write (RMW) instruction. **When the bit is set to "0"**: no counter borrow is detected. **When the bit is set to "1"**: a counter borrow is detected.
bit3	POEN1: Output enable bit	This bit enables or disables the output of PPG timer 01 pin. **Writing "0"**: the PPG timer 01 pin is used as a general-purpose port. **Writing "1"**: the PPG timer 01 pin is used as the PPG output pin. Setting this bit to "1" during 16-bit PPG operation mode sets the PPG timer 01 pin as an output. (The setting value of REV01 is output. "L" output is supplied when REV01 is "0".)
bit2 to bit0	CKS12, CKS11, CKS10: Operating clock select bits	These bits select the operating clock for 8-bit down-counter of the PPG timer 01. • The operating clock is generated from the prescaler. See CHAPTER 6 "CLOCK CONTROLLER". • In 16-bit PPG operation mode, the setting of this bit has no effect on the operation. **"000_B"**: MCLK **"001_B"**: MCLK/2 **"010_B"**: MCLK/4 **"011_B"**: MCLK/8 **"100_B"**: MCLK/16 **"101_B"**: MCLK/32 **"110_B"**: $F_{CH}/2^7$ **"111_B"**: $F_{CH}/2^8$ Note: Use of a subclock stops the time-base timer operation. Therefore, selecting "110_B" or "111_B" is prohibited.

21.5.2 8/16-bit PPG Timer 00 Control Register ch. 0 (PC00)

The 8/16-bit PPG timer 00 control register ch. 0 (PC00) sets the operating conditions and the operation mode for PPG timer 00.

■ 8/16-bit PPG Timer 00 Control Register ch. 0 (PC00)

Figure 21.5-3 8/16-bit PPG Timer 00 Control Register ch. 0 (PC00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PC00 003B$_H$	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00	00000000$_B$
PC10 003D$_H$									
PC20 003F$_H$									
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	

CKS02	CKS01	CKS00	Operating clock select bits
0	0	0	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	$F_{CH}/2^7$
1	1	1	$F_{CH}/2^8$

POEN0	Output enable bit
0	Output disabled (general-purpose port)
1	Output enabled

PUF0	Counter borrow detection flag bit for PPG cycle downcounter	
	Read	Write
0	Counter borrow not detected	Flag cleared
1	Counter borrow detected	No effect on operation

PIE0	Interrupt request enable bit
0	Interrupt disabled
1	Interrupt enabled

MD1	MD0	Operating mode select bits
0	0	8-bit PPG independent mode
0	0	8-bit prescaler + 8-bit PPG mode
1	0	16-bit PPG mode
1	1	

MCLK : Machine clock frequency
F_{CH} : Machine clock oscillation frequency
R/W : Readable/writable (The read value is the same as the write value.)
R(RM1),W : Readable/writable (The read value is different the write value. "1" is read by the read-modify-write (RMW) type of instruction.)
▨ : Initial value

Table 21.5-2 8/16-bit PPG0 Control Register (PC0)

Bit name		Function
bit7, bit6	MD1, MD0: Operation mode select bits	These bits select the PPG operation mode. Do not modify the bit settings during counting. **Writing "00$_B$":**8-bit PPG independent mode **Writing "01$_B$":**8-bit prescaler + 8-bit PPG mode **Writing "1x$_B$":**16-bit PPG mode
bit5	PIE0: Interrupt request enable bit	This bit controls interrupts of PPG timer 00. • Set this bit in 16-bit PPG operation mode. **Writing "0":** disables interrupts of PPG timer 00. **Writing "1":** enables interrupts of PPG timer 00. • An interrupt request (IRQ) is output when the counter borrow detection bit (PUF0) and PIE0 bit are both set to "1".
bit4	PUF0: Counter borrow detection flag bit for PPG cycle down-counter	This is the counter borrow detection flag for the PPG cycle down-counter of PPG timer 00. • Only this bit is effective in 16-bit PPG operation mode (PC1:PUF1 is not operable). Note: Always enable the counter borrow detection in 8-bit mode • Writing "1" to this bit is meaningless. • Writing "0" clears the bit. • "1" is read in read-modify-write (RMW) instruction. **Writing "0":** Counter borrow of PPG timer 00 not detected **Writing "1":** Counter borrow of PPG timer 00 detected
bit3	POEN0: Output enable bit	This bit enables or disables the output of PPG timer 00 pin. **Writing "0":** PPG timer 00 pin is used as a general-purpose port. **Writing "1":** PPG timer 00 pin is used as the PPG output pin. As the output is supplied from the PPG timer 00 pin in 16-bit PPG operation mode, this bit is used to control the operation.
bit2 to bit0	CKS02, CKS01, CKS00: Operating clock select bits	These bits select the operating clock for PPG down-counter PPG timer 00. • The operating clock is generated from the prescaler. See CHAPTER 6 "CLOCK CONTROLLER". • The rising and falling edge detection pulses from the PPG timer 01 output are used as the count clock for PPG timer 00 when the 8-bit prescaler + 8-bit PPG mode has been selected. Therefore, the setting of this bit has no effect on the operation. • Set this bit in 16-bit PPG operation mode. **"000$_B$":** MCLK **"001$_B$":** MCLK/2 **"010$_B$":** MCLK/4 **"011$_B$":** MCLK/8 **"100$_B$":** MCLK/16 **"101$_B$":** MCLK/32 **"110$_B$":** $F_{CH}/2^7$ **"111$_B$":** $F_{CH}/2^8$ Note: Use of a subclock stops the time-base timer operation. Therefore, selecting "110$_B$" or "111$_B$" is prohibited.

21.5.3 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

The 8/16-bit PPG timer 00/01 cycle setup buffer register (PPS01), (PPS00) sets the PPG output cycle.

■ 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

Figure 21.5-4 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PPS00	0F9C$_H$	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	11111111$_B$
PPS11	0FA0$_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PPS21	0FA6$_H$									
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PPS00	0F9D$_H$	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	11111111$_B$
PPS10	0FA1$_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PPS20	0FA7$_H$									

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

This register is used to set the PPG output cycle.

- In 16-bit PPG mode, PPS01 serves as the upper 8 bits, while PPS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load.
- 8-bit mode: Cycle = max. 255 (FF$_H$) × Input clock cycle
- 16-bit mode: Cycle = max. 65535 (FFFF$_H$) × Input clock cycle
- Initialized at reset.
- Do not set the cycle to "00$_H$" or "01$_H$" when using the unit in 8-bit PPG independent mode, or in 8-bit prescaler mode + 8-bit PPG mode
- Do not set the cycle to "0000$_H$" or "0001$_H$" when using the unit in 16-bit PPG mode.
- If the cycle settings are modified during the operation, the modified settings will be effective from the next PPG cycle.

MB95330H Series

21.5.4   8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

The 8/16-bit PPG timer 00/01 duty setup buffer register (PDS01), (PDS00) sets the duty of the PPG output.

■ 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

Figure 21.5-5 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDS01	0F9E$_H$	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	11111111$_B$
PDS11	0FA2$_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDS21	0FAA$_H$									
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDS00	0F9F$_H$	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0	11111111$_B$
PDS10	0FA3$_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDS20	0FAB$_H$									

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

This register is used to set the duty of the PPG output ("H" pulse width when normal polarity).

- In 16-bit PPG mode, PDS01 serves as the upper 8 bits while PDS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load. By writing to PDS00, PDS01 is updated.
- Initialized at reset.
- To set the duty to 0%, select "00$_H$".
- To set the duty to 100%, set it to the same value as the 8/16-bit PPG timer 00/01 cycle setup register (PPS00, PPS01).
- When the 8/16-bit PPG timer 00/01 duty setup register (PDS) is set to a larger value than the setting value of the 8/16-bit PPG cycle setup buffer register (PPS), the PPG output becomes "L" output in the normal polarity (when the output level inversion bit of 8/16-bit PPG output inversion register is "0").
- If the duty settings are modified during operation, the modified value will be effective from the next PPG cycle.

21.5.5   8/16-bit PPG Start Register (PPGS)

The 8/16-bit PPG start register (PPGS) starts or stops the down-counter. The operation enable bit of each channel is assigned to the PPGS register, allowing simultaneous activation of the PPG channels.

■ 8/16-bit PPG Start Register (PPGS)

Figure 21.5-6 8/16-bit PPG Start Register (PPGS)

MB95330H Series
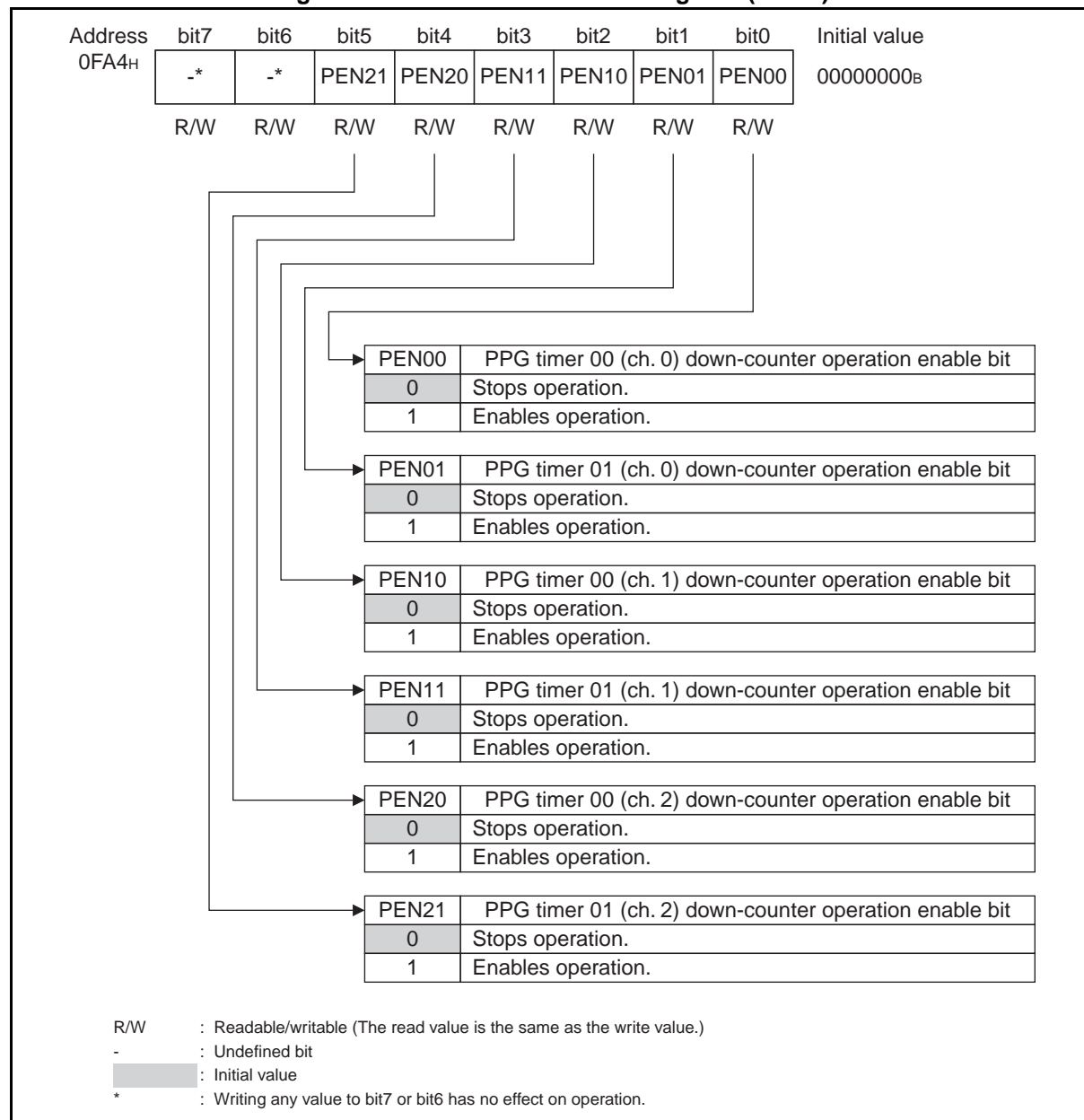
21.5.6   8/16-bit PPG Output Reverse Register (REVC)

The 8/16-bit PPG output inversion register (REVC) reverses the PPG output including the initial level.

■ **8/16-bit PPG Output Reverse Register (REVC)**

Figure 21.5-7 8/16-bit PPG Output Reverse Register (REVC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA5H	-*	-*	REV21	REV20	REV11	REV10	REV01	REV00	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

REV00	PPG timer 00 (ch. 0) output level reverse bit
0	Normal
1	Reverse

REV01	PPG timer 01 (ch. 0) output level reverse bit
0	Normal
1	Reverse

REV10	PPG timer 00 (ch. 1) output level reverse bit
0	Normal
1	Reverse

REV11	PPG timer 01 (ch. 1) output level reverse bit
0	Normal
1	Reverse

REV20	PPG timer 00 (ch. 2) output level reverse bit
0	Normal
1	Reverse

REV21	PPG timer 01 (ch. 2) output level reverse bit
0	Normal
1	Reverse

R/W   : Readable/writable (The read value is the same as the write value.)
-       : Undefined bit
▨   : Initial value
*   : Writing any value to bit7 or bit6 has no effect on operation.

21.6   Interrupts of 8/16-bit PPG

The 8/16-bit PPG outputs an interrupt request when a counter borrow is detected.

■ **Interrupts of 8/16-bit PPG**

Table 21.6-1 shows the interrupt control bits and interrupt sources of the 8/16-bit PPG.

Table 21.6-1 Interrupt Control Bits and Interrupt Sources of 8/16-bit PPG

Item	Description	
	PPG timer 01 (8-bit PPG, 8-bit prescaler)	PPG timer 00 (8-bit PPG, 16-bit PPG)
Interrupt request flag bit	PUF1 bit in PC01	PUF0 bit in PC00
Interrupt request enable bit	PIE1 bit in PC01	PIE0 bit in PC00
Interrupt source	Counter borrow of PPG cycle down-counter	

When a counter borrow occurs on the down-counter, the 8/16-bit PPG sets the counter borrow detection flag bit (PUF) in the 8/16-bit PPG timer 00/01 control register (PC) to "1". When the interrupt request enable bit is enabled (PIE = 1), an interrupt request is output to the interrupt controller.

In 16-bit PPG mode, the 8/16-bit PPG timer 00 control register (PC00) is available.

■ **Registers and Vector Table Addresses Related to Interrupts of 8/16-bit PPG**

Table 21.6-2 Registers and Vector Table Addresses Related to Interrupts of 8/16-bit PPG

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
8/16-bit PPG ch. 0 (lower)	IRQ13	ILR3	L13	FFE2$_H$	FFE3$_H$
8/16-bit PPG ch. 0 (upper)	IRQ12	ILR3	L12	FFE0$_H$	FFE1$_H$
8/16-bit PPG ch. 1 (lower)	IRQ09	ILR2	L09	FFE8$_H$	FFE9$_H$
8/16-bit PPG ch. 1 (upper)	IRQ10	ILR2	L10	FFE6$_H$	FFE7$_H$
8/16-bit PPG ch. 2 (lower)	IRQ15	ILR3	L15	FFDC$_H$	FFDD$_H$
8/16-bit PPG ch. 2 (upper)	IRQ11	ILR2	L11	FFE4$_H$	FFE5$_H$

ch.: Channel

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

21.7   Operations of 8/16-bit PPG and Setting Procedure Example

This section describes the operations of the 8/16-bit PPG.

■ Setting Procedure Example

Below is an example of procedure for setting the 8/16-bit PPG ch. 0.

● Initial setup

1) Set the port output (DDR1)

2) Set the interrupt revel (ILR3)

3) Select the operating clock, enable the output and interrupt (PC01)

4) Select the operating clock, enable the output and interrupt, select the operation mode (PC00)

5) Set the cycle (PPS)

6) Set the duty (PDS)

7) Set the output inversion (REVC)

8) Start PPG (PPGS)

● Interrupt processing

1) Process any interrupt

2) Clear the interrupt request flag (PC01: PUF1, PC00: PUF0)

3) Start PPG (PPGS)

21.7.1   8-bit PPG Independent Mode

In this mode, the unit operates as two channels (PPG timer 00 and PPG timer 01) of the 8-bit PPG.

■ Setting 8-bit PPG Independent Mode

The unit requires the register settings shown in Figure 21.7-1 to operate in 8-bit PPG independent mode.

Figure 21.7-1 8-bit PPG Independent Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0	◎	◎	◎	◎	◎	◎
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
				Set PPG output cycle for PPG timer 01				
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
				Set PPG output cycle for PPG timer 00				
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
				Set PPG output duty for PPG timer 01				
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
				Set PPG output duty for PPG timer 00				
PPGS	-	-	PEN21	PEN20	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	◎	◎
REVC	-	-	REV21	REV20	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	◎	◎

◎ : Used bit
0 : Set to "0"
* : The bit status depends on the number of channels provided.

■ Operation of 8-bit PPG Independent Mode

- This mode is selected when the operation mode select bits (MD1, MD0) in the 8/16-bit PPG timer 00 control register (PC00) are set to "00$_B$".

- When the corresponding bit (PEN) in the 8/16-bit PPG start register (PPGS) is set to "1", the value in the 8/16-bit PPG cycle setup buffer register (PPS) is loaded to start down-count operation. When the count value reaches "1", the value in the cycle setup register is reloaded to repeat the counting.

- "H" is output to the PPG output synchronizing with the count clock. When the down-counter value matches the value in the 8/16-bit PPG timer 00/01 duty setup buffer register (PDS). After "H" which is the value of duty setting is output, "L" is output to the PPG output.

If, however, the PPG output inversion bit is set to "1", the PPG output is set and reset inversely from the above process.

Figure 21.7-2 shows the operation of the 8-bit PPG independent mode.

Figure 21.7-2 Operation of 8-bit PPG Independent Mode
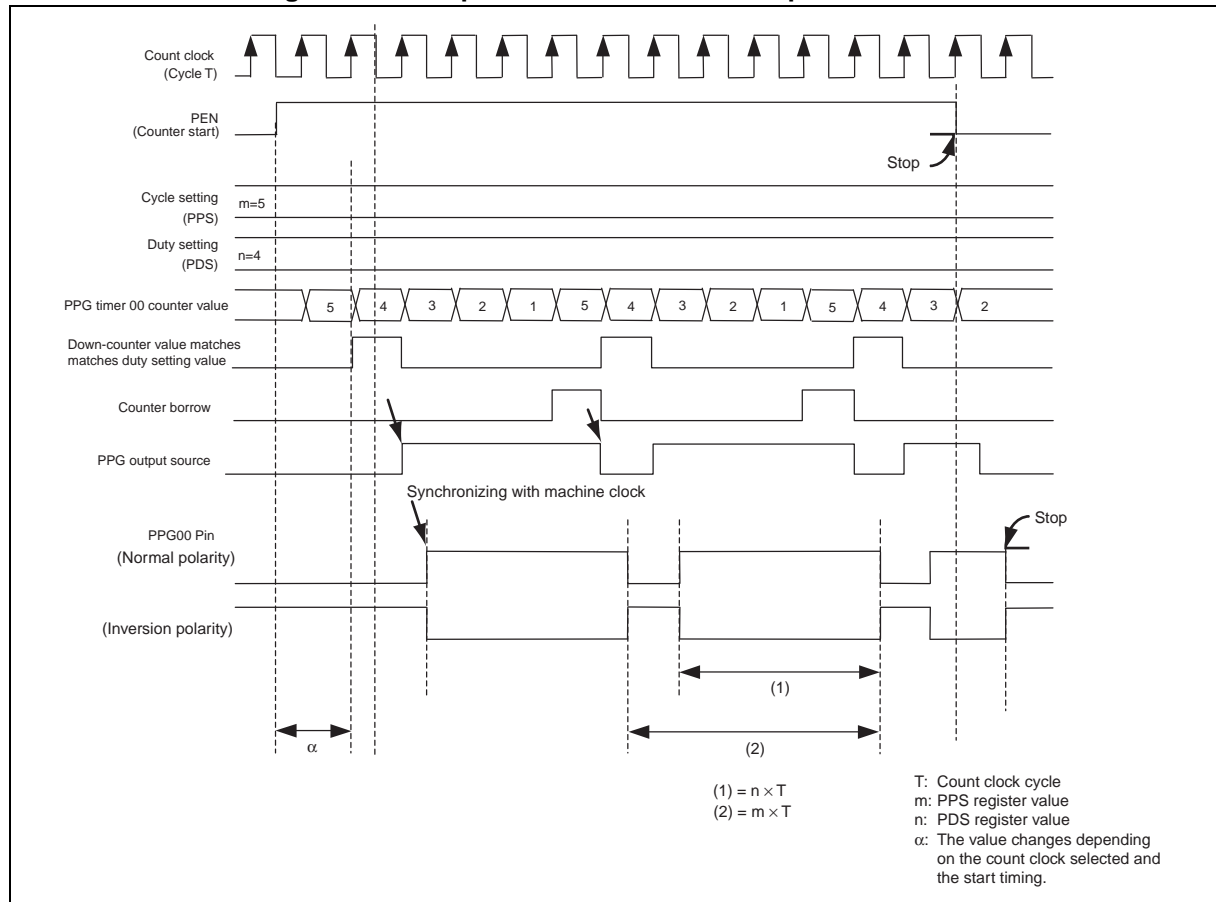
Example for setting the duty to 50%

When PDS is set to "02$_H$" with PPS set to "04$_H$", the PPG output is set at a duty ratio of 50% (PPS setting value /2 set to PDS).

21.7.2   8-bit Prescaler + 8-bit PPG Mode

In this mode, the rising and falling edge detection pulses from the PPG timer 01 output can be used as the count clock of the PPG timer 00 down-counter to allow variable-cycle 8-bit PPG output from PPG timer 00.

■ Setting 8-bit Prescaler + 8-bit PPG Mode

The unit requires the register settings shown in Figure 21.7-3 to operate in 8-bit prescaler + 8-bit PPG mode.

Figure 21.7-3 Setting 8-bit Prescaler + 8-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	1	◎	◎	◎	×	×	×
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty for PPG timer 00							
PPGS	-	-	PEN21	PEN20	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	◎	◎
REVC	-	-	REV21	REV20	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	◎	◎

◎ : Used bit
0 : Set to "0"
1 : Set to "1"
× : Setting nullified
* : The bit status varies depending of the number of channels implemented

■ Operation of 8-bit Prescaler + 8-bit PPG Mode

- This mode is selected by setting the operation mode select bits (MD1, MD0) of the 8/16-bit PPG timer 00 control register (PC00) to "01$_\text{B}$". This allows PPG timer 01 to be used as an 8-bit prescaler and PPG timer 00 to be used as an 8-bit PPG.

- When the PPG timer 00 (ch.1) down counter operation enable bit (PEN01) is set to "1", the 8-bit prescaler (PPG timer 01) loads the value in the 8/16-bit PPG timer 01 cycle setup buffer register (PPS01) and starts down-count operation. When the value of the down-counter matches the value in the 8/16-bit PPG timer 01 duty setup buffer register (PDS01), the PPG01 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG01 output is set to "L". If the output inversion signal (REV01) is "0", the polarity will remain the same. If it is "1", the polarity will be inverted

and the signal will be output to the PPG pin.

- When the PPG operation enable bit (PEN00) is set to "1", the 8-bit PPG (PPG timer 00) loads the value in the 8/16-bit PPG timer 00 cycle setup buffer register (PPS00) and starts down-count operation (count clock = rising and falling edge detection pulses of PPG01 output after PPG timer 01 operation is enabled). When the count value reaches "1", the value in the 8/16-bit PPG timer 00 cycle setup buffer register is reloaded to repeat the counting. When the value of the down-counter matches the value in the 8/16-bit PPG timer 00 duty setup buffer register (PDS00), the PPG00 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 output is reset to "L". If the output inversion signal (REV00) is "0", the polarity will remain the same. If it is "1", the polarity will be inverted and the signal will be output to the PPG00 pin.

- Set that the duty of the 8-bit prescaler (PPG timer 01) output to 50%.

- When PPG timer 00 is started with the 8-bit prescaler (PPG timer 01) being stopped, PPG timer 00 does not count.

- When the duty of the 8-bit prescaler (PPG timer 01) is set to 0% or 100%, PPG timer 00 does not perform counting as the 8-bit prescaler (PPG timer 01) output does not toggle.

Figure 21.7-4 shows the operation of 8-bit prescaler + 8-bit PPG mode.

Figure 21.7-4 Operation of 8-bit Prescaler + 8-bit PPG Mode

(1) = n1 × T
(2) = m1 × T
(3) = (1) × n0
(4) = (1) × m0

T: Count clock cycle
m0: PPS00 register value
n0: PDS00 register value
m1: PPS01 register value
n1: PDS01 register value

α: The value changes depending on the count clock selected and the PEN01 start timing.
β: The value changes depending on the PPG01 output (ch.1) waveform and the PEN00 start timing.
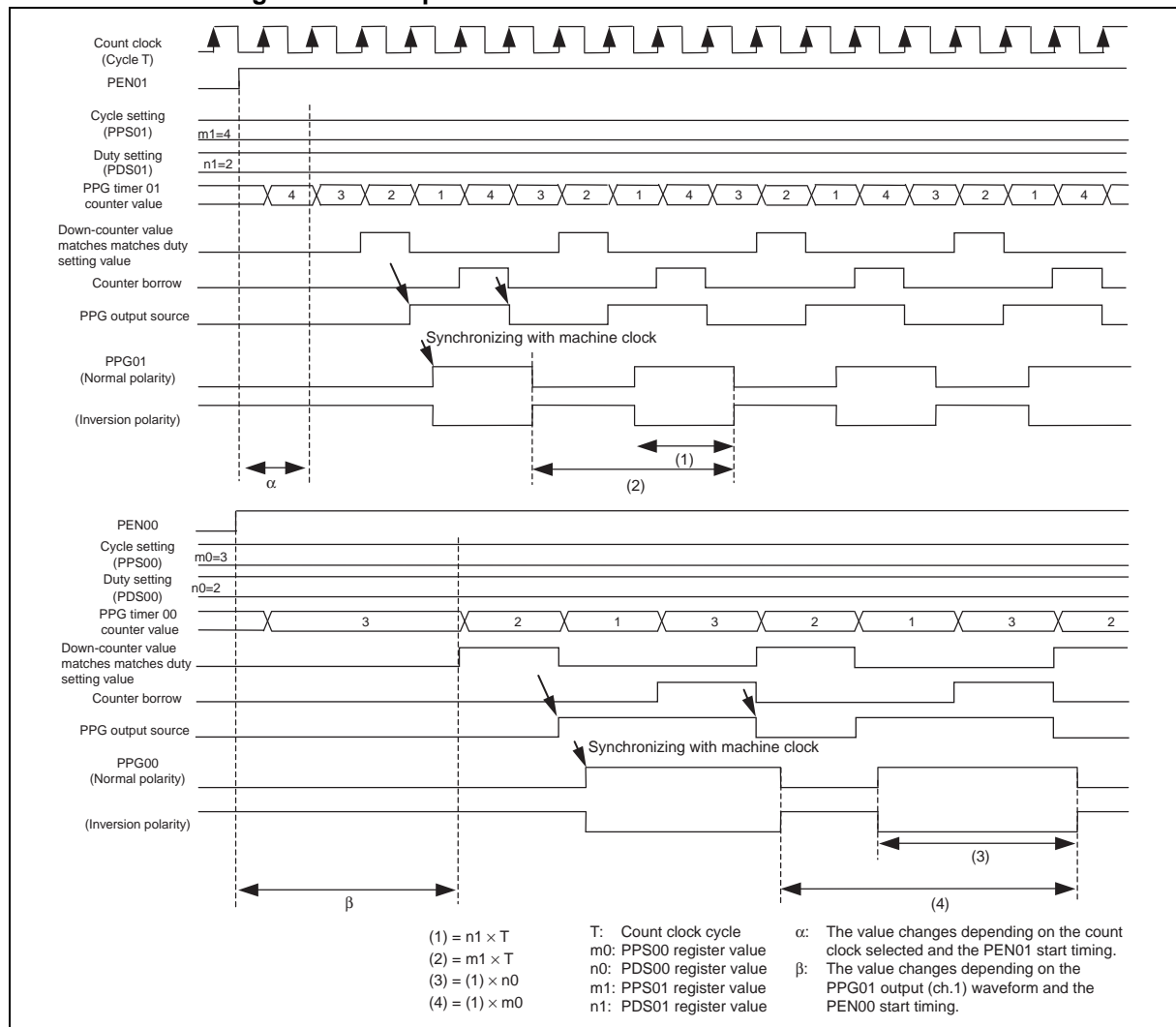
21.7.3   16-bit PPG Mode

In this mode, the unit can operate as a 16-bit PPG when PPG timer 01 and PPG timer 00 are assigned to the upper and lower bits respectively.

■ Setting 16-bit PPG Mode

The unit requires the register settings shown in Figure 21.7-5 to operate in 16-bit PPG mode.

Figure 21.7-5 Setting 16-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			◎	◎	◎	◎	◎	◎
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0/1	◎	◎	◎	◎	◎	◎
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle (Upper 8 bits) for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle (Lower 8 bits) for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty (Upper 8 bits) for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty (Lower 8 bits) for PPG timer 00							
PPGS	-	-	PEN21	PEN20	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	×	◎
REVC	-	-	REV21	REV20	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	×	◎

◎ : Used bit
0 : Set to "0"
1 : Set to "1"
× : Setting nullified
* : The bit status changes depending on the number of channels implemented.

■ Operation of 16-bit PPG Mode

- This mode is selected by setting the operation mode select bits (MD1, MD0) of the PPG timer 00 control register (PC00) to "10_B" or "11_B".

- When the PPG operation enable bit (PEN00) is set to "1" in 16-bit PPG mode, the 8-bit down-counters (PPG timer 00) and 8-bit down-counter (PPG timer 01) load the values in the 8/16-bit PPG timer 00/01 cycle setup buffer registers (PPS01 for PPG timer 01 and PPS00 for PPG timer 00) and start down-count operation. When the count value reaches "1", the values in the cycle setup register are reloaded and the counters repeat the counting.

- When the values of the down-counters match the values in the 8/16-bit PPG timer duty setup buffer registers (both the value in PDS01 for PPG timer 01 and the value in PDS00 for PPG timer 00), the PPG00 pin is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 pin is set to "L". If the output inversion signal (REV00) is "0", the signal will be output to the PPG00 with the polarity unchanged. If it is set to "1", the polarity will be inverted and the signal will be output to the PPG00 pin. (ch.0 only. ch.1 will be set to the initial value <"L" if REV01 is "0", or "H" if it is "1">.)

Figure 21.7-6 shows the operation of 16-bit PPG mode.

Figure 21.7-6 Operation of 16-bit PPG Mode
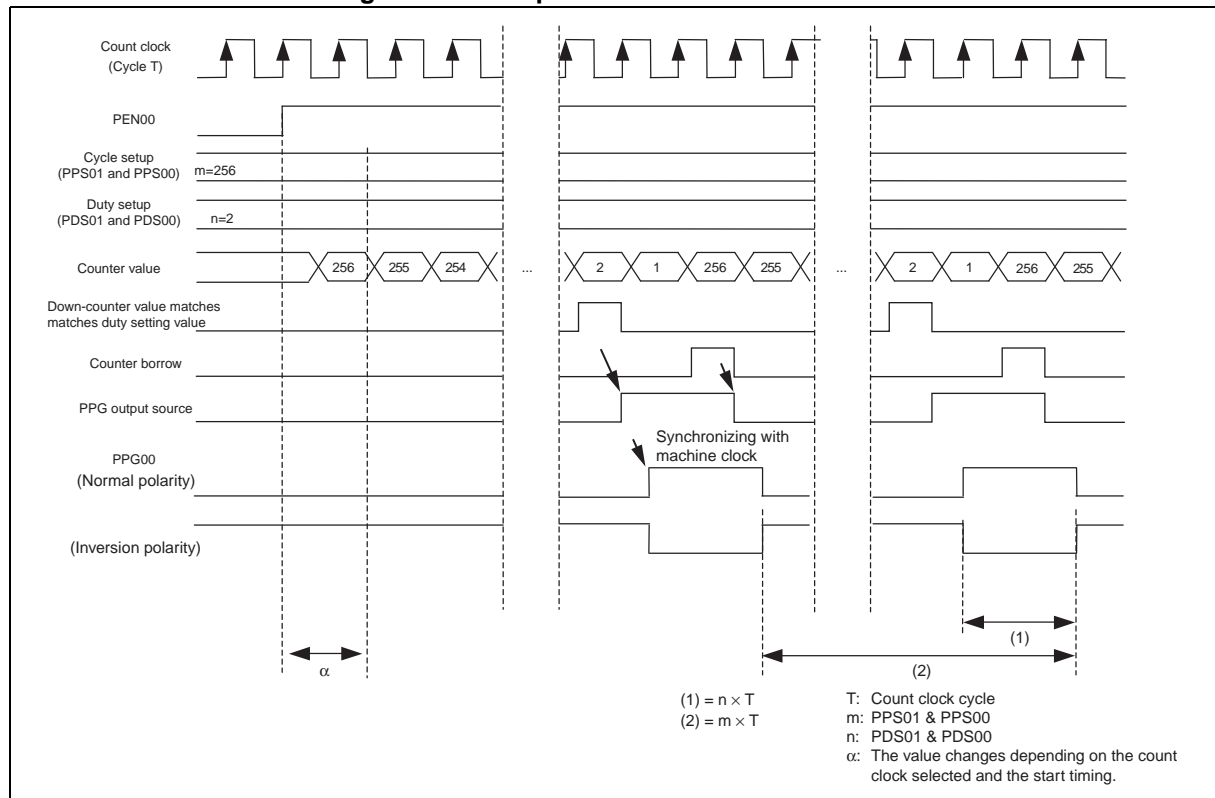
■ Setting Procedure Example

Below is an example of procedure for setting the 8/16-bit PPG ch. 0.

● Initial setup

1) Set the port output (DDR1)

2) Set the interrupt revel (ILR3)

3) Select the operating clock, enable the output and interrupt (PC01)

4) Select the operating clock, enable the output and interrupt, select the operation mode (PC00)

5) Set the cycle (PPS)

6) Set the duty (PDS)

7) Set the output inversion (REVC)

8) Start PPG (PPGS)

● Interrupt processing

1) Process any interrupt

2) Clear the interrupt request flag (PC01: PUF1, PC00: PUF0)

3) Start PPG (PPGS)

MB95330H Series

21.8 Notes on Using 8/16-bit PPG

This section provides notes on using the 8/16-bit PPG.

■ **Notes on Using 8/16-bit PPG**

● Note on operation

Depending on the timing between the activation of PPG and count clock, an error may occur in the first cycle of the PPG output immediately after the activation. The error varies depending on the count clock selected. The output, however, is performed properly in the succeeding cycles.

● Note on interrupts

A PPG interrupt is generated when the interrupt enable bit (PIE1/PIE0) is set to "1" and the interrupt request flag bit (PUF1/PUF0) in the 8/16-bit PPG timer 01/00 control register (PC01/PC00) is also set to "1". Always clear the interrupt request flag bit (PUF1/PUF0) to "0" in the interrupt routine.

21.9 Sample Settings for 8/16-bit PPG

This section provides sample settings for the 8/16-bit PPG.

■ Sample Settings

● How to enable/stop PPG operation

The PPG operation enable bit (PPGS:PEN00, PEN10 or PEN20) is used for PPG timer 00.

Operation	PPG operation enable bit (PEN00, PEN10 or PEN20)
To stop PPG operation	Set the bit to "0".
To enable PPG operation	Set the bit to "1".

PPG operation must be enabled before the PPG is activated.

The PPG operation enable bit (PPGS:PEN01, PEN11 or PEN21) is used for PPG timer 01.

Operation	PPG operation enable bit (PEN01, PEN11 or PEN21)
To stop PPG operation	Set the bit to "0".
To enable PPG operation	Set the bit to "1".

PPG operation must be enabled before the PPG is activated.

● How to set the PPG operation mode

The operation mode select bits (PC00.MD[1:0]) are used.

● How to select the operating clock

ch.1 is selected by the operating clock select bits (PC01:CKS12/CKS11/CKS10).

ch.0 is selected by the operating clock select bits (PC00:CKS02/CKS01/CKS00).

● How to enable/disable the PPG output pin

The output enable bit (PC00:POEN0 or PC01:POEN1) is used.

Operation	Output enable bit (POEN0 or POEN1)
To enable PPG output	Set the bit to "1".
To disable PPG output	Set the bit to "0".

● How to reverse the PPG output

The output level reverse bit (REVC:REV00 or REV10 or REV20) is used for PPG timer 00.

Operation	Output level reverse bit (REV00 or REV10 or REV20)
To reverse PPG output	Set the bit to "1".

The output level reverse bit (REVC:REV01 or REV11 or REV21) is used for PPG timer 01.

Operation	Output level reverse bit (REV01 or REV11 or REV21)
To reverse PPG output	Set the bit to "1".

● Interrupt-related register

The interrupt level is set by the interrupt setup register shown in the following table.

Interrupt source	Interrupt level setup register	Interrupt vector
ch. 0 (lower)	Interrupt level register (ILR3) Address:$0007C_H$	#12 Address:$0FFE2_H$
ch. 0 (upper)	Interrupt level register (ILR3) Address:$0007C_H$	#13 Address:$0FFE0_H$
ch. 1 (lower)	Interrupt level register (ILR2) Address:$0007B_H$	#09 Address:$0FFE8_H$
ch. 1 (upper)	Interrupt level register (ILR2) Address:$0007B_H$	#10 Address:$0FFE6_H$
ch. 2 (lower)	Interrupt level register (ILR3) Address:$0007C_H$	#15 Address:$0FFDC_H$
ch. 2 (upper)	Interrupt level register (ILR2) Address:$0007B_H$	#11 Address:$0FFE4_H$

● How to enable/disable/clear interrupts

Interrupt request enable flag, Interrupt request flag

The interrupt request enable bit (PC00:PIE0 or PC01:PIE1) is used to enable or disable interrupts.

Operation	Interrupt request enable bit (PIE0 or PIE1)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The interrupt request flag (PC00:PUF0 or PC01:PUF1) is used to clear an interrupt request.

Operation	Interrupt request flag (PUF0 or PUF1)
To clear an interrupt request	Set the bit to "0".

CHAPTER 22
16-BIT PPG TIMER

This chapter describes the functions and operations of the 16-bit PPG timer.

22.1 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 $\leftrightarrow$ Inverted polarity).
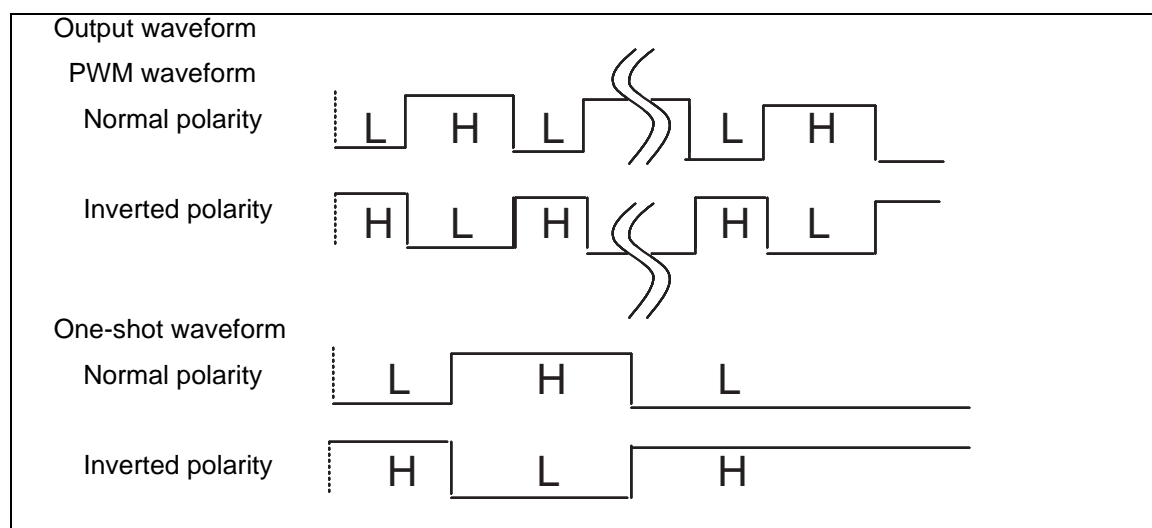
- The count operation clock can be selected from eight different clock sources (MCLK/1, MCLK/2, MCLK/4, MCLK/8, MCLK/16, MCLK/32, $F_{CH}/2^7$, or $F_{CH}/2^8$). (MCLK: Machine clock, F_{CH}: Main clock)

- 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

MB95330H Series

22.2   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 22.2-1 Block Diagram of 16-bit PPG Timer
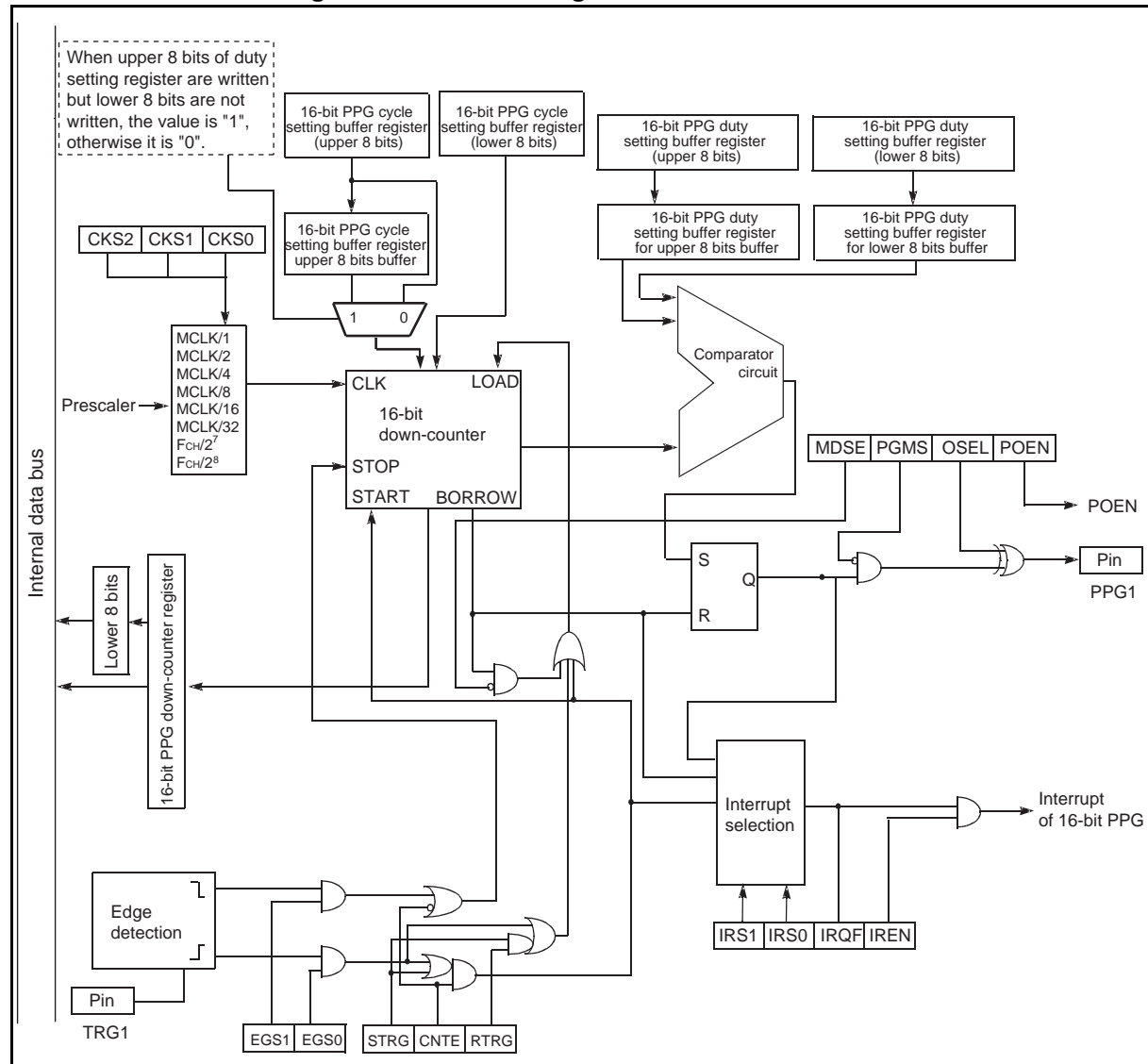
● Count clock selector

The clock for the countdown of 16-bit down-counter is selected from eight types of internal count clocks.

● 16 bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" until the value of 16-bit down-counter is corresponding to the value of 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.

● 16-bit PPG down-counter register upper, lower (PDCRH1, PDCRL1)

The value of 16-bit down-counter of 16-bit PPG timer is read.

● 16-bit PPG cycle setting buffer register upper, lower (PCSRH1, PCSRL1)

The compare value for the cycle of 16-bit PPG timer is set.

● 16-bit PPG duty setting buffer register upper, lower (PDUTH1, PDUTL1)

The compare value for "H" width of 16-bit PPG timer is set.

● 16-bit PPG status control register upper, lower (PCNTH1, PCNTL1)

The operation mode and the operation condition of 16-bit PPG timer are set.

■ Input Clock

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

MB95330H Series

22.3   Channel of 16-bit PPG Timer

This section describes the channel of the 16-bit PPG timer.

■ Channel of 16-bit PPG Timer

The MB95330H Series has one 16-bit PPG timer.

Table 22.3-1 and Table 22.3-2 show the pins and registers of the 16-bit PPG timer respectively.

Table 22.3-1 Pins of 16-bit PPG Timer

Channel	Pin name	Pin function
1	PPG1	PPG1 output
	TRG1	Trigger 1 input

Table 22.3-2 Registers of 16-bit PPG Timer

Channel	Register abbreviation	Corresponding register (Name in this manual)
1	PDCRH1	16-bit PPG down-counter register (upper)
	PDCRL1	16-bit PPG down-counter register (lower)
	PCSRH1	16-bit PPG cycle setting buffer register (upper)
	PCSRL1	16-bit PPG cycle setting buffer register (lower)
	PDUTH1	16-bit PPG duty setting buffer register (upper)
	PDUTL1	16-bit PPG duty setting buffer register (lower)
	PCNTH1	16-bit PPG status control register (upper)
	PCNTL1	16-bit PPG status control register (lower)

22.4　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 PPG1 pin and TRG1 pin.

● PPG1 pin

This pin serves as a general-purpose I/O port as well as a 16-bit PPG timer output.

PPG1: 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 (PCNTL1: POEN=1).

● TRG1 pin

TRG1: Used to start the 16-bit PPG timer by the hardware trigger.

MB95330H Series

■ Block Diagrams of Pins of 16-bit PPG Timer

Figure 22.4-1 Block Diagram of Pin PPG1 (P66/PPG20/PPG1/OPT4) of 16-bit PPG
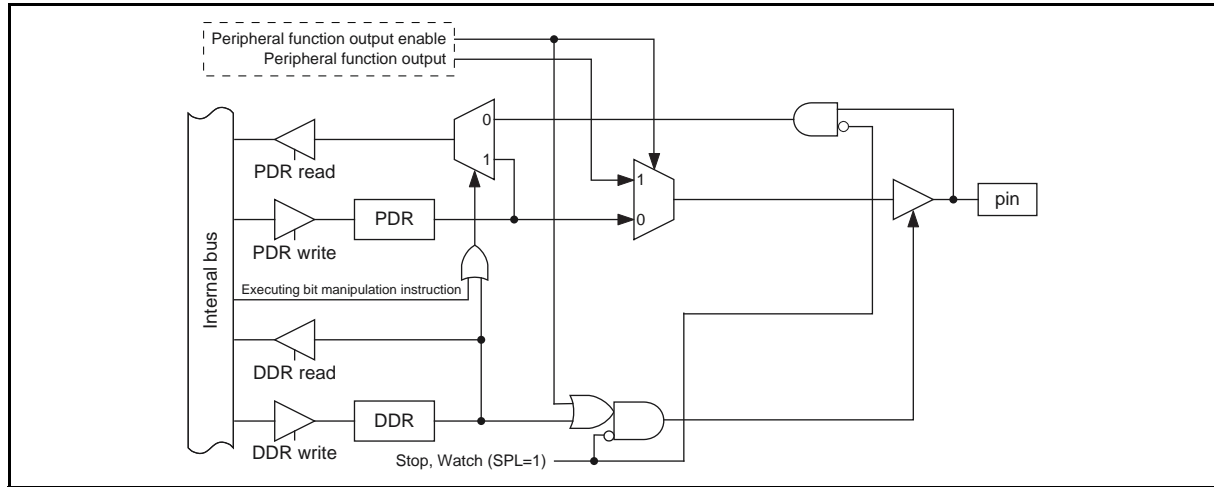
Figure 22.4-2 Block Diagram of Pin TRG1 (P67/PPG21/TRG1/OPT5) of 16-bit PPG
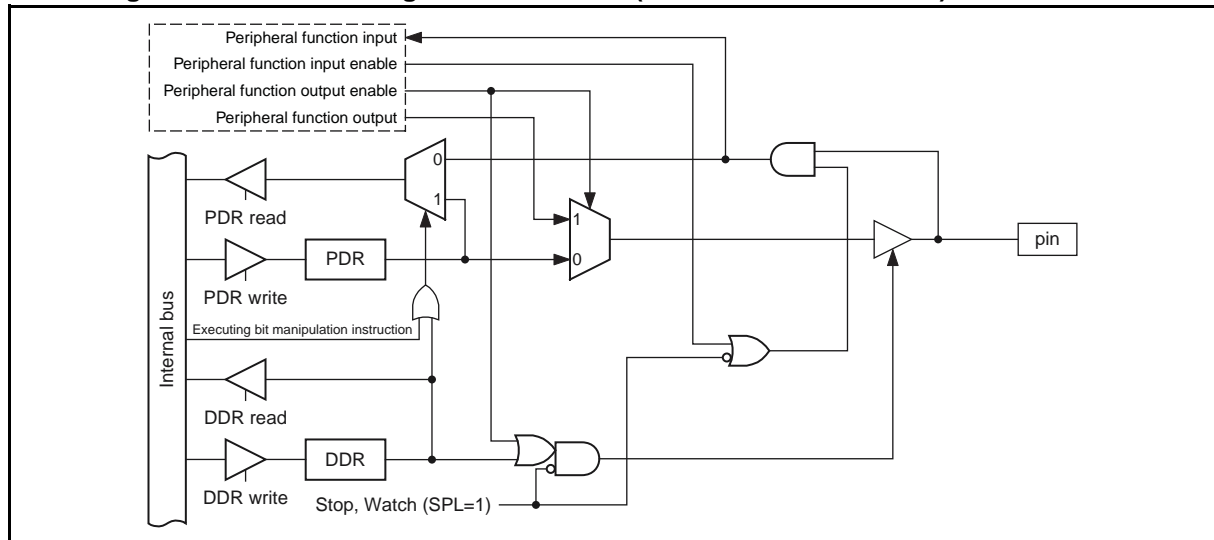
22.5 Registers of 16-bit PPG Timer

This section describes the registers of the 16-bit PPG timer.

■ Registers of 16-bit PPG Timer

Figure 22.5-1 Registers of 16-bit PPG Timer

16-bit PPG down-counter register (upper) (PDCRH1)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FB0H	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down-counter register (lower) (PDCRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB1H	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000B
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG cycle setting buffer register (upper) (PCSRH1)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FB2H	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG cycle setting buffer register (lower) (PCSRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB3H	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (upper) (PDUTH1)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FB4H	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG duty setting buffer register (lower) (PDUTL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB5H	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111B
Address	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit PPG status control register (upper) (PCNTH1)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0044H	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	

16-bit PPG status control register (lower) (PCNTL1)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	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	

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".)

MB95330H Series

22.5.1 16- bit PPG Down-counter Registers Upper, Lower (PDCRH1, PDCRL1)

The 16-bit PPG down-counter registers upper, lower (PDCRH1, PDCRL1) 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 (PDCRH1, PDCRL1)**

Figure 22.5-2 16-bit PPG Down-counter Registers Upper, Lower (PDCRH1, PDCRL1)

16-bit PPG down-counter register (upper) (PDCRH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB0$_H$	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) (PDCRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB1$_H$	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 PDCRH1 register address).
- Use the "MOV" instruction and read PDCRH1 first and then PDCRL1 (reading PDCRH1 automatically copies the lower 8 bits of the down-counter to PDCRL1).

These registers are read-only and writing has no effect on the operation.

Note:

If you use the "MOV" instruction and read PDCRL1 before PDCRH1, PDCRL1 will return the value from the previous valid read operation. Therefore, the value of the 16-bit down-counter will not be read correctly.

22.5.2　16-bit PPG Cycle Setting Buffer Registers Upper, Lower (PCSRH1, PCSRL1)

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 (PCSRH1, PCSRL1)

Figure 22.5-3 16-bit PPG Cycle Setting Buffer Registers Upper, Lower (PCSRH1, PCSRL1)

```
16-bit PPG cycle setting buffer register (upper) (PCSRH1)
Address    bit15   bit14   bit13   bit12   bit11   bit10   bit9    bit8    Initial value
0FB2_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) (PCSRL1)
Address    bit7    bit6    bit5    bit4    bit3    bit2    bit1    bit0    Initial value
0FB3_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 PCSRH1 register address).
- Use the "MOV" instruction and write to PCSRH1 first and then PCSRL1.
  If a down-counter load occurs after writing data to PCSRH1 (but before writing data to PCSRL1), the previous valid PCSRH1/PCSRL1 value will be loaded to the down-counter. If the PCSRH1/PCSRL1 value is modified during counting, the modified value will become effective from the next load of the down-counter.
- Do not set PCSRH1 and PCSRL1 to "00_H", or PCSRH1 to "01_H" and PCSRL1 to "01_H".

Note:

If the down-counter load occurs after the "MOV" instruction is used to write data to PCSRL1 before PCSRH1, the previous valid PCSRH1 value and newly written PCSRL1 value are loaded to the down-counter. It should be noted that as a result, the correct period cannot be set.

MB95330H Series

22.5.3   16-bit PPG Duty Setting Buffer Registers Upper, Lower (PDUTH1, PDUTL1)

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 (PDUTH1, PDUTL1)

Figure 22.5-4 16-bit PPG Duty Setting Buffer Registers Upper, Lower (PDUTH1, PDUTL1)

	16-bit PPG duty setting buffer register (upper) (PDUTH1)								
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FB4$_H$	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) (PDUTL1)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FB5$_H$	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 PDUTH1 register address).
- Use the "MOV" instruction and write to PDUTH1 first and then PDUTL1.
  If a down-counter load occurs after writing data to PDUTH1 (but before writing data to PDUTL1), 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.

22.5.4 16-bit PPG Status Control Register Upper, Lower (PCNTH1, PCNTL1)

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 (PCNTH1)

Figure 22.5-5 16-bit PPG Status Control Register, Upper (PCNTH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0044H	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".)
[shaded] : Initial value

Table 22.5-1 16-bit PPG Status Control Register, Upper (PCNTH1)

	Bit name	Function
bit7	CNTE: Timer enable bit	This bit is used to enable/stop PPG timer operation. **When the bit is set to "0"**, the PPG operation halts immediately and the PPG1 output goes to the initial level ("L" output if OSEL is "0"; "H" output if OSEL is "1"). **When the bit is set to "1"**, PPG operation is enabled and the PPG goes to standby to wait for a trigger.
bit6	STRG: Software trigger bit	This bit is used to start the PPG timer by software. **When the bit is set to "1"**, setting the CNTE bit to "1" starts the PPG timer. Reading this bit always returns "0".
bit5	MDSE: Mode select bit	This bit is used to set the PPG operation mode. **When the bit is set to "0"**, the PPG operates in PWM mode. **When the bit is set to "1"**, the PPG operates in one-shot mode. Note: Modifying this bit is prohibited during operation.
bit4	RTRG: Software retrigger enable bit	This bit is used to enable or disable the software retrigger function of the PPG during operation. **When the bit is set to "0"**, the software retrigger function is "disabled". **When the bit is set to "1"**, the software retrigger function is "enabled".
bit3 to bit1	CKS2 to CKS0: Count clock select bits	These bits select the operating clock for the 16-bit PPG timer. The count clock signal is generated by the prescaler. See Section 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 (PCSRH1, PCSRL1), and duty setting (PDUTH1, PDUTL1). **When the bit is set to "0"**, the PPG1 output mask function is disabled. **When the bit is set to "1"**, the PPG1 output mask function is enabled. When the PPG0 output polarity setting is set to "normal" (PCNTL1: OSEL = 0), the output is always masked to "L". When the polarity setting is se to "inverted" (PCNTL1: OSEL = 1), the PPG0 output is always masked to "H".

■ 16-bit PPG Status Control Register, Lower (PCNTL1)

Figure 22.5-6 16-bit PPG Status Control Register, Lower (PCNTL1)

Address 0045H	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value 00000000B
	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	
	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	Trigger, software trigger, and retrigger by TRG1 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 TRG1 has no effect on operation.
1	The operation is started by the rising edge of TRG1.

EGS1	Hardware trigger enable bit1
0	The falling edge of TRG1 has no effect on operation.
1	The operation is stopped by the falling edge of TRG1.

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 22.5-2 16-bit PPG Status Control Register, Lower (PCNTL1)

Bit name		Function
bit7	EGS1: Hardware trigger enable bit1	This bit determines whether to allow or disallow the falling edge of TRG1 input to stop operation. **When the bit is set to "0"**, the falling edge of TRG1 has no effect on operation. **When the bit is set to "1"**, the operation is stopped by the falling edge of TRG1.
bit6	EGS0: Hardware trigger enable bit0	This bit determines whether to allow or disallow the rising edge of TRG1 input to start operation. **When the bit is set to "0"**, the rising edge of TRG1 has no effect on operation. **When the bit is set to "1"**, the operation is started by the rising edge of TRG1.
bit5	IREN: PPG interrupt request enable bit	This bit enables or disables PPG interrupt request to the interrupt controller. **When the bit is set to "0"**, an interrupt request is disabled. **When the bit is set to "1"**, an interrupt request is enabled.
bit4	IRQF: PPG interrupt flag bit	This bit is set to "1" when a PPG interrupt occurs. **When the bit is set to "0"**, clears the bit. **When the bit is set to "1"**, has no effect on operation. "1" is always read in read-modify-write (RMW) instruction.
bit3, bit2	IRS1, IRS0: Interrupt type select bits	These bits select the interrupt type for the PPG timer. | IRS1	IRS0	Type of interrupt
0	0	Trigger by input, software trigger, or retrigger
0	1	Counter borrow
1	0	Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity
1	1	Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	
bit1	POEN: Output enable bit	This bit enables or disables output from the PPG output pin. **When the bit is set to "0"**, the pin serves as a general-purpose port. **When the bit is set to "1"**, the pin serves as the PPG timer output pin.
bit0	OSEL: Output inversion bit	This bit selects the polarity of PPG output pin. **When the bit is set to "0"**, the PPG output goes to "H" when "L" is output in the internal start and the 16-bit down-counter value matches the duty setting register value, and goes to "L" when a down-counter borrow occurs (Normal polarity). **When the bit is set to "1"**, the PPG output is inverted (Inverted polarity).

22.6   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 22.6-1 shows interrupt control bits and interrupt sources of the 16-bit PPG timer.

Table 22.6-1 Interrupt Control Bits and Interrupt Sources of 16-bit PPG Timer

Item	Description
Interrupt flag bit	PCNTL1:IRQF
Interrupt request enable bit	PCNTL1:IREN
Interrupt type select bits	PCNTL1:IRS1, IRS0
Interrupt sources	PCNTL1:IRS1, IRS0=00$_B$ Hardware trigger by TRG1 Pin input of 16-bit down-counter, software trigger and retrigger
	PCNTL1:IRS1, IRS0=01$_B$ Counter borrow of 16-bit down-counter
	PCNTL1:IRS1, IRS0=10$_B$ Rising edge of PPG1 output in normal polarity, or falling edge of PPG1 output in inverted polarity
	PCNTL1: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 (PCNTL1) is set to "1" and interrupt requests are enabled (PCNTL1: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 22.6-2 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

*: 16-bit PPG timer ch. 1 shares the interrupt request number and the vector table addresses mentioned in the table with the MPG (position detection/compare match).

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

MB95330H Series

22.7   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 (PCSRH1, PCSRL1) values are loaded and the 16-bit down-counter starts down-count operation when a software trigger is input or a hardware trigger by TRG1 pin input is input. When the count value reaches "1", the 16-bit PPG cycle setting buffer register (PCSRH1, PCSRL1) 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 TRG1 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 TRG1 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 TRG1 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 TRG1 Pin input: 3 machine clock cycles

● Invalidating the retrigger (RTRG of PCNTH1 register: bit4 = 0)

Figure 22.7-1 When Retrigger Is Invalid in PWM Mode
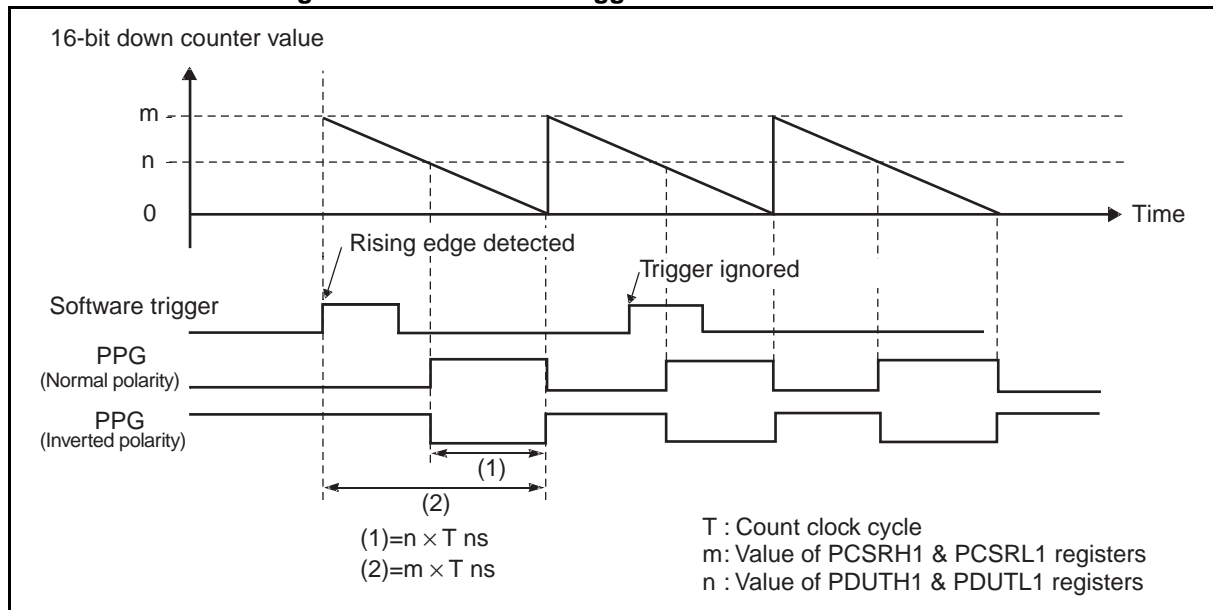
● Validating the retrigger (RTRG of PCNTH1 register: bit4 = 1)

Figure 22.7-2 When Retrigger Is Valid in PWM Mode

MB95330H Series
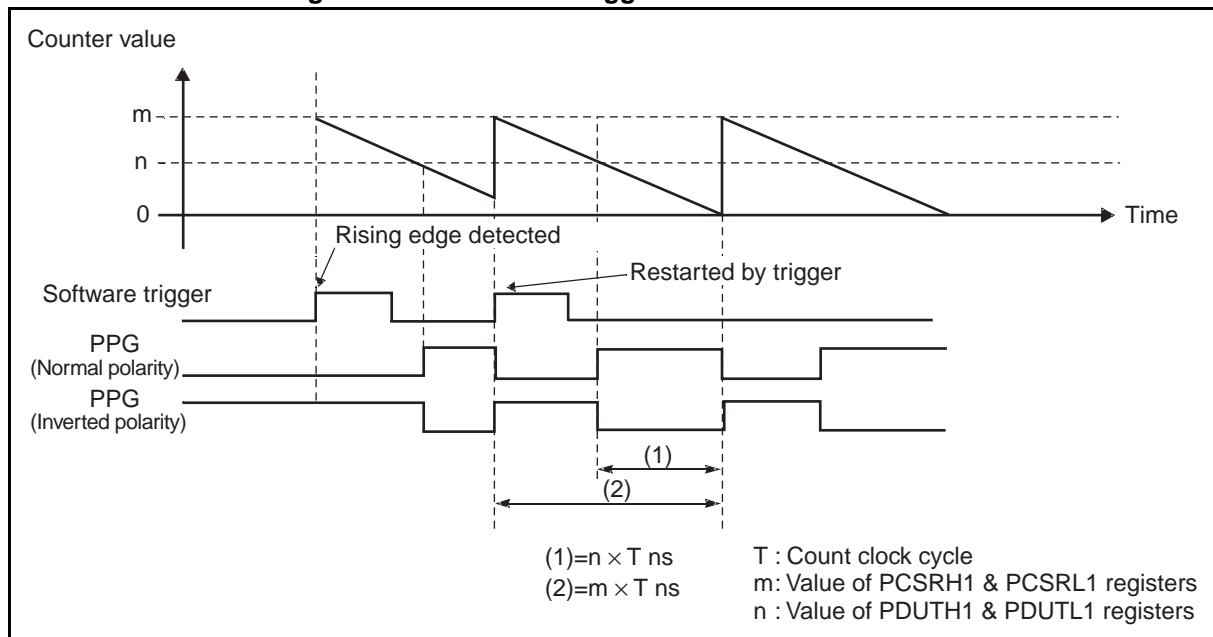
■ One-shot Mode (MDSE of PCNTH1 Register: bit5 = 1)

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 PCNTH1 register: bit4 = 0)

Figure 22.7-3 When Retrigger Is Invalid in One-shot Mode
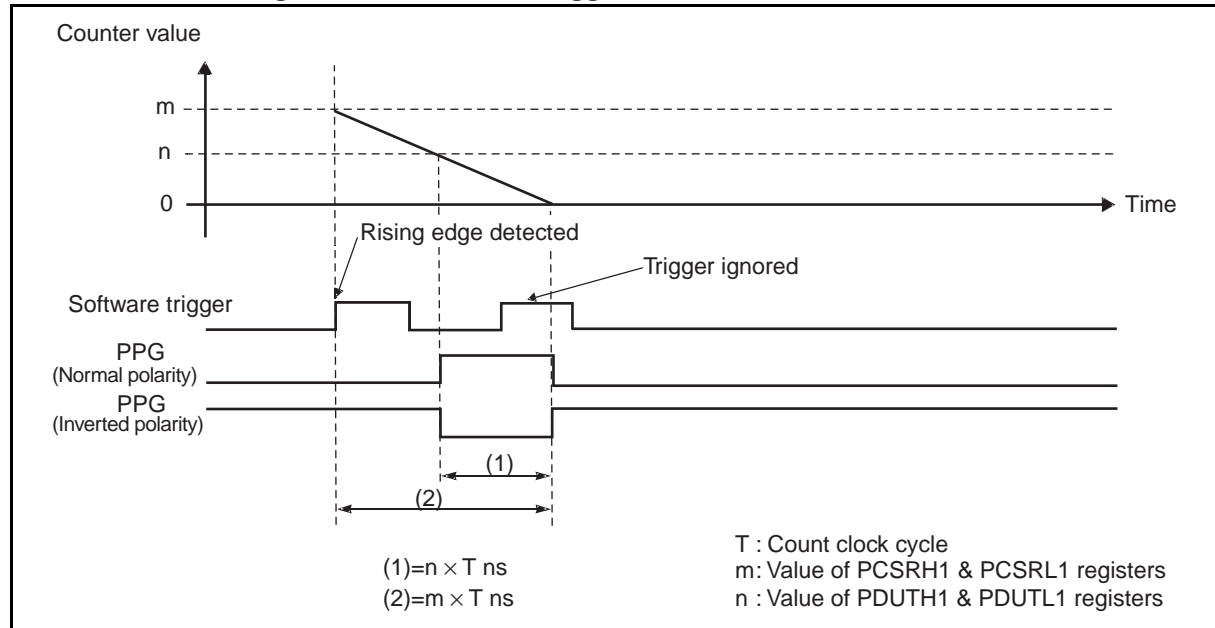
● Validating the retrigger (RTRG of PCNTH1 register: bit4 = 1)

Figure 22.7-4 When Retrigger Is Valid in One-shot Mode
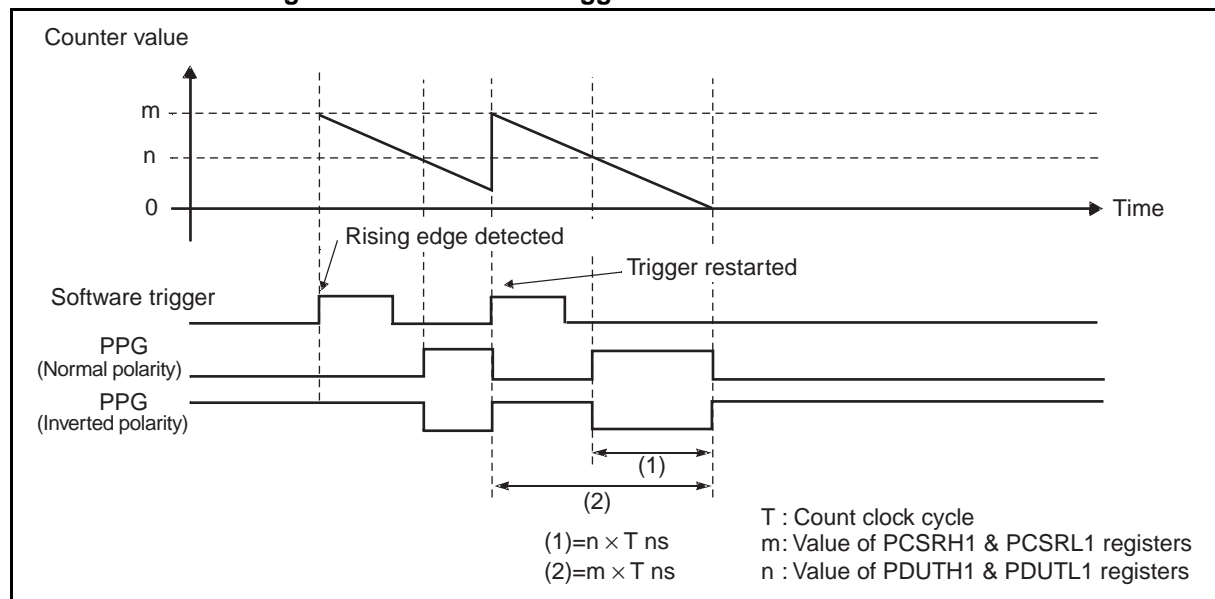
■ Hardware Trigger

"Hardware trigger" refers to PPG activation by signal input to the TRG1 input pin. When EGS1 and EGS0 are set to "11_B" and the hardware trigger is used with TRG1 input, PPG starts operation on a rising edge and halts the operation upon the detection of a falling edge.

Moreover, the PPG timer begins operation of the following rising edge from the beginning.

The operation can be retriggered by a valid TRG1 input hardware trigger regardless of the retrigger setting of the RTRG bit when the TRG1 input hardware trigger has been selected.

Figure 22.7-5 Hardware Trigger in PWM Mode
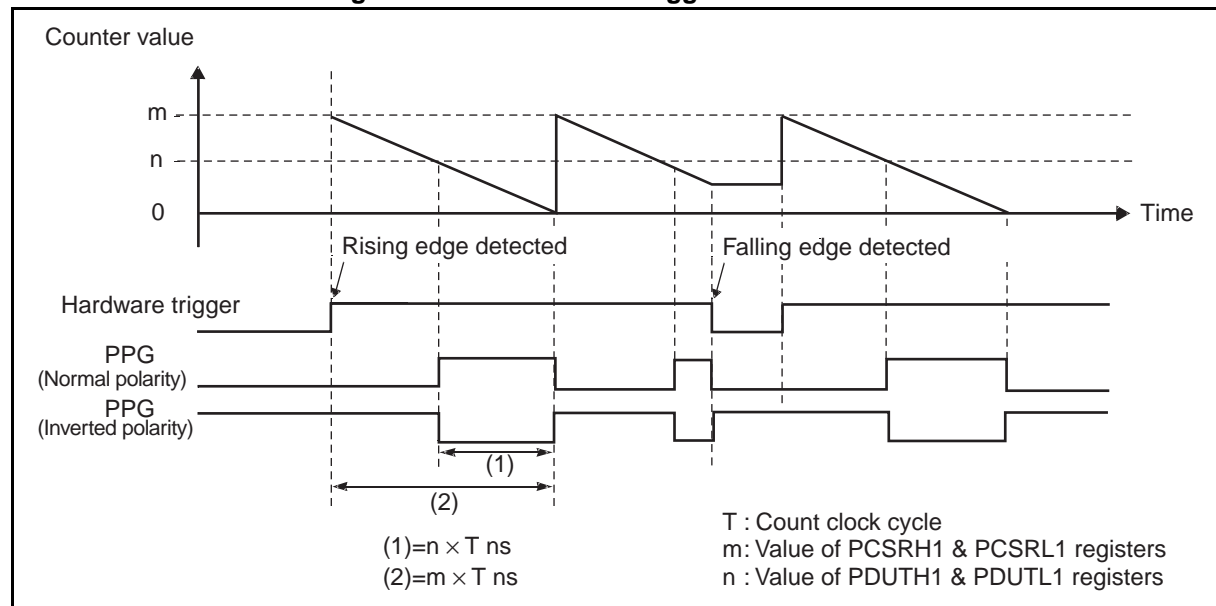
■ Setting Procedure Example

Below is an example of procedure for setting the 16-bit PPG timer.

● Initial setup

1) Set the interrupt level (ILR4)
2) Enable the hardware trigger and interrupts, select the interrupt type, and enable output (PCNTL1)
3) Select the count clock and the mode, and enable timer operation (PCNTH1)
4) Set the cycle (PCSRH1, PCSRL1)
5) Set the duty (PDUTH1, PDUTL1)
6) Start the PPG by the software trigger (PCNTH1:STRG = 1)

● Interrupt processing

1) Process any interrupt
2) Clear the interrupt request flag (PCNTL1:IRQF)

MB95330H Series

22.8   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 TRG1 pin setting may change and cause the device to malfunction. Therefore, disable the timer enable bit (PCNTH1:CNTE = 0) or disable the hardware trigger enable bit (PCNTL1: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 (PCNTH1:CNTE = 0) and software trigger (PCNTH1:STRG = 1) at the same time when retrigger by the software is enabled (PCNTH1:RTRG = 1) and the retrigger is selected as an interrupt type(PCNTL1:IRS1, IRS0 = 00_B) during count operation. If it occurs, interrupt flag bit may set by retrigger although timer stops.

22.9   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 (PCNTH1: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 (PCNTH1:CKS2/CKS1/CKS0) are used to select the clock.

● How to enable/disable the PPG output pin

The output enable bit (PCNTL1: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 (PCNTH1: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 (PCNTH1:STRG) is used.

Operation	Software trigger bit (STRG)
To start PPG operation with software	Set the bit to "1".

MB95330H Series

● How to enable/disable the retrigger function of the software trigger

The retrigger enable bit (PCNTH1: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/stop operation on a rising edge of trigger input

The hardware trigger enable bit (PCNTL1:EGS0) is used.

Operation	Hardware trigger enable bit (EGS0)
To start operation at a rising edge	Set the bit to "1".
To stop operation at a rising edge	Set the bit to "0".

● How to start/stop operation on a falling edge of trigger input

The hardware trigger enable bit (PCNTL1:EGS1) is used.

Operation	Hardware trigger enable bit (EGS1)
To start operation at a falling edge	Set the bit to "1".
To stop operation at a falling edge	Set the bit to "0".

● How to invert PPG output

The output inversion bit (PCNTL1: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 (PCNTH1:PGMS) and the output inversion bit (PCNTL1: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 the interrupt source

The interrupt select bits (PCNTL1: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 (PCNTL1:IREN) is used to enable interrupts.

Operation	Interrupt request enable bit (IREN)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The interrupt request flag (PCNTL1:IRQF) is used to clear an interrupt request.

Operation	Interrupt request flag (IRQF)
To clear an interrupt request	Set this bit to "0".

CHAPTER 23
16-BIT RELOAD TIMER

This chapter describes the functions and operations of the 16-bit reload timer.

23.1 Overview of 16-bit Reload Timer

The 16-bit reload timer has two counter operation modes available in the following two clock modes.
The 16-bit reload timer can be used as an interval timer by generating an interrupt when an underflow occurs in the timer.

■ Operation Modes of 16-bit Reload Timer

Table 23.1-1 shows the operation modes of the 16-bit reload timer.

Table 23.1-1 Operation Modes of 16-bit Reload Timer

Clock mode	Counter operating mode	Trigger operation mode
Internal clock mode	Reload mode	Software trigger operation External trigger input operation External gate input operation
	One-shot mode	
Event count mode (external clock mode)	Reload mode	Software trigger operation
	One-shot mode	

■ Internal Clock Mode

Internal clock mode is selected when any value other than "111_B" is set in the count clock setting bits (CSL2 to CSL0) of the timer control status register upper (TMCSRH1).

In internal clock mode, the following three trigger operation modes are available.

● Software trigger operation

The count starts when the count enable bit (CNTE) in the timer control status register lower (TMCSRL1) is set to "1" and the software trigger bit (TRG) is set to "1".

● External trigger input operation

When the count enable bit (CNTE) in the timer control status register lower (TMCSRL1) is set to "1", the count will start if a valid edge (rising, falling, or both selectable) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin.

● External gate input operation

When the count enable bit (CNTE) in the timer control status register lower (TMCSRL1) is set to "1", the count will start if a valid trigger input level ("L" or "H" selectable) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin.

■ Event Count Mode (External Clock Mode)

When the count clock setting bits (CSL2 to CSL0) in the timer control status register upper (TMCSRH1) are set to "111_B", the count will start if a valid edge of trigger input (rising, falling, or both) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin. When an external clock is input in regular cycles, the reload timer can also be used as an interval timer.

MB95330H Series

■ Counter Operating Mode

● Reload mode

The value of the 16-bit reload register (TMRLRH1/TMRLRL1) is loaded to the 16-bit down-counter and the count continues when an underflow occurs on the 16-bit down-counter ("0000_H" → "$FFFF_H$"). Also, the interrupt request is output by an underflow, so the mode can be used as the interval timer.

● One-shot mode

An interrupt is generated when an underflow occurs on the 16-bit down-counter.

During counter operation, the TO1 pin outputs a square waveform indicating that the counter is currently running.

23.2   Configuration of 16-bit Reload Timer

The 16-bit reload timer consists of the following blocks:
- **Count clock generation circuit**
- **Reload control circuit**
- **Output control circuit**
- **Operation control circuit**
- **16-bit timer register (TMRH1, TMRL1)**
- **16-bit reload register (TMRLRH1, TMRLRL1)**
- **Timer control status register (TMCSRH1, TMCSRL1)**

■ Block Diagram of 16-bit Reload Timer

Figure 23.2-1 shows the block diagram of the 16-bit reload timer.

Figure 23.2-1 Block Diagram of 16-bit Reload Timer

MB95330H Series
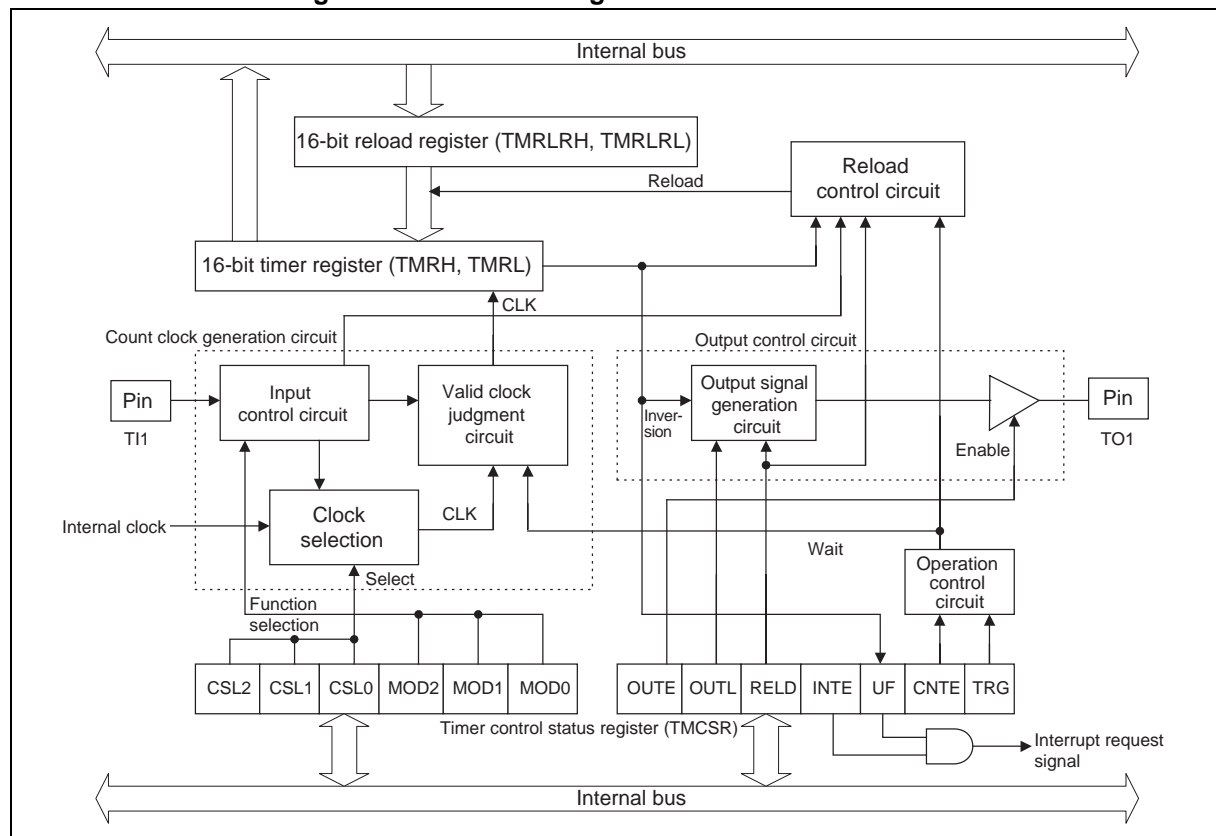
● Count clock generation circuit

The count clock for the 16-bit reload timer is generated from the internal clock or TI1 pin input signal.

● Reload control circuit

This circuit controls reload operation when the timer is started or an underflow occurs.

● Output control circuit

This circuit controls the inversion of TO1 pin output by an underflow of the 16-bit down-counter and the enabling and disabling of TO1 pin output.

● Operation control circuit

This circuit controls the starting and stopping of the 16-bit down-counter.

● 16-bit timer register (TMRH1, TMRL1)

TMRH and TMRL form a 16-bit down-counter. Reading returns the current count value.

● 16-bit reload register (TMRLRH1, TMRLRL1)

This register sets the load value to the 16-bit down-counter. The register loads the setting value of the 16-bit reload register to the 16-bit down-counter to down count.

● Timer control status register (TMCSRH1, TMCSRL1)

This register controls the count clock operation mode, clock selection, interrupts and other aspects of the 16-bit reload timer as well as indicates the current operation status.

■ Input Clock

The 16-bit reload timer uses the output clock from the prescaler or the input signal from the TI1 pin as its input clock (count clock).

23.3   Channel of 16-bit Reload Timer

This section describes the channel of the 16-bit reload timer.

■ Channels of 16-bit Reload Timer

The MB95330H Series has one channel of 16-bit reload timer.

Table 23.3-1 and Table 23.3-2 show the pins and registers of the 16-bit reload timer respectively.

Table 23.3-1 Pins of 16-bit Reload Timer

Channel	Pin name	Pin function
1	TO1	Timer output
	TI1	Timer input

Table 23.3-2 Registers of 16-bit Reload Timer

Channel	Register abbreviation	Corresponding register (Name in this manual)
1	TMCSRH1	16-bit reload timer control status register (upper)
	TMCSRL1	16-bit reload timer control status register (lower)
	TMRH1	16-bit reload timer timer register (upper)
	TMRL1	16-bit reload timer timer register (lower)
	TMRLRH1	16-bit reload timer reload register (upper)
	TMRLRL1	16-bit reload timer reload register (lower)

MB95330H Series

23.4　Pins of 16-bit Reload Timer

This section describes the pins of the 16-bit reload timer and shows the block diagram of these pins.

■ Pins of 16-bit Reload Timer

The pins of the 16-bit reload timer are namely the TI1 and TO1 pins.

● TI1 pin

This pin is used both as a general-purpose I/O port and as an external pulse input pin for the counter (TI1).

TI1:　Any pulse edge input to this pin is counted during counter operation. To use it as the TI1 pin in counter operation, set the port direction register (DDR6) to "0" and use the pin as an input port.

● TO1 pin

This pin is used both as a general-purpose I/O port and as the output pin of the 16-bit reload timer (TO1).

TO1:　The pin outputs a waveform of the 16-bit reload timer.

When using this pin as the TO1 pin for the 16-bit reload timer, enabling timer output (TMCSRL1:OUTE = 1) allows output to be performed automatically regardless of the setting of the port direction register (DDR1) and the pin to serve as the TO1 pin of the timer output.

■ Block Diagrams of Pins of 16-bit Reload Timer

Figure 23.4-1 Block Diagram of Pin TI1 (P61/INT09/SCL/TI1) of 16-bit Reload Timer
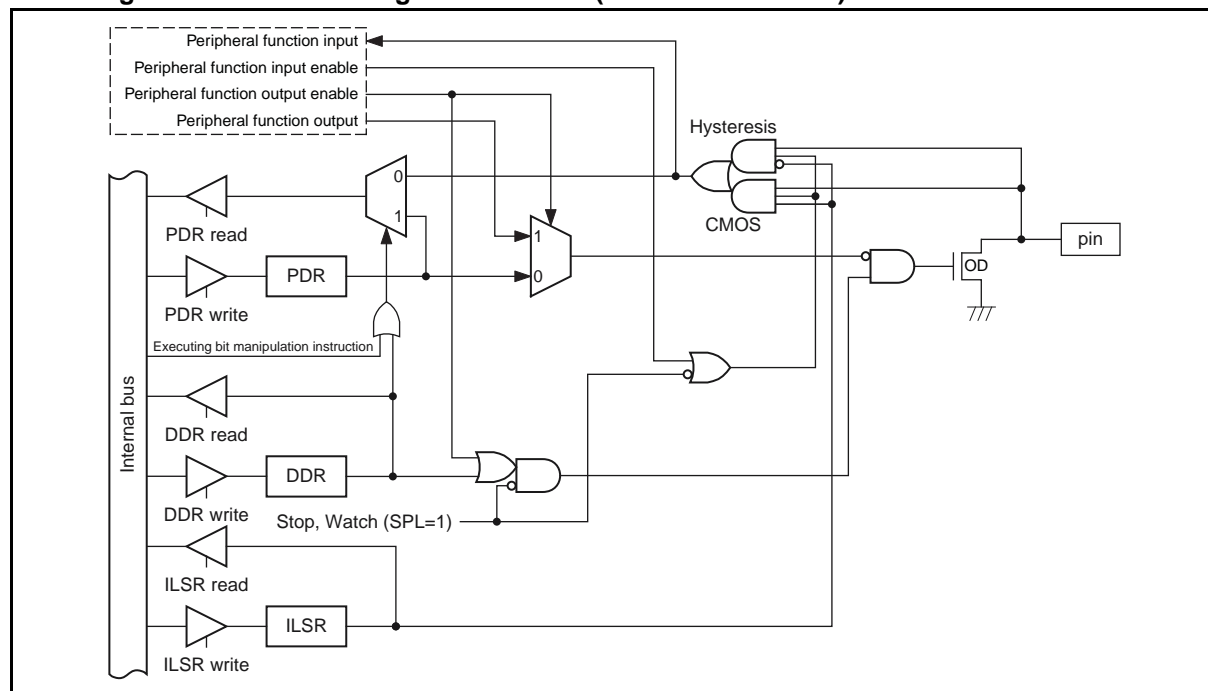
Figure 23.4-2 Block Diagram of Pin TO1 (P17/TO1/SNI0) of 16-bit Reload Timer

MB95330H Series
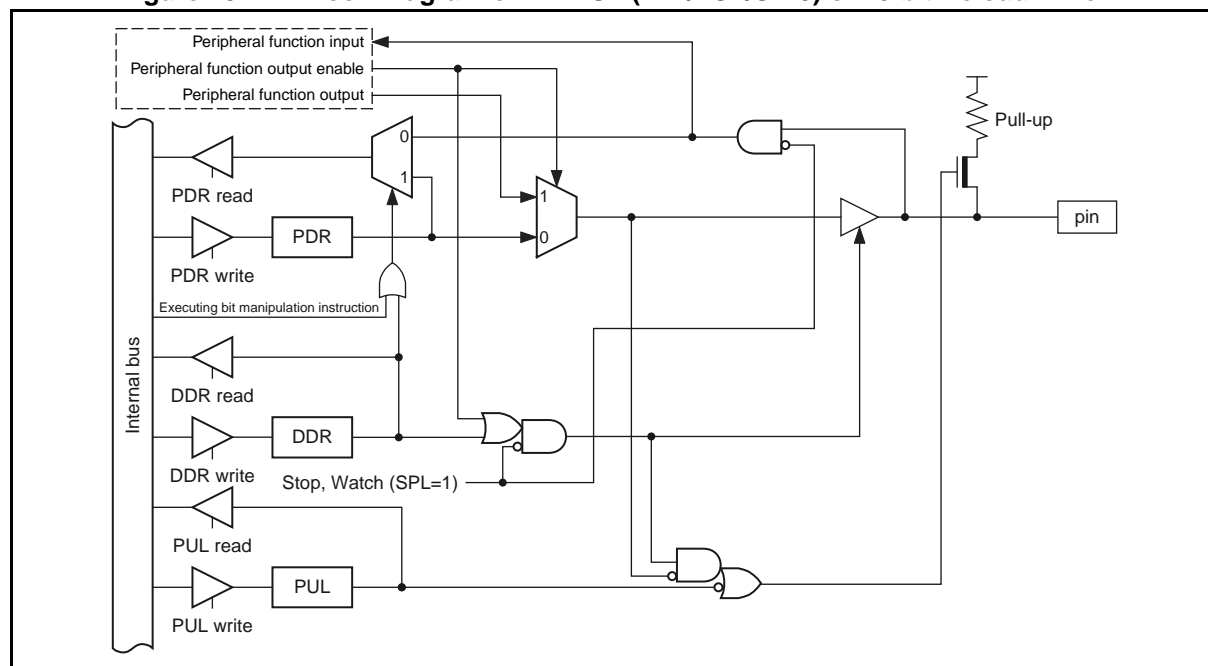
23.5 Registers of 16-bit Reload Timer

This section describes the registers of the 16-bit reload timer.

■ Registers of 16-bit Reload Timer

Figure 23.5-1 shows the registers of the 16-bit reload timer.

Figure 23.5-1 Registers of 16-bit Reload Timer

16-bit reload timer control status register (upper) (TMCSRH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0040$_H$	-	-	CSL2	CSL1	CSL0	MOD2	MOD1	MOD0	00000000$_B$
	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit reload timer control status register (lower) (TMCSRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0041$_H$	-	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	00000000$_B$
	R0/WX	R/W	R/W	R/W	R/W	R(RM1),W	R/W	R0,W	

16-bit reload timer timer register (upper) (TMRH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA8$_H$	D15	D14	D13	D12	D11	D10	D9	D8	00000000$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit reload timer timer register (lower) (TMRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA9$_H$	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	

16-bit reload timer reload register (upper) (TMRLRH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA8$_H$	D15	D14	D13	D12	D11	D10	D9	D8	00000000$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit reload timer reload register (lower) (TMRLRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FA9$_H$	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.)
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

Note: TMRH1 and TMRLRH1 are assigned to the same address.
      TMRL1 and TMRLRL1 are assigned to the same address.

23.5.1   16-bit Reload Timer Control Status Register Upper (TMCSRH1)

The 16-bit reload timer control status register (TMCSRH1) sets the operating mode and operating conditions of the 16-bit reload timer.

■ 16-bit Reload Timer Control Status Register Upper (TMCSRH1)

Figure 23.5-2 16-bit Reload Timer Control Status Register Upper (TMCSRH1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0040H	-	-	CSL2	CSL1	CSL0	MOD2	MOD1	MOD0	00000000B
	R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	

MOD2	MOD1	MOD0	Operation mode select bits (In internal clock mode, CSL2, CSL1, CSL0 = any value other than "111B")	
			Input pin function	Valid edge, level
0	0	0	External input invalid	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edges
1	X	0	Gate input	"L" level
1	X	1		"H" level

MOD2	MOD1	MOD0	Operation mode select bits (In event count mode, CSL2, CSL1, CSL0 = 111B)	
			Input pin function	Valid edge
0	0	0	External input invalid	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edges
1	X*	X*	Setting disabled	

* X: Either "0" or "1" can be selected.

CSL2	CSL1	CSL0	Count clock select bits	
			Operation mode	Count clock
0	0	0	Internal clock	MCLK/2
0	0	1		MCLK/4
0	1	0		MCLK/8
0	1	1		MCLK/16
1	0	0		MCLK/32
1	0	1		$F_{CH}/2^7$
1	1	0		$F_{CH}/2^8$
1	1	1	Event count	TI1 pin

MCLK   : Machine clock frequency
F_{CH} : Machine clock oscillation 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
▨   : Initial value

MB95330H Series

Table 23.5-1 16-bit Reload Timer Timer Control Status Register Upper (TMCSRH1)

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	CSL2, CSL1, CSL0: Count clock select bits	These bits select the count clock for the 16-bit reload timer. **Writing any value other than "111_B":** Internal clock is counted (internal clock mode). The internal clock is generated by the prescaler. See Section 6.12 "Operation of Prescaler". **Writing "111_B":** Edge of the external event clock is counted (event count mode).
bit2 to bit0	MOD2, MOD1, MOD0: Operating mode select bits	These bits set the operating conditions of the 16-bit reload timer. • **Internal clock mode (CSL2 to CSL0 = any value other than "111_B")** MOD2 bit selects the input pin function. **When MOD2 bit is set to "0":** - TI1 pin serves as a trigger input. - MOD1 and MOD0 bits are used to select the edge to be detected. - When the edge is detected, the value set in the 16-bit reload timer reload register is reloaded in the 16-bit reload timer timer register (TMR) and the TMR starts counting. **When MOD2 bit is set to "1":** - TI1 pin serves as a gate input. - Setting the MOD1 bit is invalid. - The MOD0 bit is used to select the valid signal level ("H" or "L"). The TMR only counts while the valid signal level is being input. Note: External input is disabled when MOD2 to MOD0 are "000_B". In this case, the TRG bit is used to start operation by software. • **Event count mode (CSL2 to CSL0 = 111_B)** - The MOD2 bit is always fixed to "0". - The external event clock is input from the TI1 pin. - The MOD1 and MOD0 bits are used to select the edge to be detected.

23.5.2   16-bit Reload Timer Control Status Register Lower (TMCSRL1)

The 16-bit reload timer control status register lower (TMCSRL1) sets the operating conditions of the 16-bit reload timer, enables or disables counting, controls interrupts, and checks the interrupt request status.

■ 16-bit Reload Timer Control Status Register Lower (TMCSRL1)

Figure 23.5-3 16-bit Reload Timer Control Status Register Lower (TMCSRL1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0041H	-	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	00000000B
	R0/WX	R/W	R/W	R/W	R/W	R(RM1),W	R/W	R0,W	

TRG	Software trigger bit	
	Read	Write
0	Always reads "0"	No effect on operation
1		Starts counting after reload

CNTE	Count enable bit
0	Stops count
1	Enables count (waiting for start trigger)

UF	Underflow interrupt request flag bit	
	Read	Write
0	No underflow	Clears this bit
1	Underflow	No effect on operation

INTE	Underflow interrupt request enable bit
0	Disables underflow interrupt
1	Enables underflow interrupt

RELD	Reload select bit
0	One-shot mode
1	Reload mode

OUTL	Pin output level select bit	
	One-shot mode	Reload mode
0	Outputs "H" square waveform during counting	Outputs "L" toggle when counting starts
1	Outputs "L" square waveform during counting	Outputs "H" toggle when counting starts

OUTE	Timer output enable bit
0	Disables timer output (general-purpose I/O port)
1	Enables timer output

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 23.5-2 16-bit Reload Timer Control Status Register Lower (TMCSRL1)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit6	OUTE: Timer output enable bit	This bit sets the TO1 pin function of the 16-bit reload timer. **Writing "0":** The pin functions as a general-purpose I/O port. **Writing "1":** The pin functions as the TO1 pin of the 16-bit reload timer.

bit5	OUTL:
	Pin output level select bit
		This bit sets the output level of the output pin of the 16-bit reload timer.
		• **When one-shot mode is selected (RELD = 0):**
		  "0": Outputs "H" level square waveform while the 16-bit reload timer counts.
		  "1": Outputs "L" level square waveform while the 16-bit reload timer counts.
		• **When reload mode is selected (RELD = 1):**
		  "0": Outputs an "L" when the 16-bit reload timer is started and then toggles each time an underflow occurs.
		  "1": Outputs an "H" when the 16-bit reload timer is started and then toggles each time an underflow occurs.

bit4	RELD: Reload select bit	This bit sets reload operation when an underflow occurs. **"0":** When an underflow occurs, counting is suspended. (One-shot mode) **"1":** When an underflow occurs, the value that has been set to the 16-bit reload register is loaded to the 16-bit timer register, and counting continues. (Reload mode)
bit3	INTE: Underflow interrupt request enable bit	This bit enables or disables underflow interrupts. **Writing "0":** Interrupt requests are disabled. **Writing "1":** Interrupt requests are enabled.
bit2	UF: Underflow interrupt request flag bit	This bit indicates that an underflow has occurred on the 16-bit reload timer. **Writing "0":** UF bit is cleared. **Writing "1":** Writing is nullified. • "1" is always read in read-modify-write instructions.
bit1	CNTE: Count enable bit	This bit enables or disables the operation of the 16-bit reload timer. **"0":** Counting is halted. **"1":** The unit goes to standby to wait for a start trigger. When a start trigger is input, the 16-bit timer register starts counting.
bit0	TRG: Software trigger bit	This bit allows the 16-bit reload timer to be started by software. The TRG bit is valid only when timer operation is enabled (CNTE = 1). **"0":** No effect on operation **"1":** The value set in the 16-bit reload register is reloaded to the 16-bit timer register and then the 16-bit timer register starts counting from the next count clock input. Note: This bit can be set to "1" at the same time as the CNTE bit without affecting the operation. • **Reading always returns "0":** However, "1" is read during the time between writing "1" to start the timer and the timer count actually starting.

23.5.3 16-bit Reload Timer Timer Register Upper (TMRH1)/Lower (TMRL1)

The 16-bit reload timer timer register upper (TMRH1) and lower (TMRL1) can be used to read the value of the 16-bit down-counter.

■ **16-bit Reload Timer Timer Register Upper (TMRH1)/Lower (TMRL1)**

Figure 23.5-4 16-bit Reload Timer Timer Register Upper (TMRH1)/Lower (TMRL1)

TMRH1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	D15	D14	D13	D12	D11	D10	D9	D8	00000000_B
$0FA8_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TMRL1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
$0FA9_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

The 16-bit timer register can read the count value of the 16-bit down-counter.

If counting is enabled (TMCSRL1:CNTE=1) at the beginning of a count, the value written in the 16-bit reload register will be reloaded to this register and the timer will start counting down.

Notes:

- This register can read the count value even during counting. When reading, use a word transfer instruction, or read the upper byte first and the lower byte second. The circuit is configured so that the value in the lower byte is saved when the upper byte is read.
- The registers are read-only and located at the same address as the 16-bit reload register. Accordingly, writing to these registers also writes to the 16-bit reload register.

MB95330H Series

23.5.4 16-bit Reload Timer Reload Register Upper (TMRLRH1)/Lower (TMRLRL1)

The 16-bit reload timer reload register upper (TMRLRH1)/lower (TMRLRL1) set the reload value for the 16-bit down-counter. The value set in the 16-bit reload registers is reloaded to the 16-bit down-counter to down count.

■ 16-bit Reload Timer Reload Register Upper (TMRLRH1)/Lower (TMRLRL1)

Figure 23.5-5 16-bit Reload Timer Reload Register Upper (TMRLRH1)/Lower (TMRLRL1)

TMRLRH1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	D15	D14	D13	D12	D11	D10	D9	D8	00000000_B
$0FA8_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
TMRLRL1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
$0FA9_H$	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

These registers set the reload value to the 16-bit down-counter.

The value set in the 16-bit reload timer reload registers is reloaded to the 16-bit down-counter to start down-counting at the timing of start or underflow. (Also rewritable during counter operation)

Notes:

- The registers can be written to even while the counter is running. Perform write access using a word transfer instruction or write the upper byte first and lower byte second. (The circuit is implemented so that the upper byte is not used until the lower byte is written.)

- These are write-only registers and located at the same address as the 16-bit timer register. Therefore, reading from them also reads from the 16-bit timer register.

23.6   Interrupts of 16-bit Reload Timer

The 16-bit reload timer outputs an interrupt request when an underflow occurs on the 16-bit down-counter.

■ Interrupts of 16-bit Reload Timer

Table 23.6-1 shows the interrupt control bits and interrupt sources of the 16-bit reload timer.

Table 23.6-1 Interrupt Control Bits and Interrupt Sources of 16-bit Reload Timer

Item	Description
Interrupt request flag bit	UF bit in TMCSRL1 register
Interrupt request enable bit	INTE bit in TMCSRL1 register
Interrupt source	Underflow of down-counter (TMRH1/TMRL1)

The 16-bit reload timer sets the underflow interrupt request flag bit (UF) in the 16-bit reload timer control status register lower (TMCSRL1) to "1" when an underflow occurs in the 16-bit down-counter ("0000_H" → "$FFFF_H$"). If the underflow interrupt request enable bit is enabled (INTE = 1), the interrupt request will be output to the interrupt controller.

■ Register and Vector Table Addresses Related to Interrupts of 16-bit Reload Timer

Table 23.6-2 Register and Vector Table Addresses Related to Interrupts of 16-bit Reload Timer

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
16-bit reload timer ch. 1[*]	IRQ16	ILR4	L16	$FFDA_H$	$FFDB_H$

ch.: Channel

* 16-bit reload timer ch. 1 shares the interrupt request number and the vector table addresses mentioned in the table with the MPG (write timing/compare clear) and I^2C.

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

MB95330H Series

23.7 Operations of 16-bit Reload Timer and Setting Procedure Example

---

This section describes the operating status of the 16-bit reload timer counter.

■ Operating Status of Counter

The counter status is determined by the value of the count enable bit (CNTE) in the 16-bit reload timer control status register (TMCSRL1) and the internal signal start trigger wait signal (WAIT). The STOP state (halted), WAIT state (waiting for a start trigger) and RUN state (operating state) can be set.

Figure 23.7-1 shows the status transition of these counters.

Figure 23.7-1 Diagram of Counter State Transition
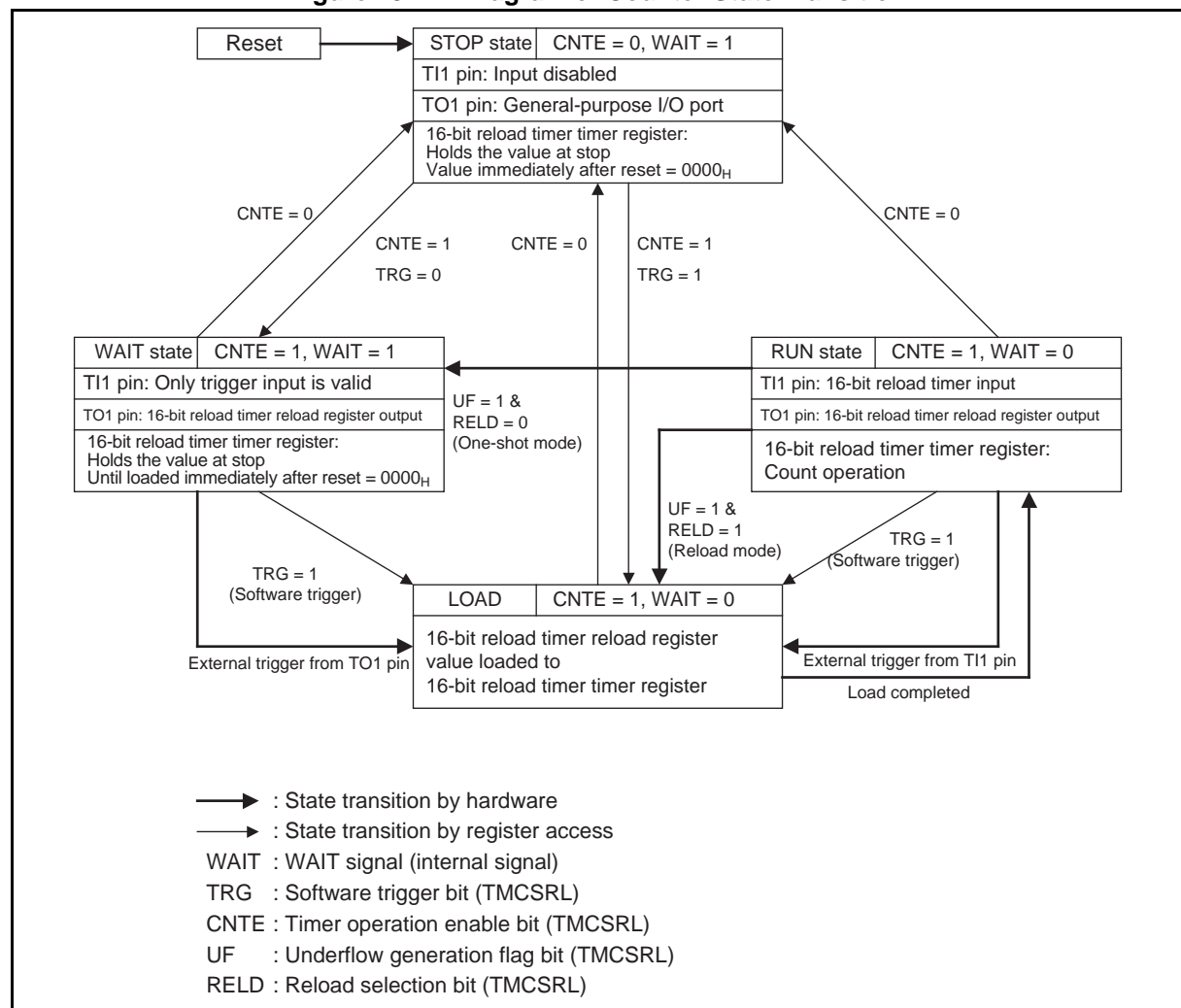
■ Setting Procedure Example

Below is an example of procedure for setting the 16-bit reload timer.

● Initial setup

1) Set the interrupt level. (ILR4)

2) Set the reload value. (TMR1)

3) Select the clock. (TMCSRH1:CSL2 to CSL0)

4) Select the operating mode. (TMCSRH1:MOD2 to MOD0)

5) Enable the output. (TMCSRL1:OUTE = 1)

6) Select the output level. (TMCSRL1:OUTL)

7) Select reload. (TMCSRL1:RELD)

8) Enable a count. (TMCSRL1:CNTE = 1)

9) Perform the software trigger. (TMCSRL1:TRG = 1)

10) Enable underflow interrupt. (TMCSRL1:INTE = 1)

● Interrupt processing

1) Clear the underflow interrupt request flag. (TMCSRL1:UF=0)

2) Disable underflow interrupt. (TMCSRL1:INTE = 0)

3) Process any interrupt.

4) Enable underflow interrupt. (TMCSRL1:INTE = 1)

MB95330H Series

23.7.1   Internal Clock Mode

In this mode, the 16-bit down-counter counts down while being synchronized with the internal count clock, and outputs an interrupt request to the interrupt controller every time an underflow occurs ("0000$_H$" → "FFFF$_H$"). In addition, the TO1 pin can output the toggle waveform.

■ **Setting Internal Clock Mode**

The timer requires the register settings shown in Figure 23.7-2 to operate as an interval timer.

Figure 23.7-2 Internal Clock Mode Setup

■ **Operation of Internal Clock Mode (Reload Mode)**

When "1" is set to the count enable bit (CNTE) to enable counting, and the timer is started by setting "1" to the software trigger bit (TRG) or by an external trigger, the value set in the 16-bit reload register (TMRLR1) is reloaded to the 16-bit down-counter and down-counting starts. If counting is enabled when the count enable bit (CNTE) and software trigger bit (TRG) are set to "1" at the same time, the count is started at the same time.

If the reload select bit (RELD) is "1", the value of the 16-bit reload register (TMRLR1) is reloaded to the 16-bit down-counter and the count continues when the 16-bit counter underflows ("0000$_H$" → "FFFF$_H$"). If the underflow interrupt request flag bit (UF) is "1" when the underflow interrupt request enable bit (INTE) is set to "1", an interrupt request is output.

The TO1 pin can output a toggle waveform that is inverted every time an underflow occurs.

● Software trigger operation

When the count enable bit (CNTE) is set to "1", setting "1" to the software trigger bit (TRG) starts counting.

Figure 23.7-3 shows the software trigger operation in reload mode.

Figure 23.7-3 Count Operation in Reload Mode (Software Trigger Operation)
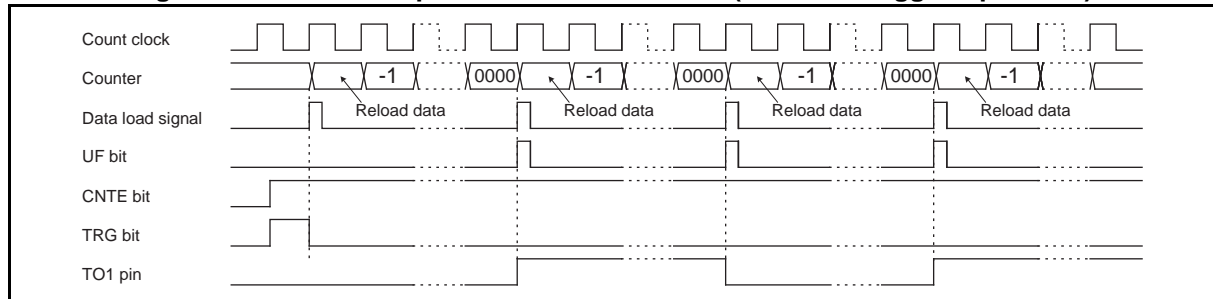
● External trigger input operation

The count starts when the count enable bit (CNTE) is set to "1" and a valid edge of trigger input (rising, falling, or both selectable) set by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin.

The timer start with the software trigger becomes effective as well as the one with an external trigger.

Figure 23.7-4 shows the external trigger input operation in reload mode.

Figure 23.7-4 Count Operation in Reload Mode (External Trigger Input Operation)
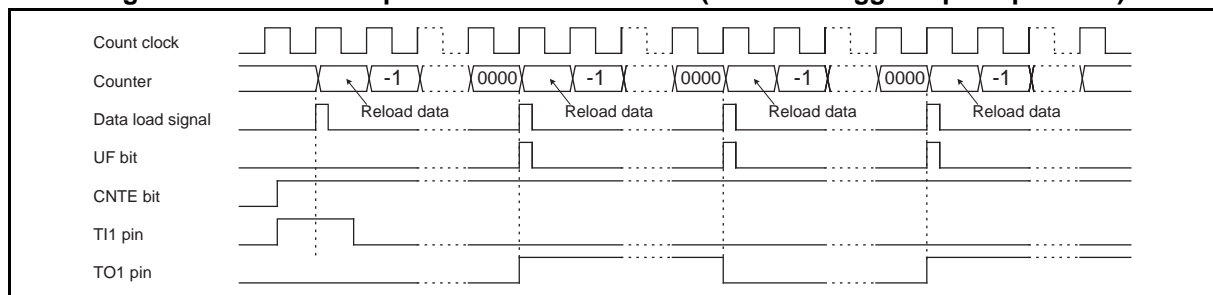
● Gate input operation

The count starts when the count enable bit (CNTE) is set to "1" and the software trigger bit (TRG) is also set to "1".

The timer continues counting while the valid gate input level ("L" or "H" selectable) set by the operating mode select bits (MOD2 to MOD0) is being input to the TI1 pin.

The timer start with the software trigger becomes effective as well as the one with an external trigger.

Figure 23.7-5 shows the gate input operation in reload mode.

Figure 23.7-5 Count Operation in Reload Mode (External Gate Input Operation)
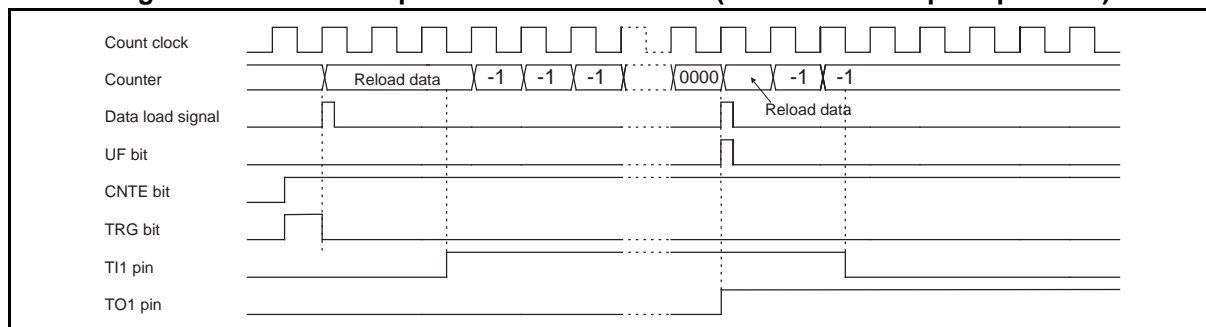
■ Operation of Internal Clock Mode (One-shot Mode)

When the count enable bit (CNTE) is set to "1" and the software trigger bit (TRG) is set to "1" or the valid edge (rising, falling or both edges selectable) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin, the value set in the 16-bit reload register is reloaded to the 16-bit down-counter and down-counting starts. When the count enable bit (CNTE) and software trigger bit (TRG) are set to "1" at the same time and then counting is enabled, the count is started simultaneously.

If the reload select bit (RELD) is "0", the 16-bit counter halts at "FFFF$_H$" when the 16-bit counter underflows ("0000$_H$" $\rightarrow$ "FFFF$_H$"). In this case, the underflow interrupt request flag bit (UF) is set to "1" and if the underflow interrupt request enable bit (INTE) is "1", an interrupt request is output.

A square waveform can be output from the TO1 pin to indicate that the count is in progress.

● Software trigger operation

The count starts when the count enable bit (CNTE) is "1" and the software trigger bit (TRG) is set to "1".

Figure 23.7-6 shows the software trigger operation in one-shot mode.

Figure 23.7-6 Count Operation in One-shot Mode (Software Trigger Operation)
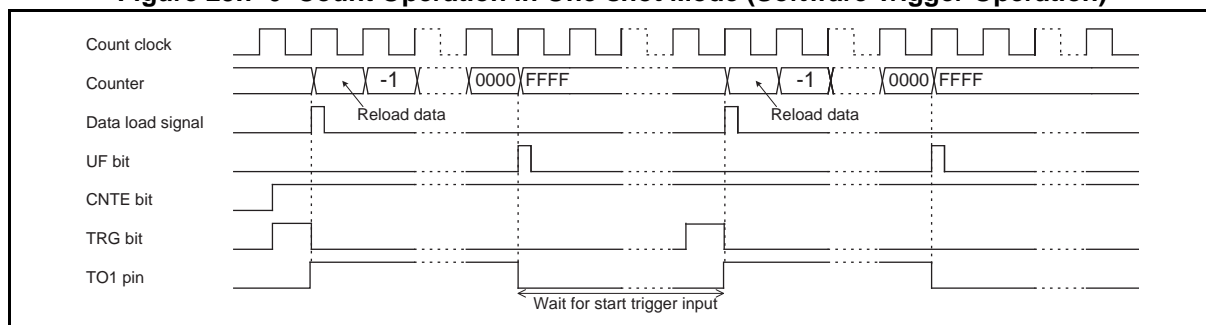
● External trigger input

The count starts when the count enable bit (CNTE) is "1" and the valid edge of trigger input (rising, falling, or both edges) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin.

Figure 23.7-7 shows the external trigger input operation in one-shot mode.

Figure 23.7-7 Count Operation in One-shot Mode (External Trigger Input Operation)
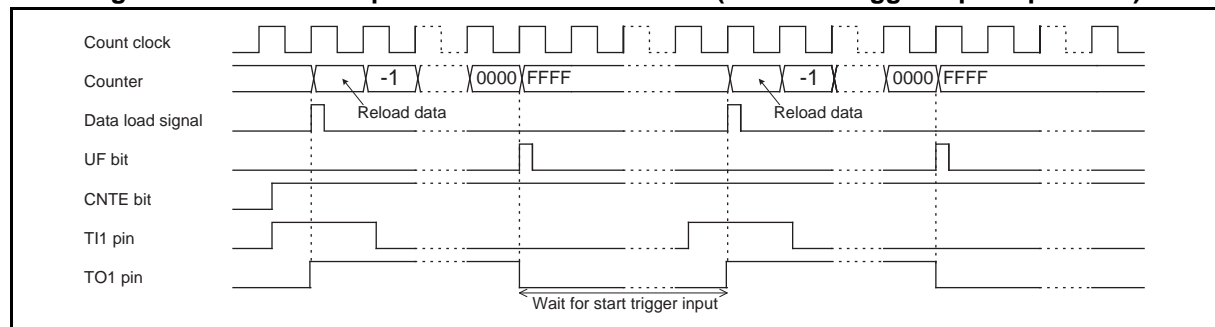
● Gate input operation

The count starts when the count enable bit (CNTE) is "1" and the software trigger bit (TRG) is also set to "1".

The timer continues counting as long as the trigger input enable level ("L" or "H" selectable) specified by the operating mode select bits (MOD2 to MOD0) is input to the TI1 pin.

Figure 23.7-8 shows the external gate input operation in one-shot mode.

Figure 23.7-8 Count Operation in One-shot Mode (External Gate Input Operation)

MB95330H Series
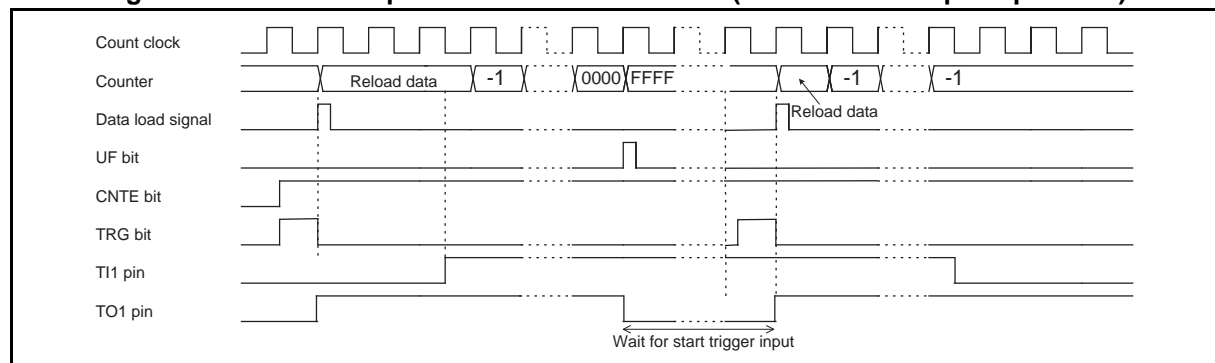
23.7.2 Event Count Mode

In this mode, the 16-bit down-counter counts down each time the valid edge is detected on the pulses input to the TI1 pin, and an interrupt request is output to the interrupt controller when an underflow occurs ("0000_H" $\rightarrow$ "$FFFF_H$"). In addition, a toggle waveform or square waveform can be output from the TO1 pin.

■ Event Count Mode Setup

The timer requires the register settings shown in Figure 23.7-9 to operate as an event counter.

Figure 23.7-9 Event Count Mode Setup

◉ : Used bit
1 : Set to "1"
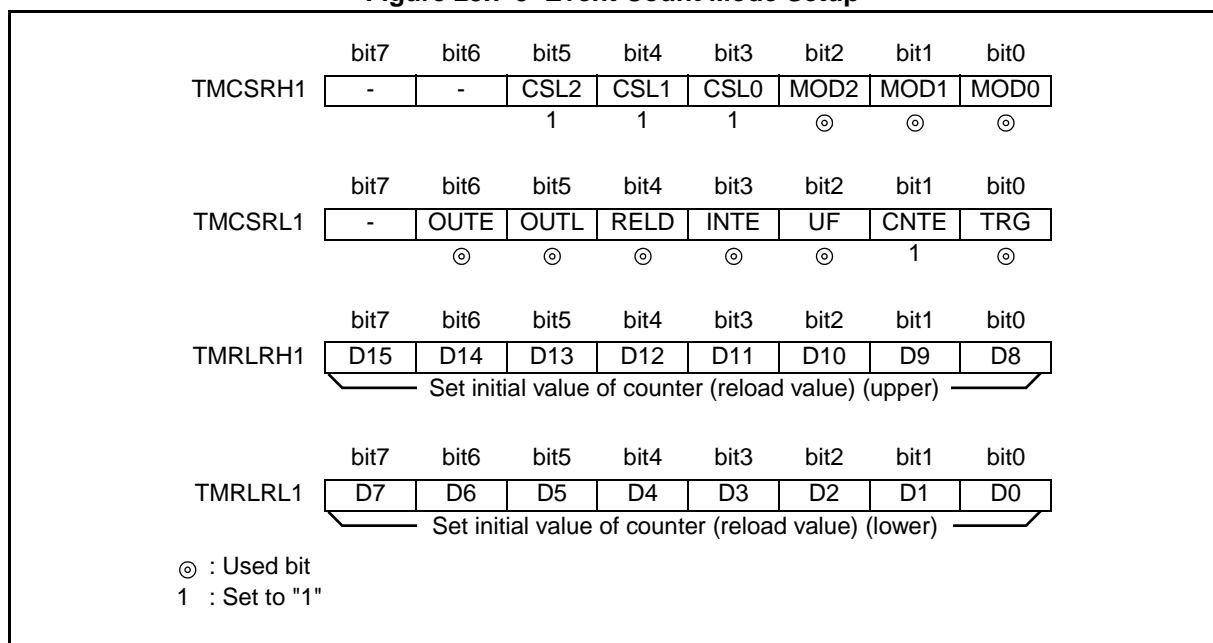
■ Event Count Mode

The value set in the 16-bit reload register (TMRLRH1/TMRLRL1) is reloaded to the 16-bit counter when the count enable bit (CNTE) is set to "1" and the software trigger bit (TRG) is set to "1". The counter counts each time the valid edge (rising, falling, or both edges selectable) is detected on the pulses input to the TI1 pin (external count clock).

● Operation of reload mode

If the reload select bit (RELD) is "1", the value set in the 16-bit reload register (TMRLRH1/ TMRLRL1) is reloaded to the 16-bit counter and the count continues when the 16-bit counter underflows ("0000_H" $\rightarrow$ "$FFFF_H$").

The underflow interrupt request flag bit (UF) in the lower timer control status register (TMCSRL1) is set to "1" when an underflow occurs ("0000_H" $\rightarrow$ "$FFFF_H$") in the 16-bit counter, and an interrupt request is output if the underflow interrupt enable bit (INTE) is set to "1".

The TO1 pin can output a toggle waveform that is inverted each time an underflow occurs. Figure 23.7-10 shows the count operation in reload mode.

Figure 23.7-10 Count Operation in Reload Mode (Event Count Mode)
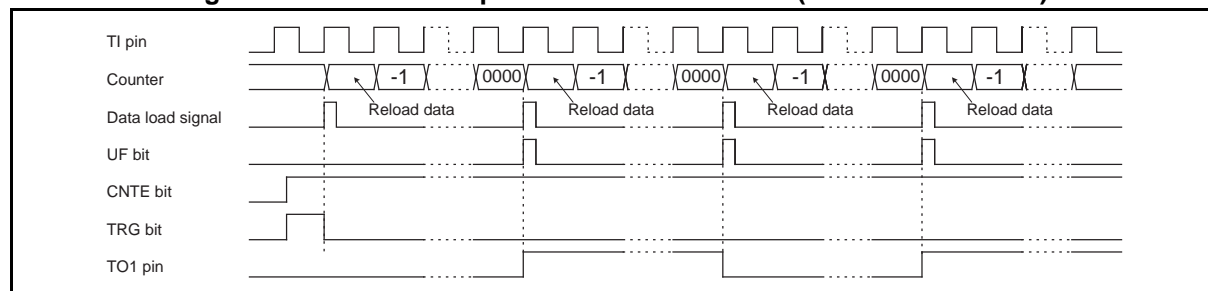
● Operation of one-shot mode

If the reload select bit (RELD) is "0", the value of the 16-bit counter halts at "FFFF$_H$" when the 16-bit counter underflows ("0000$_H$" → "FFFF$_H$").

An interrupt request is output when the underflow request flag bit (UF) in the lower timer control status register (TMCSRL1) is set to "1" with the underflow interrupt enable bit (INTE) set to "1".

The TO1 pin outputs a square waveform indicating that counting is in progress. Figure 23.7-11 shows the count operation in one-shot mode.

Figure 23.7-11 Counter Operation in One-shot Mode (Event Count Mode)

MB95330H Series
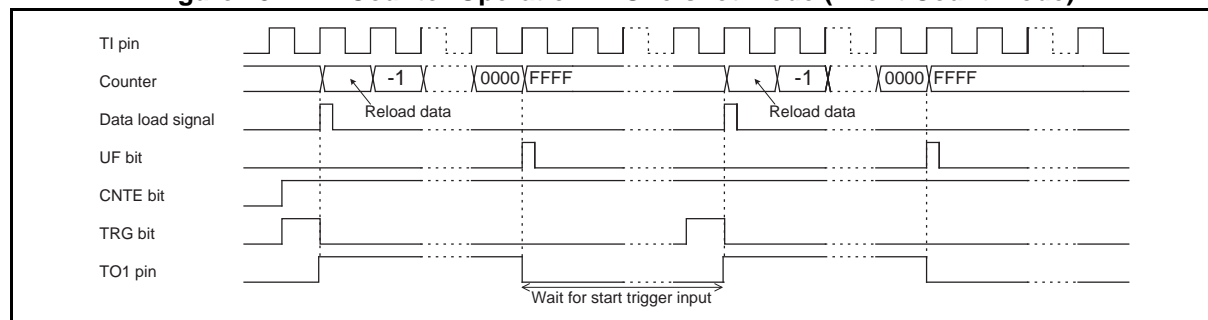
23.8   Notes on Using 16-bit Reload Timer

This section provides notes on using the 16-bit reload timer.

■ **Notes on Using 16-bit Reload Timer**

● Notes on setting the program

- A value can be read from the 16-bit timer register even during counting. As for read access, use a word transfer instruction or read the upper byte first and the lower byte second.
- A value can be written to the 16-bit reload register even during counting. As for write access, use a word transfer instruction or write the upper byte first and the lower byte second.

● Notes on interrupts

The unit cannot recover from interrupt processing when the underflow interrupt request enable bit (INTE) is set to "1" and "1" is set to the underflow interrupt request flag bit (UF) of the lower timer control status register (TMCSRL1). Always set the underflow interrupt request flag bit (UF) to "0".

23.9   Sample Settings for 16-bit Reload Timer

This section provides sample settings for the 16-bit reload timer.

■ Sample Settings

● How to select the count clock

The count clock select bits (TMCSR1:CSL[2:0]) are used.

Operation	Count clock select bits (CSL[2:0])
To select an internal clock	Set the bits to any value except "111_B".
To select the external event clock	Set the bits to "111_B".

● How to select the operating conditions of internal clock mode

The operating mode select bits (TMCSR1:MOD[2:0]) are used to set the conditions.

Operating condition	Operating mode select bits (MOD[2:0])
Trigger input from TI1 pin (rising edge)	Set the bits to "001_B".
Trigger input from TI1 pin (falling edge)	Set the bits to "010_B".
Trigger input from TI1 pin (both edges)	Set the bits to "011_B".
Gate input from TI1 pin (L level)	Set the bits to "$1x0_B$".
Gate input from TI1 pin (H level)	Set the bits to "$1x1_B$".

● How to select the operating conditions of event count mode

The operating mode select bits (TMCSR1:MOD[1:0]) are used to set the conditions.

Operating condition	Operating mode select bits (MOD[1:0])
Rising edge	Set the bits to "01_B".
Falling edge	Set the bits to "10_B".
Both edges	Set the bits to "11_B".

The setting of MOD2 has no effect on operation, whether it is "0" or "1".

MB95330H Series

● How to enable/stop the count operation of the reload timer

The count enable bit of the timer (TMCSR1:CNTE) is used.

Operation	Operation enable bit (CNTE)
To stop the reload timer	Set the bit to "0".
To enable the count operation of the reload timer	Set the bit to "1".

The count cannot be resumed from the stop state. Enable the operation before or at the same time as the activation.

● How to set reload the timer mode (reload/one-shot)

The mode select bit (TMCSR1:RELD) is used.

Operating mode	Mode select bit (RELD)
To select the one-shot mode	Set the bit to "0".
To select the reload mode	Set the bit to "1".

● How to invert the output level

The output level is specified as shown in the following table.

The pin output level select bit (TMCSR1:OUTL) is used to set the output level.

Output level	Pin output level select bit (OUTL)
"L" toggle output when count starts in reload mode	Set the bit to "0".
"H" toggle output when count starts in reload mode	Set the bit to "1".
Outputting "H" square waveform during counting in one-shot mode	Set the bit to "0".
Outputting "L" square waveform during counting in one-shot mode	Set the bit to "1".

● How to switch the TI1 pin to an external event input pin or to an external trigger input pin

"0" is set to the data direction specification bit (DDR6:P61).

Pin	Control bit	
TI1 pin	Data direction register DDR6	Direction bit (P61)

● How to enable/disable the TO1 pin

The timer output enable bit (TMCSR1:OUTE) is used.

Operation	Timer output enable bit (TMCSR1:OUTE)
To enable the TO1 pin	Set the bit to "1".
To disable the TO1 pin	Set the bit to "0".

● How to generate a start trigger

- How to generate the software trigger

  The software trigger bit (TMCSR1:TRG) is used.

  Writing "1" to the software trigger bit (TRG) generates a trigger.

  When enabling and starting operation at the same time, set the count enable bit (TMCSR1:CNTE) and the software trigger bit (TMCSR1:TRG) at the same time.

- How to generate an external trigger

  An external trigger is generated when the edge specified by the operating mode select bits is input to the trigger pin corresponding to each reload timer.

Timer	Trigger pin
Reload timer	TI1

● Interrupt-related register

The interrupt level is set by the interrupt level registers shown in the following table.

	Interrupt level setting bit	Interrupt vector
Reload timer ch. 1	Interrupt level register (ILR4) Address: $0007D_H$	#16 Address: $0FFDA_H$

● How to enable interrupts

Interrupt request enable bit, Interrupt request flag

The interrupt request enable bit (TMCSR1:INTE) is used to enable interrupts.

Operation	Interrupt request enable bit (INTE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The interrupt request bit (TMCSR1:UF) is used to clear an interrupt request.

Operation	Interrupt request bit (UF)
To clear an interrupt request	Set the bit to "0".

CHAPTER 24

MULTI-PULSE

GENERATOR

This chapter describes the specifications and operations of the multi-pulse generator.

24.1 Overview of Multi-pulse Generator

The multi-pulse generator consists of a 16-bit PPG timer, a 16-bit reload timer and a waveform sequencer. By using the waveform sequencer, 16-bit PPG timer output signal can be directed to multi-pulse generator output (OPT5 to OPT0) according to the input signal of the multi-pulse generator (SNI2 to SNI0). Meanwhile, the OPT5 to OPT0 output signal can be hardware terminated by DTTI input in case of emergency. The OPT5 to OPT0 output signals are synchronized with the PPG signal in order to eliminate the unwanted glitch. For details of the 16-bit PPG timer and the 16-bit reload timer, see CHAPTER 22 "16-BIT PPG TIMER" and CHAPTER 23 "16-BIT RELOAD TIMER" respectively.

■ Function of Waveform Sequencer

● Output Signal Control

With waveform sequencer, it is possible to generate 16-bit PPG waveform output and DC chopper waveform output at the multi-pulse generator output (OPT5 to OPT0).

- When an effective edge of the input signal from multi-pulse generator position detect input (SNI2 to SNI0) or when the 16-bit reload timer is underflow or when the OPDBRH0 and OPDBRL0 registers are set, one pairs of the output data buffer registers (OPDBRHx, OPDBRLx) will be loaded into the output data register upper (OPDUR) and the output data register lower (OPDLR).

- The output data register (OPDUR, OPDLR) determines the 16-bit PPG timer output to which OPT output (OPT5 to OPT0). By loading different output data buffer registers (OPDBRHx, OPDBRLx) into the output data register (OPDUR, OPDLR), various combination of OPT outputs (OPT5 to OPT0) can be obtained.

- Therefore, the 16-bit PPG timer output can be presented/absented at multi-pulse generator output (OPT5 to OPT0) or switch the PPG timer output signal from one OPT output to another OPT output according to the sequence set in the output data register (OPDUR, OPDLR) and 12 pairs of output data buffer registers (OPDBRHx, OPDBRLx). Meanwhile, the 16-bit reload timer can insert a delay when switch OPT output.

- Table 24.1-1 shows the combination the data transfer from the OPDBRHx and OPDBRLx registers to the OPDUR and OPDLR registers.

Table 24.1-1 Data Transfer from OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0 to OPDUR and OPDLR

Combination	Data transfer from OPDBRHx and OPDBRLx to OPDUR and OPDLR
1	Data transfer from OPDBRHx and OPDBRLx to OPDUR and OPDLR after values are written to OPDBRHx and OPDBRLx by software.
2	Triggered by the16-bit reload timer underflow.
3	Triggered by the position detection input (SNI2 to SNI0).
4	Triggered by the 16-bit reload timer underflow. The 16-bit timer is started by the position detection comparison circuit.
5	Triggered either by the 16-bit reload timer underflow, or by the position detection input.

- In the waveform sequencer, there is a 16-bit timer that can be used to measure the speed of

the motor and disable the OPT output in case of position detect missing.

- Forced stop control using DTTI pin input
  External pin control can be performed through DTTI pin input. (The pin level can be set by each pin or software.) There is selectable noise filter for DTTI input. Table 24.1-2 shows the noise width for noise filter of DTTI pin.

Table 24.1-2 Noise Width for Noise Filter

Selection	Noise width for DTTI and SNI2 to SNI0 pins
1	Cancel 4-cycle noise.
2	Cancel 8-cycle noise.
3	Cancel 16-cycle noise.
4	Cancel 32-cycle noise.

● PPG Synchronization for Output Signal

In order to avoid short pulse (or glitch) during sequencer state changes, the write timing (WTO) needs to be delayed and synchronized with the next coming edge of PPG output waveform. See Figure 24.1-1 and Figure 24.1-2 for details. This function can be enabled or disabled by software. The WTS1 and WTS0 bits in the input control register upper (IPCUR) are used to disable this function and to select the polarity of the PPG edge to synchronize with.

Figure 24.1-1 PPG Rising Edge Synchronization
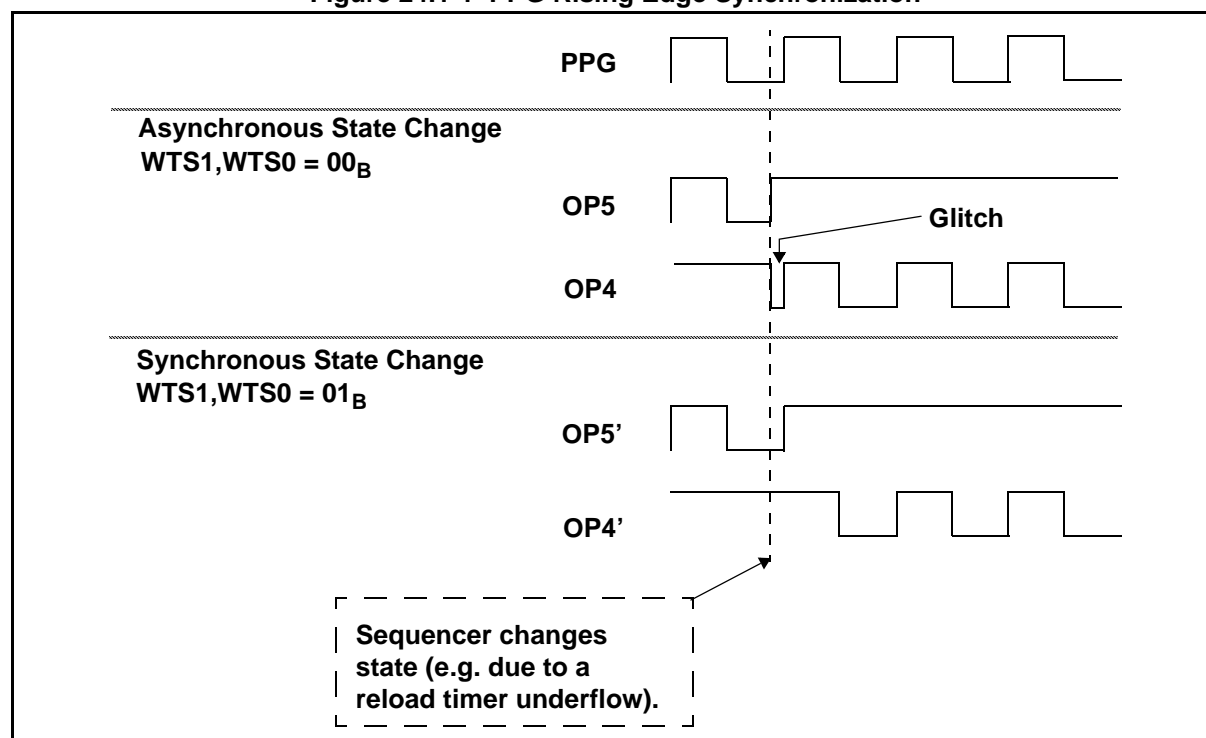
Figure 24.1-2 PPG Falling Edge Synchronization
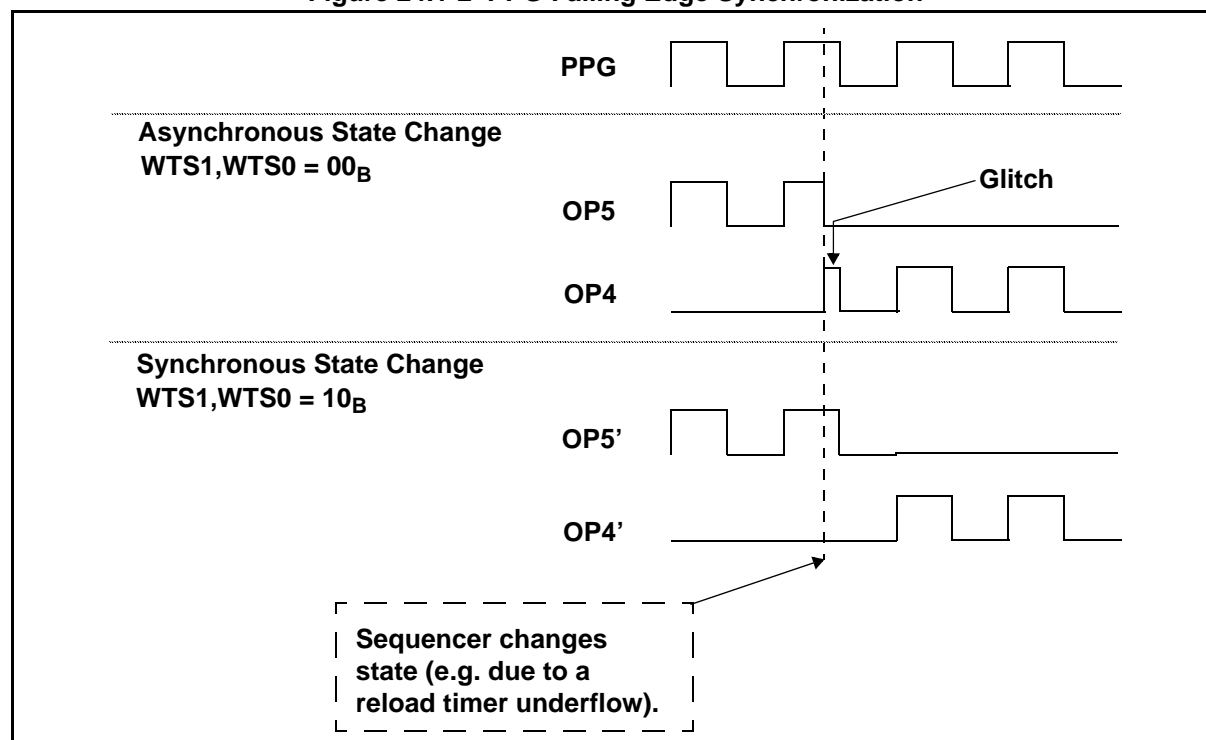
Note:

Directly changing from one PPG synchronization mode to another PPG synchronization mode (e.g. from rising-edge synchronization to falling-edge synchronization or vice versa) is prohibited. To change from one PPG synchronization mode to another PPG synchronization mode, disable PPG edge synchronization temporarily before changing to another PPG synchronization mode.

● Input Position Detect Control

The input signal at the multi-pulse generator input pins (SNI2 to SNI0) is used to detect the rotor position of the DC motor. There is a noise filter for all SNI2 to SNI0 input and Table 24.1-2 shows the noise width for noise filter of SNI2 to SNI0 pins. The followings are conditions for the input position detect circuit:

• 3 edge selection for all SNI2 to SNI0; Rising edge, falling edge and both edges.

• Compare the levels of SNI2 to SNI0 inputs with RDA2 to RDA0 bits in the output data register upper (OPDUR:RDA2 to RDA0).

After above condition met, the writing timing signal will be generated for the data transfer between the OPDBRHx and OPDBRLx registers and the OPDUR and OPDLR registers.

Furthermore, the edge detection for individual input (SNI2 to SNI0) can be disable/enable.

MB95330H Series

24.2   Block Diagram of Multi-pulse Generator

**Figure 24.2-1 shows the block diagram of the multi-pulse generator and
Figure 24.2-2 the block diagram of the waveform sequencer.**

■ **Block Diagram of Multi-pulse Generator**

Figure 24.2-1 Block Diagram of Multi-pulse Generator

*: The dotted line represents the TI1 path in the MB95330H Series.
  The 16-bit reload timer can be used independently in the MB95330H Series.
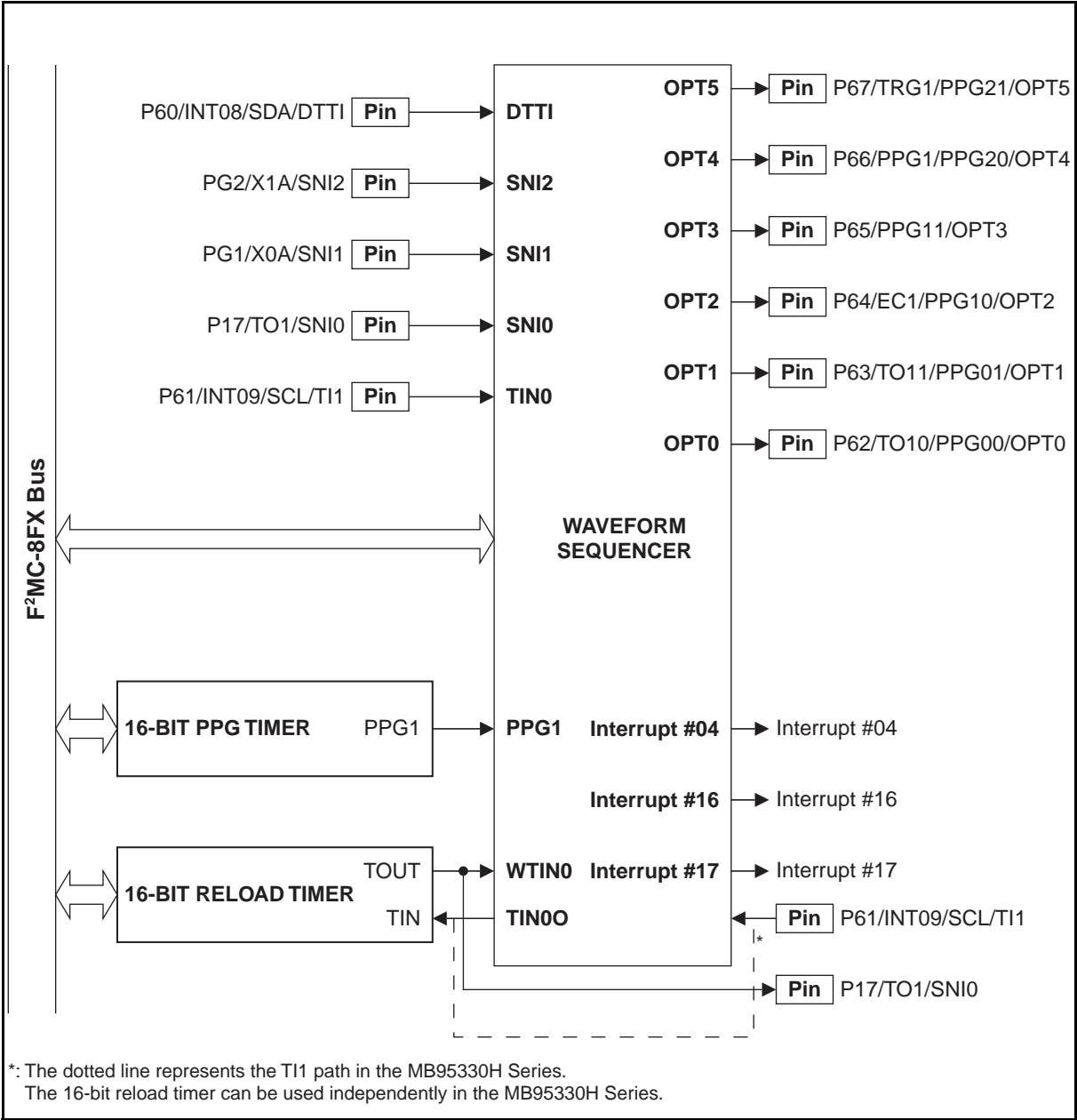
● 16-bit PPG Timer

The 16-bit PPG timer is used to provide the PPG signal for the waveform sequencer. Details of
the 16-bit PPG Timer are described in CHAPTER 22 "16-BIT PPG TIMER".

● 16-bit Reload Timer

The 16-bit reload timer is used to act as the interval timer for the waveform sequencer. Details of the 16-bit reload timer are described in CHAPTER 23 "16-BIT RELOAD TIMER".

● Waveform Sequencer

The waveform sequencer is the core of the multi-pulse generator, which can generate various waveforms. Its block diagram is shown in Figure 24.2-2.

MB95330H Series

■ Block Diagram of Waveform Sequencer

Figure 24.2-2 Block Diagram of Waveform Sequencer
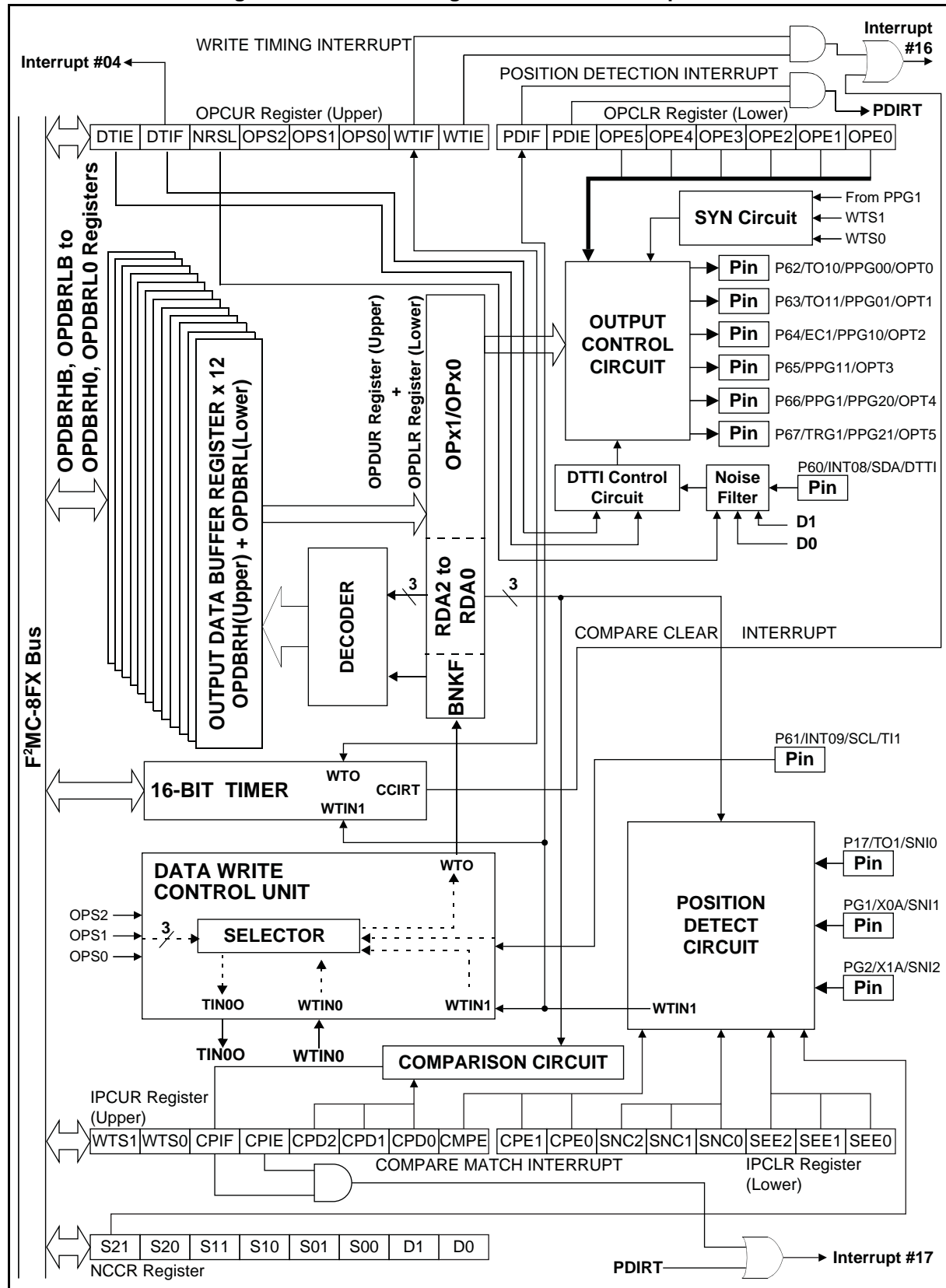
● 16-bit Timer

The 16-bit timer is used to act as an interval timer for motor speed checking and abnormal detection timer for controlling a DC sensorless motor. The detail is shown in Figure 24.2-3.

● Comparison Circuit

The comparison circuit is used to compare the RDA2 to RDA0 bits in the output data register (OPDUR) with the CPD2 to CPD0 bits in the input control register upper (IPCUR) for motor direction change. A compare match interrupt is generated when a match is happened.

● Data Write Control Unit

The data write control Unit is used to generate the write signal (WTO) for transferring data from the output data buffer register upper (OPDBRHx) and output data buffer register lower (OPDBRLx) to the output data register upper (OPDUR) and output data register lower (OPDLR). The detail is shown in Figure 24.2-4.

● Decoder

The decoder is used to decode the BNKF bit and RDA2 to RDA0 bits in the output data register upper (OPDUR) to select which pair of the output data buffer registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0) is loaded into the output data register.

● DTTI Control

The DTTI control is used to stop the multi-pulse generator output in case of emergency, which is triggered by level "0" of DTTI input.

● Noise Filter

The noise filter is used to filter out the noise of the input signal in which there are four types of sampling clock for selection.

● Output Control Unit

The output control unit is used to enable/disable PPG signal to the multi-pulse generator outputs (OPT5 to OPT0).

● Position Detect Circuit

The position detect circuit is used to detect the edge/level of the position input (SNI2 to SNI0). The detail is shown in Figure 24.2-5.

● SYN Circuit

The SYN Circuit is used to synchronize the OPT5 to OPT0 outputs with the PPG signal.

● Noise Cancellation Control Register (NCCR)

The noise cancellation control register (NCCR) is used to select one of four sampling clock for the noise filter.

● Output Control Register Upper (OPCUR) and Output Control Register Lower (OPCLR)

The output control register upper (OPCUR) and the output control register lower (OPCLR) are registers which enable the write timing interrupt and flag, position detect interrupt and flag, set the data transfer method, and set the control of the OPT5 to OPT0 and DTTI pins.

● Output Data Buffer Registers (OPDBRHx, OPDBRLx)

The output data buffer register is composed of twelve pairs of registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0). OPDBRHx is the upper byte register and OPDBRLx the lower byte register. The values of the OPDBRHx and OPDBRLx registers specified by the BNKF, RDA2 to RDA0 bits are loaded into the OPDUR and OPDLR registers at the rising edge of the write signal generated by the Data Write Control Unit.

● Output Data Register Upper (OPDUR) and Output Data Register Lower (OPDLR)

The output data register upper (OPDUR) and the output data register lower (OPDLR) are used to store the output data to the OPT5 to OPT0 pins.

■ **Block Diagram of 16-bit Timer**

Figure 24.2-3 Block Diagram of 16-bit Timer
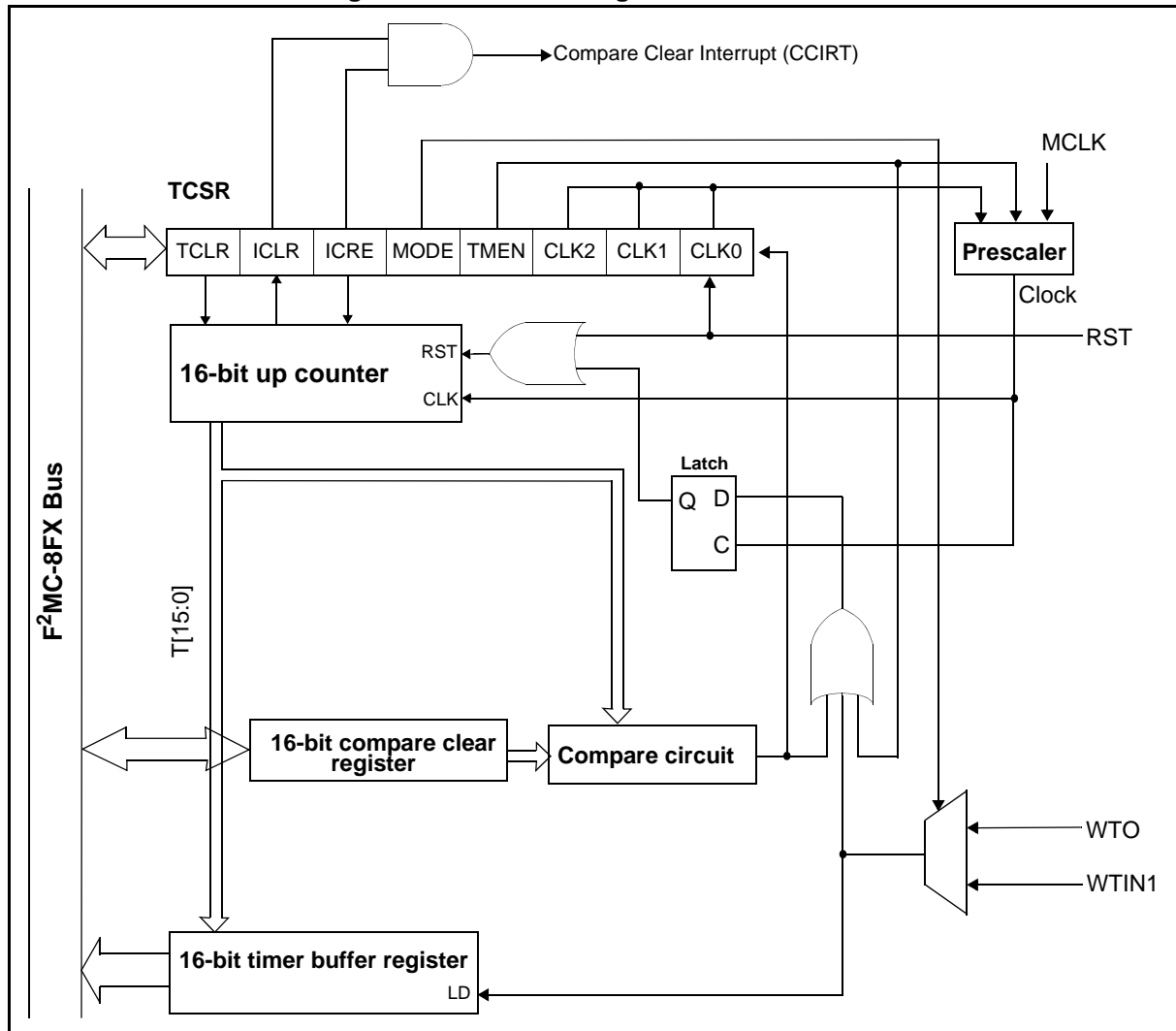
● 16-bit Up Counter

> The 16-bit Up Counter will be cleared when the match is happened between the count value and the Compare Clear register.

● Compare Circuit

> The Compare Circuit is used to compare the count value of the 16-bit Up Counter and the Compare Clear register.

● Compare Clear Register Upper (CPCUR) and Compare Clear Register Lower (CPCLR)

> The compare clear register upper (CPCUR) and the compare clear register lower (CPCLR) are used to store the 16-bit value which is used to compare the value of the 16-bit Up Counter.

● Timer Buffer Register Upper (TMBUR) and Timer Buffer Register Lower (TMBLR)

The timer buffer register upper (TMBUR) and the timer buffer register lower (TMBLR) are used store the value of the 16-bit Up Counter when a write timing interrupt or position detect interrupt occurs.

● Timer Control Register (TCSR)

The timer control status register (TCSR) is used to control the operation of the 16-bit timer such as the clock frequency, enable/disable the interrupt.

■ Block Diagram of Data Write Control Unit

Figure 24.2-4 Block Diagram of Data Write Control Unit
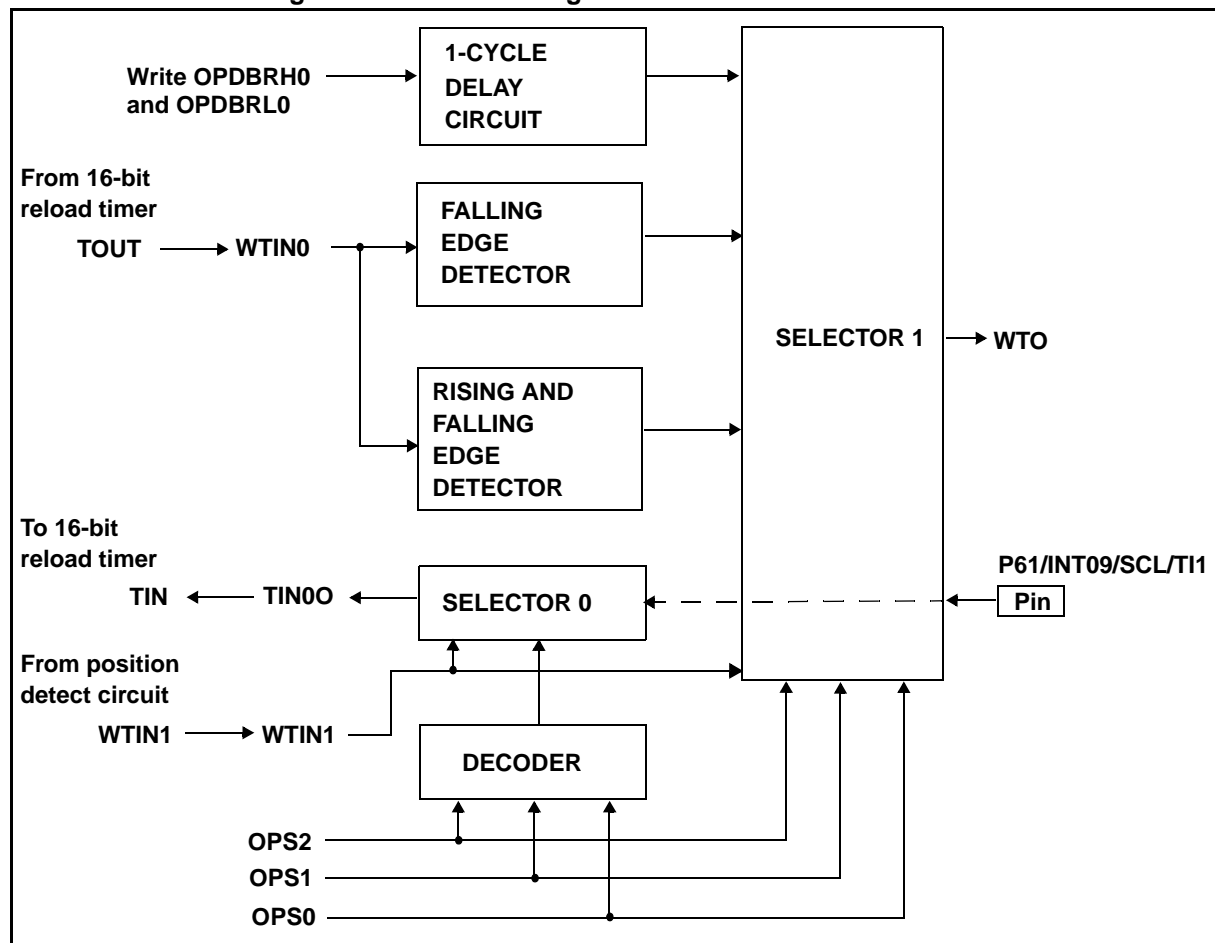
● 1-Cycle Delay Circuit

The 1-Cycle Delay Circuit is used to delay one CPU clock cycle of the trigger signal when the output data buffer register 0 (OPDBRH0 and OPDBRL0) is written.

● Selector 0

The Selector 0 is used to select from either WTIN1 of the position detect circuit or external pin (P61/INT09/SCL/TI1) to enable the count of the 16-bit reload timer.

● Selector 1

The Selector 1 is used to select from among Write both OPDBRHx and OPDBRLx or TOUT of 16-bit reload timer or WTIN1 of position detect circuit to generate the Write Timing signal (WTO).

● Falling Edge Detector

The Falling Edge Detector is used to detect the falling edge of the 16-bit reload timer output (TOUT).

● Rising and Falling Edge Detector

The Rising and Falling Edge Detector is used to detect the rising and falling edge of the 16-bit reload timer output (TOUT).

When timer underflow trigger is used in following modes, the WTIN0 signal is generated by the trigger edge selected by OPS2 to OPS0 bits:

Table 24.2-1 TOUT Trigger Edge Selection for WTIN0

OPS2	OPS1	OPS0	TOUT Trigger Edge for WTIN0
0	0	0	-
0	0	1	Rise and Fall
0	1	0	-
0	1	1	Fall
1	0	0	Rise and Fall
1	0	1	Rise and Fall
1	1	0	-
1	1	1	Fall

■ **Block Diagram of Position Detection Circuit**

Figure 24.2-5 Block Diagram of Position Detection Circuit
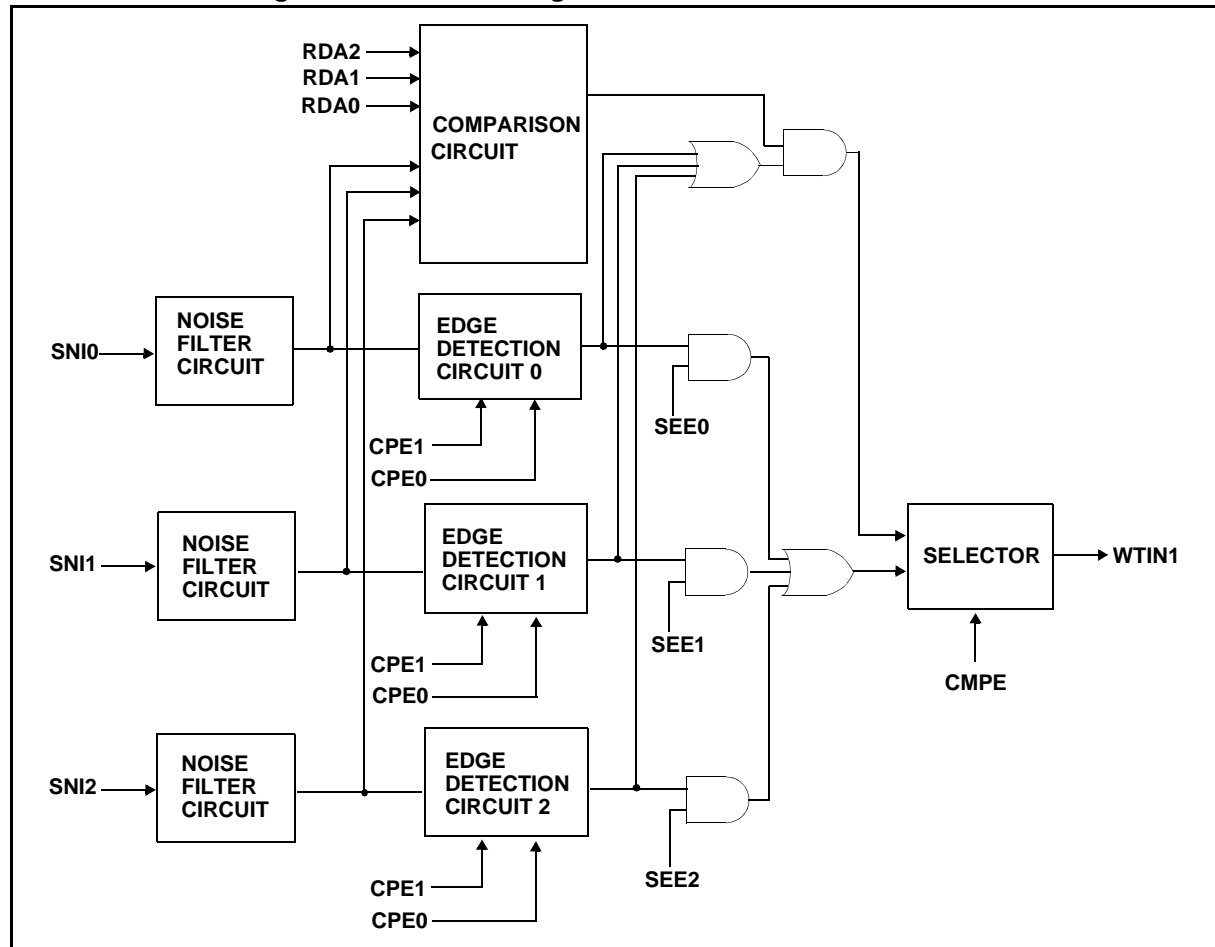
● Comparison Circuit

The Comparison Circuit is used to compare the level of the position detection input (SNI2 to SNI0) with RDA2 to RDA0 bits in the output data register (OPDUR). If the selector is selected, a data write time output signal is generated when a match is detected.

● Edge Detect Circuit 0, 1, 2

Edge Detect Circuit 0, 1 and 2 are identical.

The Edge Detect Circuit is used to compare the edge of the position input (SNI2 to SNI0) with 3 different kind of edge setting. If the selector is selected, a data write time output signal is generated when an effective edge is detected at the one of SNI2 to SNI0 inputs.

● Noise Filter

The noise filter is used to filter out the noise of the input signal in which there are 4 kind of sampling clock for selection.

● Selector

The Selector is used to select from either Edge Detect Circuit or Comparison Circuit to generate data write time output signal to the Data Write Control Unit.

24.3 Pins of Multi-pulse Generator

This section describes the pins of the multi-pulse generator and provides pin block diagrams.

■ Pins of Multi-pulse Generator

The multi-pulse generator uses P62/OPT0 to P67/OPT5, P17/SNI0, PG1/SNI1, PG2/SNI2, P60/INT08/SDA/DTTI and P61/INT09/SCL/TI1.

● P62/OPT0 to P67/OPT5 Pins

P62/OPT0 to P67/OPT5 pins can function either as a general-purpose I/O port (P62 to P67) or as the waveform output for the multi-pulse generator.

Enabling waveform output bit (OPCLR:OPE5 to OPE0 = 111111_B) automatically sets the P62/OPT0 to P67/OPT5 pin as an output pin, regardless of the port data direction register (DDR6:bit7 to bit2) value, and sets the pin to function as the OPT5 to OPT0 pins.

● P17/SNI0, PG1/SNI1, PG2/SNI2 Pins

P17/SNI0, PG1/SNI1 and PG2/SNI2 pins can function either as a general-purpose I/O port (P17, PG1 and PG2) or as the position detect input for the multi-pulse generator.

Set P17/SNI0, PG1/SNI1 and PG2/SNI2 pins as an input port in the data direction register (DDR6:bit7 = 0 and DDRG:bit2 to bit1 = 00_B) when using as the SNI2 to SNI0 pins.

● P60/INT08/SDA/DTTI Pins

P60/INT08/SDA/DTTI pins can function as a general-purpose I/O port (P60), or as the external interrupt INT8, or as the DTTI input for the multi-pulse generator.

Set P60/INT08/SDA/DTTI pins as an input port in the data direction register (DDR6: bit0 = 0) when using as the DTTI pin.

● P61/INT09/SCL/TI1 Pins

P61/INT09/SCL/TI1 pins can function as a general-purpose I/O port (P61), or as the external interrupt INT9, or as the input of 16-bit reload timer for the multi-pulse generator.

Set P61/INT09/SCL/TI1 pin as an input port in the data direction register (DDR6:bit1= 0) when using as the TI1 pin.

MB95330H Series

■ Block Diagrams of Pins of Multi-pulse Generator

Figure 24.3-1 Block Diagram of Pins OPT0, OPT1, OPT3 and OPT4 (P62/TO10/PPG00/OPT0, P63/TO11/PPG01/OPT1, P65/PPG11/OPT3 and P66/PPG1/PPG20/OPT4) of Multi-pulse Generator
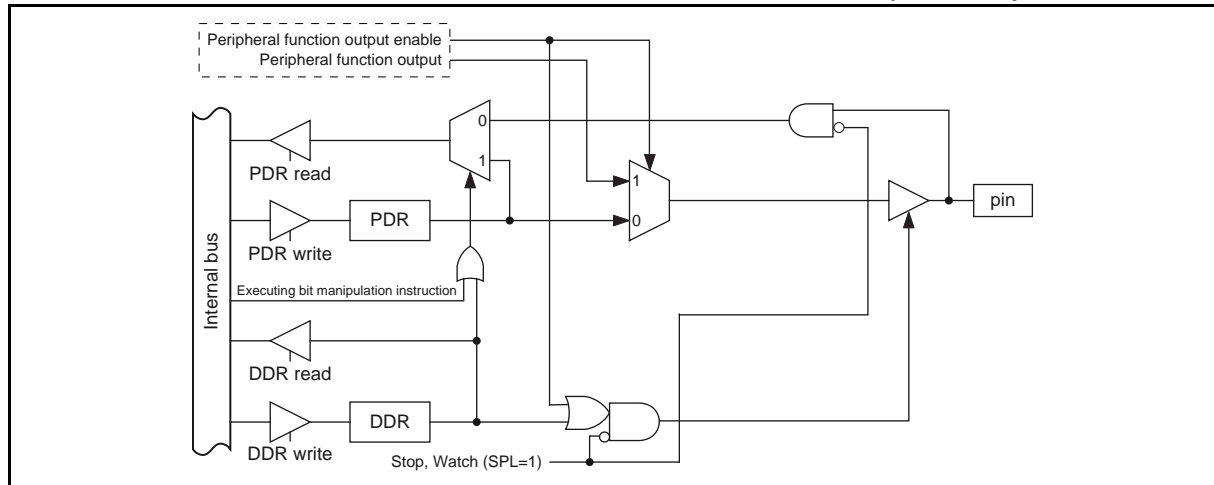
Figure 24.3-2 Block Diagram of Pins OPT2 and OPT5 (P64/EC1/PPG10/OPT2 and P67/TRG1/PPG21/OPT5) of Multi-pulse Generator
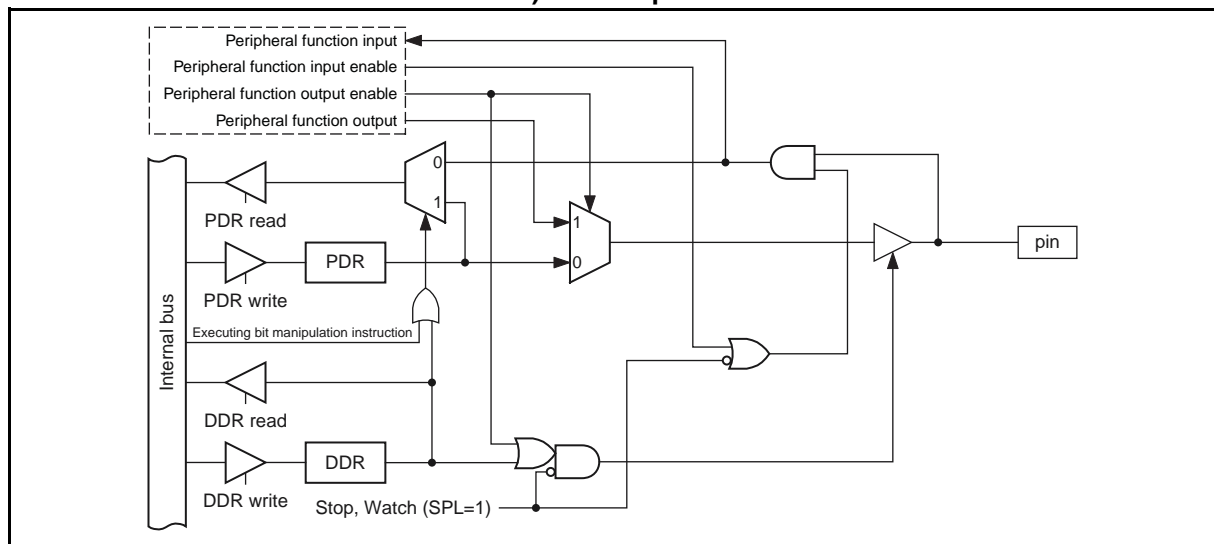
Figure 24.3-3 Block Diagram of Pins DTTI and TI1 (P60/INT08/SDA/DTTI and P61/INT09/SCL/TI1) of Multi-pulse Generator
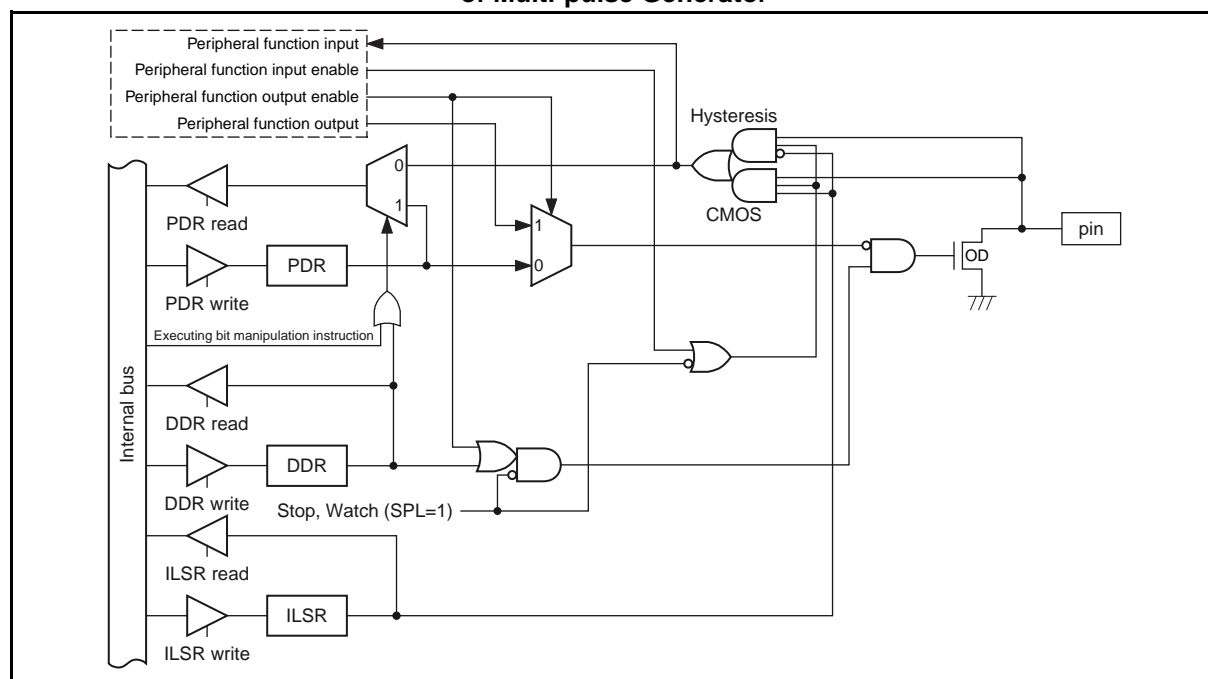
Figure 24.3-4 Block Diagram of Pins SNI1 and SNI2 (PG1/X0A/SNI1 and PG2/X1A/SNI2) of Multi-pulse Generator

MB95330H Series
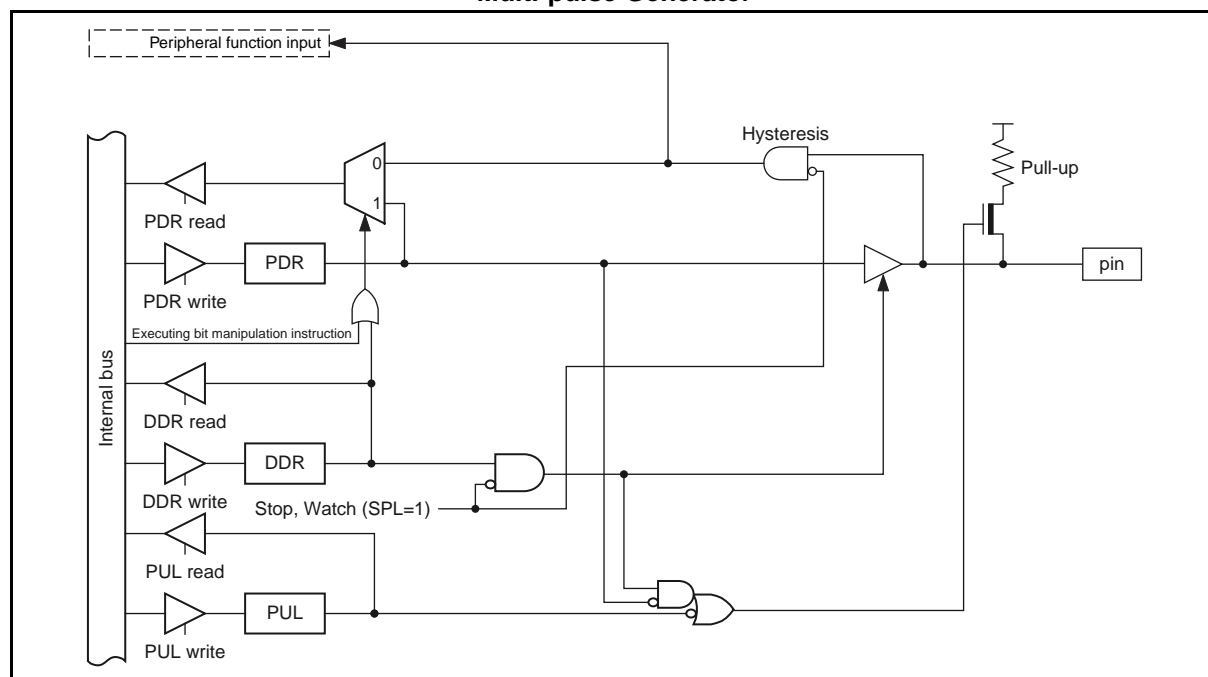
Figure 24.3-5 Block Diagram of Pin SNI0 (P17/TO1/SNI0) of Multi-pulse Generator
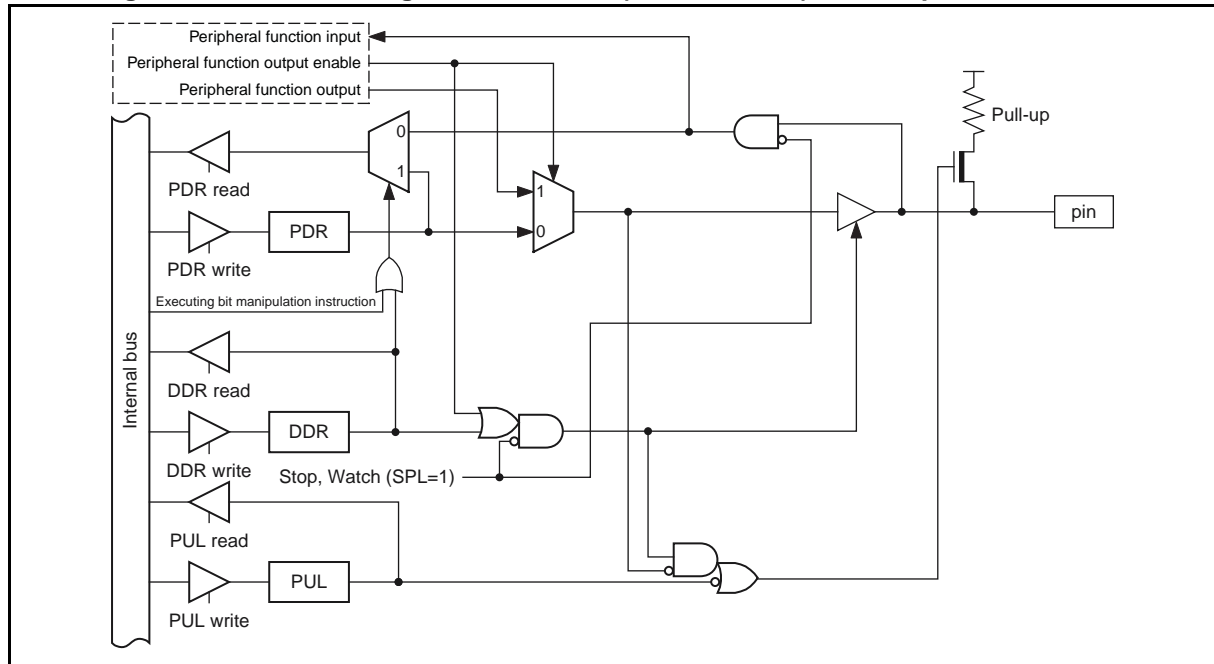
24.4 Registers of Multi-pulse Generator

This section describes the registers of the multi-pulse generator.

■ Registers of Multi-pulse Generator

Figure 24.4-1 Registers of Multi-pulse Generator

Output control register (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPCUR	0066$_H$	DTIE	DTIF	NRSL	OPS2	OPS1	OPS0	WTIF	WTIE	00000000$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output control register (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPCLR	0067$_H$	PDIF	PDIE	OPE5	OPE4	OPE3	OPE2	OPE1	OPE0	00000000$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output data register (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDUR	0FDC$_H$	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	0000XXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Output data register (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDLR	0FDD$_H$	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Output data buffer registers (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDBRHB - OPDBRH0	0FC4$_H$ - 0FDA$_H$	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	00000000$_B$
(Even addresses)		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output data buffer registers (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDBRLB - OPDBRL0	0FC5$_H$ - 0FDB$_H$	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	00000000$_B$
(Odd addresses)		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

(Continued)

MB95330H Series

(Continued)

Input control register (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
IPCUR	0068_H	WTS1	WTS0	CPIF	CPIE	CPD2	CPD1	CPD0	CMPE	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Input control register (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
IPCLR	0069_H	CPE1	CPE0	SNC2	SNC1	SNC0	SEE2	SEE1	SEE0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare clear register (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCUR	0FDE_H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	XXXXXXXX_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare clear register (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCLR	0FDF_H	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	XXXXXXXX_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer buffer register (upper)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBUR	0FE2_H	T15	T14	T13	T12	T11	T10	T09	T08	XXXXXXXX_B
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Timer buffer register (lower)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBLR	0FE3_H	T07	T06	T05	T04	T03	T02	T01	T00	XXXXXXXX_B
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Timer control status register

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TCSR	006B_H	TCLR	MODE	ICLR	ICRE	TMEN	CLK2	CLK1	CLK0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Noise cancellation control register

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
NCCR	006A_H	S21	S20	S11	S10	S01	S00	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.)
R/WX : Read only (Readable. Writing a value to it has no effect on operation.)
X   : Indeterminate

24.4.1 Output Control Register (OPCUR, OPCLR)

The output control register is composed of two 8-bit registers (OPCUR, OPCLR), which enable the write timing interrupt and flag, position detect interrupt and flag, set the data transfer method, and set the control of the OPT5 to OPT0 and DTTI pins. OPCUR is the upper byte register and OPCLR the lower byte register.

■ Output Control Register Upper (OPCUR)

Figure 24.4-2 Output Control Register Upper (OPCUR)

Address	bit	7	6	5	4	3	2	1	0	Initial value
0066H		DTIE	DTIF	NRSL	OPS2	OPS1	OPS0	WTIF	WTIE	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WTIE	Write timing interrupt enable bit
0	Disables interrupts.
1	Enables interrupts.

WTIF	Write timing interrupt request flag bit	
	Read	Write
0	No write timing interrupt has been generated.	Clears this bit.
1	A write timing interrupt has been generated.	No effect.

OPS2	OPS1	OPS0	Function
0	0	0	Data transfer from OPDBRH0 and OPDBRL0 to OPDUR and OPDLR after both OPDBRH0 and OPDBRL0 are written by software.
0	0	1	Data transfer from OPDBRH0 and OPDBRL0 to OPDUR and OPDLR is triggered by the 16-bit reload timer underflow.
0	1	0	Data transfer from OPDBRH0 and OPDBRL0 to OPDUR and OPDLR is triggered by the position detection input.
0	1	1	Data transfer from OPDBRH0 and OPDBRL0 to OPDUR and OPDLR is triggered by the write signal generated by the 16-bit reload timer underflow; the 16-bit timer is started by the position detection comparison circuit.
1	0	0	Data transfer from OPDBRH0 and OPDBRL0 to OPDUR and OPDLR is triggered by the write signal generated either by the 16-bit reload timer underflow or by the position detection input.
1	0	1	One-shot position detection or timer underflow
1	1	0	One-shot position detection
1	1	1	One-shot position detection and timer underflow

NRSL	Noise filter enable bit
0	DTTI input does not go through the noise filter.
1	DTTI input goes through the noise filter.

DTIF	DTTI interrupt request flag bit	
	Read	Write
0	No valid edge has been detected.	Clears this bit.
1	A valid edge has been detected.	No effect on operation.

DTIE	DTTI control enable bit
0	Disables control by DTTI input.
1	Enables control by DTTI input.

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

▨ : Initial value

Table 24.4-1 Functions of Bits in Output Control Register Upper (OPCUR) Bits

Bit name		Function
bit7	DTIE: DTTI control enable bit	• DTTI pin input enable bit. • This bit is used to enable the DTT1 pin to control the output levels of the OPT5 to OPT0 pins. The software can set the inactive level for each OPTx pin in PDRx of PORTx.
bit6	DTIF: DTTI interrupt request flag bit	• DTTI interrupt request flag. • It is an interrupt request flag of the DTTI input, which is set whenever a falling edge of DTTI is detected and the DTTI control enable bit is set to "1". • When this bit is set to "1", the interrupt is generated. This bit is cleared by writing "0". Writing "1" has no effect on operation. • In read-modify-write operation, "1" is always read.
bit5	NRSL: Noise filter enable bit	• This bit is used to select the noise cancellation function when DTTI pin input is enabled. • The noise cancellation circuit starts the internal n-bit counter when an active level is input (the value of n can be 2, 3, 4 or 5, which depends on the setting of D1,D0 bits in the noise cancellation control register). If the active level is held until the counter overflows, the circuit accepts input from the DTTI pin. Therefore, the pulse width of noise that can be cancelled is about 2^n machine cycles. Note: When the noise cancellation circuit is enable, the input becomes invalid in a mode such as STOP mode in which the internal clock is stopped.
bit4 to bit2	OPS2 to OPS0: Data transfer method select bits	• OPTx pin output timing control selection bits. • These bits are used to select the OPDUR and OPDLR register write timing control operation mode. Data is transferred from the output data buffer register to the output data register at the write timing controlled by the selected operation mode.
bit1	WTIF: Write timing interrupt request flag bit	• Write timing interrupt request flag. • It is an interrupt request flag of the output timing switch, which is set by the write signal. Data in the OPDBRHx and OPDBRLx registers that are specified by the BNKF, RDA2 to RDA0 bits in the output data register upper (OPDUR) is transferred to OPDUR and OPDLR at the rising edge of the write signal and the WTIF bit is set to "1". • When this bit is set to "1", the interrupt is generated if the write timing interrupt enable bit (WTIE) is also set to "1". This bit is cleared by writing "0". Writing "1" has no effect on operation. • In read-modify-write operation, "1" is always read.
bit0	WTIE: Write timing interrupt enable bit	• Write timing interrupt enable bit. • When this bit is set to "1", the interrupt is generated if write timing interrupt request flag bit (WTIF) is also set to "1".

■ Output Control Register Lower (OPCLR)

Figure 24.4-3 Output Control Register Lower (OPCLR)

Address	bit	7	6	5	4	3	2	1	0	Initial value
0067H		PDIF	PDIE	OPE5	OPE4	OPE3	OPE2	OPE1	OPE0	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

OPE0	OPT0 output enable bit
0	Disables OPT0 pin output.
1	Enables OPT0 pin output.

OPE1	OPT1 output enable bit
0	Disables OPT1 pin output.
1	Enables OPT1 pin output.

OPE2	OPT2 output enable bit
0	Disables OPT2 pin output.
1	Enables OPT2 pin output.

OPE3	OPT3 output enable bit
0	Disables OPT3 pin output.
1	Enables OPT3 pin output.

OPE4	OPT4 output enable bit
0	Disables OPT4 pin output.
1	Enables OPT4 pin output.

OPE5	OPT5 output enable bit
0	Disables OPT5 pin output.
1	Enables OPT5 pin output.

PDIE	Position detection interrupt enable bit
0	Disables interrupts.
1	Enables interrupts.

PDIF	Position detection interrupt request flag bit	
	Read	Write
0	No position detection interrupt has been generated.	Clears this bit.
1	A position detection interrupt has been generated.	No effect.

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

☐ : Initial value

Table 24.4-2 Functions of Bits in Output Control Register Lower (OPCLR) Bits

Bit name		Function
bit7	PDIF: Position detection interrupt request flag bit	• Position detection interrupt request flag. • It is an interrupt request flag for the position detection. When CMPE is set to "1" and the SNI2 to SNI0 bits are compared and matched with the RDA2 to RDA0 bits, or when CMPE is set to "0" and any effective edge is detected at SNI2 to SNI0 pins, this bit is set to "1". • When this bit is set to "1", the interrupt is generated if the position detection interrupt enable bit (PDIE) is also set to "1". This bit is cleared by writing "0". Writing "1" to it has no effect on operation. • In read-modify-write operation, "1" is always read.
bit6	PDIE: Position detection interrupt enable bit	• Position detection interrupt enable bit. • When this bit is set to "1", the interrupt is generated if position detection interrupt request flag (PDIF) is also set to "1".
bit5 to bit0	OPE5 to OPE0: OPT5 to OPE0 output enable bits	• Output enable bits of OPT5 to OPE0 pins. • When these bits are set, the outputs to the OPT5 to OPE0 pins are enable.

24.4.2   Output Data Register (OPDUR, OPDLR)

The output data register is composed of two 8-bit registers (OPDUR, OPDLR), which store the output data to the OPT5 to OPT0 pins. OPDUR is the upper byte register and OPDLR the lower byte register.

These are two 8-bit registers used to read the output data register value.

Always use one of the following procedures to read these registers.

- Use the "MOVW" instruction (use a 16-bit access instruction to read the OPDUR register address).

- Use the "MOV" instruction and read OPDUR first and then OPDLR (OPDLR will be updated when OPDUR is read).

MB95330H Series

■ **Output Data Register Upper (OPDUR)**

Figure 24.4-4 Output Data Register Upper (OPDUR)

Address	bit 7	6	5	4	3	2	1	0	Initial value
0FDC$_H$	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	0000XXXX$_B$
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

OP41	OP40	OPT4 output waveform select bits
0	0	Pin OPT4 outputs "L" level.
0	1	Pin OPT4 outputs the output of the PPG timer.
1	0	Pin OPT4 outputs the inverted output of the PPG timer.
1	1	Pin OPT4 outputs "H" level.

OP51	OP50	OPT5 output waveform select bits
0	0	Pin OPT5 outputs "L" level.
0	1	Pin OPT5 outputs the output of the PPG timer.
1	0	Pin OPT5 outputs the inverted output of the PPG timer.
1	1	Pin OPT5 outputs "H" level.

BNKF	RDA2	RDA1	RDA0	OPDBRHx and OPDBRLx register select bits
0	0	0	0	Data in OPDBRH0 and OPDBRL0 is loaded to OPDUR and OPDLR.
0	0	0	1	Data in OPDBRH1 and OPDBRL1 is loaded to OPDUR and OPDLR.
0	0	1	0	Data in OPDBRH2 and OPDBRL2 is loaded to OPDUR and OPDLR.
0	0	1	1	Data in OPDBRH3 and OPDBRL3 is loaded to OPDUR and OPDLR.
0	1	0	0	Data in OPDBRH4 and OPDBRL4 is loaded to OPDUR and OPDLR.
0	1	0	1	Data in OPDBRH5 and OPDBRL5 is loaded to OPDUR and OPDLR.
0	1	1	0	Data in OPDBRH6 and OPDBRL6 is loaded to OPDUR and OPDLR.
0	1	1	1	Data in OPDBRH7 and OPDBRL7 is loaded to OPDUR and OPDLR.
1	0	0	0	Data in OPDBRH8 and OPDBRL8 is loaded to OPDUR and OPDLR.
1	0	0	1	Data in OPDBRH9 and OPDBRL9 is loaded to OPDUR and OPDLR.
1	0	1	0	Data in OPDBRHA and OPDBRLA is loaded to OPDUR and OPDLR.
1	0	1	1	Data in OPDBRHB and OPDBRLB is loaded to OPDUR and OPDLR.
Other values				Setting prohibited

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

Table 24.4-3 Functions of Bits in Output Data Register Upper (OPDUR) Bits

Bit name		Function
bit7 to bit4	BNKF, RDA2 to RDA0: OPDBRHx and OPDBRLx registers select bits	• These bits indicate the addresses of the OPDBRHx and OPDBRLx registers and decide which output data buffer register value is loaded into the OPDUR and OPDLR registers.
bit3, bit2	OP51, OP50: OPT5 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT5 pin.
bit1, bit0	OP41, OP40: OPT4 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT4 pin.

MB95330H Series

■ **Output Data Register Lower (OPDLR)**

Figure 24.4-5 Output Data Register Lower (OPDLR)

OP01	OP00	OPT0 output waveform select bits
0	0	Pin OPT0 outputs "L" level.
0	1	Pin OPT0 outputs the output of the PPG timer.
1	0	Pin OPT0 outputs the inverted output of the PPG timer.
1	1	Pin OPT0 outputs "H" level.

OP11	OP10	OPT1 output waveform select bits
0	0	Pin OPT1 outputs "L" level.
0	1	Pin OPT1 outputs the output of the PPG timer.
1	0	Pin OPT1 outputs the inverted output of the PPG timer.
1	1	Pin OPT1 outputs "H" level.

OP21	OP20	OPT2 output waveform select bits
0	0	Pin OPT2 outputs "L" level.
0	1	Pin OPT2 outputs the output of the PPG timer.
1	0	Pin OPT2 outputs the inverted output of the PPG timer.
1	1	Pin OPT2 outputs "H" level.

OP31	OP30	OPT3 output waveform select bits
0	0	Pin OPT3 outputs "L" level.
0	1	Pin OPT3 outputs the output of the PPG timer.
1	0	Pin OPT3 outputs the inverted output of the PPG timer.
1	1	Pin OPT3 outputs "H" level.

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

X : Indeterminate

Table 24.4-4 Functions of Bits in Output Data Register Lower (OPDLR) Bits

Bit name		Function
bit7, bit6	OP31, OP30: OPT3 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT3 pin.
bit5, bit4	OP21, OP20: OPT2 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT2 pin.
bit3, bit2	OP11, OP10: OPT1 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT1 pin.
bit1, bit0	OP01, OP00: OPT0 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT0 pin.

MB95330H Series

24.4.3   Output Data Buffer Register (OPDBRH, OPDBRL)

The output data buffer register is composed of twelve pairs of registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0). OPDBRHx is the upper byte register and OPDBRLx the lower byte register. The values of the OPDBRHx and OPDBRLx registers specified by the BNKF, RDA2 to RDA0 bits are loaded into the OPDUR and OPDLR registers at the rising edge of the write signal generated by the Data Write Control Unit.

■ **Output Data Buffer Register Upper (OPDBRH)**

Figure 24.4-6 Output Data Buffer Register Upper (OPDBRH)

Address	bit	7	6	5	4	3	2	1	0	Initial value
0FC4H to OFDAH		BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

OP41	OP40	OPT4 output waveform select bits
0	0	Setting for OPT4 pin to output "L" level.
0	1	Setting for OPT4 pin to output the output of the PPG timer.
1	0	Setting for OPT4 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT4 pin to output "H" level.

OP51	OP50	OPT5 output waveform select bits
0	0	Setting for OPT5 pin to output "L" level.
0	1	Setting for OPT5 pin to output the output of the PPG timer.
1	0	Setting for OPT5 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT5 pin to output "H" level.

BNKF	RDA2	RDA1	RDA0	OPDBRH and OPDBRL registers select bits
0	0	0	0	Set OPDBRH0 and OPDBRL0 as the next registers to be loaded into OPDUR and OPDLR.
0	0	0	1	Set OPDBRH1 and OPDBRL1 as the next registers to be loaded into OPDUR and OPDLR.
0	0	1	0	Set OPDBRH2 and OPDBRL2 as the next registers to be loaded into OPDUR and OPDLR.
0	0	1	1	Set OPDBRH3 and OPDBRL3 as the next registers to be loaded into OPDUR and OPDLR.
0	1	0	0	Set OPDBRH4 and OPDBRL4 as the next registers to be loaded into OPDUR and OPDLR.
0	1	0	1	Set OPDBRH5 and OPDBRL5 as the next registers to be loaded into OPDUR and OPDLR.
0	1	1	0	Set OPDBRH6 and OPDBRL6 as the next registers to be loaded into OPDUR and OPDLR.
0	1	1	1	Set OPDBRH7 and OPDBRL7 as the next registers to be loaded into OPDUR and OPDLR.
1	0	0	0	Set OPDBRH8 and OPDBRL8 as the next registers to be loaded into OPDUR and OPDLR.
1	0	0	1	Set OPDBRH9 and OPDBRL9 as the next registers to be loaded into OPDUR and OPDLR.
1	0	1	0	Set OPDBRHA and OPDBRLA as the next registers to be loaded into OPDUR and OPDLR.
1	0	1	1	Set OPDBRHB and OPDBRLB as the next registers to be loaded into OPDUR and OPDLR.
Other values				Setting prohibited

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

: Initial value

Table 24.4-5 Functions of Bits in Output Data Buffer Register Upper (OPDBRH) Bits

Bit name		Function
bit7 to bit4	BNKF, RDA2 to RDA0: OPDBRH and OPDBRL registers select bits	• These bits are used to select the next OPDBRHx and OPDBRLx registers whose values will be loaded into the OPDUR and OPDLR registers.
bit3, bit2	OP51, OP50: OPT5 output waveform select bits	• These bits are used to select the kind of the output waveform to be output to the OPT5 pin after the values of the output data buffer register upper and output data buffer register chosen are loaded into the OPDUR and OPDLR registers.
bit1, bit0	OP41, OP40: OPT4 output waveform select bits	• These bits are used to select the kind of the output waveform to be output to the OPT4 pin after the values of the output data buffer register upper and output data buffer register chosen are loaded into the OPDUR and OPDLR registers.

MB95330H Series

■ Output Data Buffer Register Lower (OPDBRL)

Figure 24.4-7 Output Data Buffer Register Lower (OPDBRL)

Address	bit 7	6	5	4	3	2	1	0	Initial value
0FC5H to 0FDBH	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

OP01	OP00	OPT0 output waveform select bits
0	0	Setting for OPT0 pin to output "L" level.
0	1	Setting for OPT0 pin to output the output of the PPG timer.
1	0	Setting for OPT0 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT0 pin to output "H" level.

OP11	OP10	OPT1 output waveform select bits
0	0	Setting for OPT1 pin to output "L" level.
0	1	Setting for OPT1 pin to output the output of the PPG timer.
1	0	Setting for OPT1 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT1 pin to output "H" level.

OP21	OP20	OPT2 output waveform select bits
0	0	Setting for OPT2 pin to output "L" level.
0	1	Setting for OPT2 pin to output the output of the PPG timer.
1	0	Setting for OPT2 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT2 pin to output "H" level.

OP31	OP30	OPT3 output waveform select bits
0	0	Setting for OPT3 pin to output "L" level.
0	1	Setting for OPT3 pin to output the output of the PPG timer.
1	0	Setting for OPT3 pin to output the inverted output of the PPG timer.
1	1	Setting for OPT3 pin to output "H" level.

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

Table 24.4-6 Functions of Bits in Output Data Buffer Register Lower (OPDBRL) Bits

Bit name		Function
bit7, bit6	OP31, OP30: OPT3 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT3 pin after the values of the OPDBRHx and OPDBRLx registers chosen are loaded into the OPDUR and OPDLR registers.
bit5, bit4	OP21, OP20: OPT2 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT2 pin after the values of the OPDBRHx and OPDBRLx registers chosen are loaded into the OPDUR and OPDLR registers.
bit3, bit2	OP11, OP10: OPT1 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT1 pin after the values of the OPDBRHx and OPDBRLx registers chosen are loaded into the OPDUR and OPDLR registers.
bit1, bit0	OP01, OP00: OPT0 output waveform select bits	• These bits are used to select the kind of the output waveform to the OPT0 pin after the values of the OPDBRHx and OPDBRLx registers chosen are loaded into the OPDUR and OPDLR registers.

MB95330H Series

24.4.4 Input Control Register (IPCUR, IPCLR)

The input control register is composed of two 8-bit registers (IPCUR, IPCLR), which are used to control position detection inputs. IPCUR is the upper byte register and IPCLR the lower byte register.

■ **Input Control Register Upper (IPCUR)**

Figure 24.4-8 Input Control Register Upper (IPCUR)

Address	bit	7	6	5	4	3	2	1	0	Initial value
0068H		WTS1	WTS0	CPIF	CPIE	CPD2	CPD1	CPD0	CMPE	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CMPE	Position detection comparison enable bit
0	Disables comparison operation. (Initial value)
1	Enables comparison operation.

CPD2	CPD1	CPD0	Comparison bits
0	0	0	Compare match if RDA2 to RDA0 = 000.
0	0	1	Compare match if RDA2 to RDA0 = 001.
0	1	0	Compare match if RDA2 to RDA0 = 010.
0	1	1	Compare match if RDA2 to RDA0 = 011.
1	0	0	Compare match if RDA2 to RDA0 = 100.
1	0	1	Compare match if RDA2 to RDA0 = 101.
1	1	0	Compare match if RDA2 to RDA0 = 110.
1	1	1	Compare match if RDA2 to RDA0 = 111.

CPIE	Comparison interrupt request enable bit
0	Disable interrupt. (Initial value)
1	Enable interrupt.

CPIF	Comparison interrupt request flag bit	
	Read	Write
0	No comparison interrupt has been generated.	Clears this bit.
1	A comparison interrupt has been generated.	No effect.

WTS1	WTS0	PPG edge synchronization select bits
0	0	No synchronization. (Initial value)
0	1	Rising edge synchronization. ↑
1	0	Falling edge synchronization. ↓
1	1	Both edges synchronization. ↑ & ↓

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

▨ : Initial value

Table 24.4-7 Functions of Bits in Input Control Register Upper (IPCUR) Bits

Bit name		Function
bit7, bit6	WTS1, WTS0: PPG edge synchronization select bits	• These bits are used to select the synchronization edge of the next coming of PPG signal with the write timing.
bit5	CPIF: Comparison interrupt request flag bit	• Comparison interrupt request flag. • It is a comparison interrupt request flag for the comparison circuit. When the RDA2 to RDA0 bits are compared and matched with the CPD2 to CPD0 bits, this bit is set to "1". • When comparison interrupt enable bit (CPIE) is also set to "1", the interrupt is generated. • This bit is cleared by writing "0". Writing "1" has no effect on operation. • In read-modify-write operation, "1' is always read.
bit4	CPIE: Comparison interrupt request enable bit	• Comparison interrupt enable bit. • When this bit is set to "1" and the comparison interrupt request flag (CPIF) is also set to "1", the interrupt is generated.
bit3 to bit1	CPD2 to CPD0: Comparison bits	• These bits are used to compare with the RDA2 to RDA0 bits of the output data register, when the value of these bits are matched with the value of RDA2 to RDA0 bits, the compare interrupt flag (CPIF) is set to "1".
bit0	CMPE: Position detection comparison enable bit	• This bit is used to enable the comparison operation for the position detection.

MB95330H Series

■ Input Control Register Lower (IPCLR)

Figure 24.4-9 Input Control Register Lower (IPCLR)

Address	bit 7	6	5	4	3	2	1	0	Initial value
0069H	CPE1	CPE0	SNC2	SNC1	SNC0	SEE2	SEE1	SEE0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SEE0	SNI0 enable bit
0	Disable SNI0 edge detection. (Initial value)
1	Enable SNI0 edge detection.

SEE1	SNI1 enable bit
0	Disable SNI1 edge detection. (Initial value)
1	Enable SNI1 edge detection.

SEE2	SNI2 enable bit
0	Disable SNI2 edge detection. (Initial value)
1	Enable SNI2 edge detection.

SNC0	Noise filter enable bit for SNI0
0	SNI0 input do not go through the noise cancellation circuit.
1	SNI0 input goes through the noise cancellation circuit.

SNC1	Noise filter enable bit for SNI1
0	SNI1 input do not go through the noise cancellation circuit.
1	SNI1 input goes through the noise cancellation circuit.

SNC2	Noise filter enable bit for SNI2
0	SNI2 input do not go through the noise cancellation circuit.
1	SNI2 input goes through the noise cancellation circuit.

CPE1	CPE0	Input edge polarity select bits
0	0	No edge detection. (stopped state) (Initial value)
0	1	Rising edge detection. ↑
1	0	Falling edge detection. ↓
1	1	Both edges detection. ↑ & ↓

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

▨ : Initial value

Table 24.4-8 Functions of Bits in Input Control Register Lower (IPCLR) Bits

Bit name		Function
bit7, bit6	CPE1, CPE0: Input edge polarity select bits	• Input edge polarity select bits. • These bits are used to select the polarity of the input edge for the position detection, the position detection operates according to the input edge polarity set to these bits.
bit5 to bit3	SNC2 to SNC0: Noise filter enable bits for SNI2 to SNI0	• These bits are used to select the noise cancellation function when the inputs of the pins SNI2 to SNI0 are enable. • The noise cancellation circuit starts the internal n-bit counter when an active level is input (the value of n can be 2, 3, 4, 5, which depends on the setting of S21,S20, S11,S10 and S01,S00 bits in the noise cancellation control register). If the active level is held until the counter overflows, the circuit accepts input from the SNI2 to SNI0 pins. Therefore, the pulse width of noise that can be cancelled is about 2^n machine cycles. Note: When the noise cancellation circuit is enable, the input becomes invalid in a mode such as STOP mode in which the internal clock is stopped.
bit2 to bit0	SEE2 to SEE0: SNI2 to SNI0 enable bits	• Pins SNI2 to SNI0 edge detection enable bits. • When they are set to "1", the edge detection of the pins SNI2 to SNI0 are enable. • Please set these bits before setting CMPE in the input control register upper to "1".

MB95330H Series

24.4.5   Compare Clear Register (CPCUR, CPCLR)

The compare clear register is composed of two 8-bit registers (CPCUR, CPCLR). CPCUR is the upper byte register and CPCLR the lower byte register. When the values of these registers match the count value of the 16-bit timer, the 16-bit timer is reset to "0000$_H$".

■ Compare Clear Register (CPCUR, CPCLR)

These two register are 8-bit registers used to hold the compare clear register value.

Always use one of the following procedures to read and write these registers.

- Use the "MOVW" instruction (use a 16-bit access instruction to read and write the CPCUR register address).

- Use the "MOV" instruction, and read or write CPCUR first and then CPCLR.

The compare clear register upper and the compare clear register lower are two 8-bit registers and are compared with the count value of the 16-bit timer. The initial values of these registers are indeterminate, and therefore it is necessary to write specific values to these registers before starting an operation.

Notes:

To access these registers, the word access instruction must be used.
When the values of these registers match the count value of the 16-bit timer, the 16-bit timer is reset to "0000$_H$" and the compare clear interrupt request flag is set. In addition, when the interrupt operation is enabled, an interrupt request is sent to the CPU.
If the values loaded to the compare clear register upper (CPCUR) and the compare clear register lower (CPCLR) are the same as the timer counter value, the comparison operation will NOT be performed until the next occasion in which the values of CPCUR and CPCLR are the same as the timer counter value.

Figure 24.4-10 Compare Clear Register (CPCUR, CPCLR)

Compare clear register (upper)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCUR	0FDE$_H$	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	XXXXXXXX$_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare clear register (lower)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCLR	0FDF$_H$	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	XXXXXXXX$_B$
		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.)
X : Indeterminate

24.4.6 Timer Buffer Register (TMBUR, TMBLR)

The timer buffer register is composed of two 8-bit registers (TMBUR, TMBLR), which are used to read the count value of 16-bit timer. TMBUR is the upper byte register and TMBLR the lower byte register.

■ Timer Buffer Register (TMBUR,TMBLR)

These two registers are 8-bit registers used to hold the timer buffer register value.

Always use one of the following procedures to read this register.

- Use the "MOVW" instruction (use a 16-bit access instruction to read the TMBUR register address).

- Use the "MOV" instruction, and read or write TMBUR first and then TMBLR.

The timer buffer register upper and the timer buffer register lower are used to store the count value of the 16-bit timer at the moment when a write timing or position detection trigger is generated, and the counter is then cleared to "0000_H".

Note:
Use only the word access instruction to access TMBUR and TMBLR.

Figure 24.4-11 Timer Buffer Register (TMBUR,TMBLR)

Timer buffer register (upper)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBUR	0FE2$_H$	T15	T14	T13	T12	T11	T10	T09	T08	XXXXXXXX$_B$
		R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

Timer buffer register (lower)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBLR	0FE3$_H$	T07	T06	T05	T04	T03	T02	T01	T00	XXXXXXXX$_B$
		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.)
X : Indeterminate

MB95330H Series

24.4.7   Timer Control Status Register (TCSR)

The timer control status register (TCSR) is used to control the operation of the 16-bit timer.

■ **Timer Control Status Register (TCSR)**

Figure 24.4-12 Timer Control Status Register (TCSR)

Address	bit	7	6	5	4	3	2	1	0	Initial value
006BH		TCLR	MODE	ICLR	ICRE	TMEN	CLK2	CLK1	CLK0	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CLK2	CLK1	CLK0	Clock frequency select bits				
			Count clock	MCLK 16 MHz	MCLK 8 MHz	MCLK 4 MHz	MCLK 1 MHz
0	0	0	MCLK	62.5 ns	125 ns	0.25 μs	1 μs
0	0	1	MCLK/2	125 ns	0.25 μs	0.5 μs	2 μs
0	1	0	MCLK/4	0.25 μs	0.5 μs	1 μs	4 μs
0	1	1	MCLK/8	0.5 μs	1 μs	2 μs	8 μs
1	0	0	MCLK/16	1 μs	2 μs	4 μs	16 μs
1	0	1	MCLK/32	2 μs	4 μs	8 μs	32 μs
1	1	0	MCLK/64	4 μs	8 μs	16 μs	64 μs
1	1	1	MCLK/128	8 μs	16 μs	32 μs	128 μs

MCLK: machine clock

TMEN	Timer enable bit
0	Disables counting.
1	Enables counting.

ICRE	Compare clear interrupt request enable bit
0	Disables interrupts.
1	Enables interrupts.

ICLR	Compare clear interrupt request flag bit	
	Read	Write
0	No interrupt request.	Clears this bit.
1	There is an interrupt request.	No effect.

MODE	Timer reset condition bit
0	The timer is to be reset by the write timing trigger.
1	The timer is to be reset by the position detection trigger.

TCLR	Timer clear bit	
	Read	Write
0	Always reads "0".	No effect.
1		Initializes the counter to "0000H".

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

Table 24.4-9 Functions of Bits in Timer Control Status Register (TCSR)

Bit name		Function
bit7	TCLR: Timer clear bit	• The read value is always "0". • Writing "1" to this bit initialize the counter to "0000$_H$". • Writing "0" has no effect on operation.
bit6	MODE: Timer reset condition bit	• This bit is used to set the reset condition for the 16-bit timer. • When it is "0", 16-bit timer is reset by the write timing signal. • When it is "1", 16-bit timer is reset by the position detection signal. Note: Reset of the timer value is done at the changing point of the timer value.
bit5	ICLR: Compare clear interrupt request flag bit	• This bit is an interrupt request flag for compare clear. • When the compare clear register and 16-bit timer value are matched, the counter is cleared and this bit becomes "1". • Interrupt is generated when the interrupt request enable bit (bit12:ICRE) is set to "1". • Writing "0" clears this bit. • Writing "1" has no effect on operation. • In read-modify-write operation, "1" is always read.
bit4	ICRE: Compare clear interrupt request enable bit	• This is the interrupt request enable bit for the compare clear. • When this bit is "1" and the interrupt flag (bit13:ICLR) is set to "1", an interrupt is generated.
bit3	TMEN: Timer enable bit	• This bit is used to enable/disable the counting of the 16-bit timer. • Writing "1" to this bit enables the counting of the 16-bit timer. • Writing "0" to this bit disables the counting of the 16-bit timer. (Note) When the 16-bit timer is disable, the output compare operation is also disabled.
bit2 to bit0	CLK2 to CLK0: Clock frequency select bits	• These bits are used to select count clock for the 16-bit timer. Note: It is recommend to change these bits when the timer is in stop state because the clock is changed as soon as these bits are updated.

MB95330H Series

24.4.8   Noise Cancellation Control Register (NCCR)

The noise cancellation control register (NCCR) is used to control the noise pulse width to be cancelled for DTTI and SNIx pins.

■ Noise Cancellation Control Register (NCCR)

Figure 24.4-13 Noise Cancellation Control Register (NCCR)

D1	D0	DTTI noise width select bit
0	0	Cancels 4-cycle noise.
0	1	Cancels 8-cycle noise.
1	0	Cancels 16-cycle noise.
1	1	Cancels 32-cycle noise.

S01	S00	SNI0 noise width select bit
0	0	Cancels 4-cycle noise.
0	1	Cancels 8-cycle noise.
1	0	Cancels 16-cycle noise.
1	1	Cancels 32-cycle noise.

S11	S10	SNI1 noise width select bit
0	0	Cancels 4-cycle noise.
0	1	Cancels 8-cycle noise.
1	0	Cancels 16-cycle noise.
1	1	Cancels 32-cycle noise.

S21	S20	SNI2 noise width select bit
0	0	Cancels 4-cycle noise.
0	1	Cancels 8-cycle noise.
1	0	Cancels 16-cycle noise.
1	1	Cancels 32-cycle noise.

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

▨ : Initial value

Table 24.4-10Functions of Bits in Noise Cancellation Control Register (NCCR) Bits

Bit name		Function
bit7, bit6	S21,S20: Noise width select bits	• These bits are used to specify the noise pulse width to be removed for SNI2 pin.
bit5, bit4	S11,S10: Noise width select bits	• These bits are used to specify the noise pulse width to be removed for SNI1 pin.
bit3, bit2	S01,S00: Noise width select bits	• These bits are used to specify the noise pulse width to be removed for SNI0 pin.
bit1, bit0	D1,D0: Noise width select bits	• These bits are used to specify the noise pulse width to be removed for DTTI pin.

MB95330H Series

24.5   Interrupts of Multi-pulse Generator

The multi-pulse generator can generate an interrupt request due to the following sources:
- **Write timing output is generated by the Data Write Control Unit**
- **Any valid position detection input is detected**
- **Comparison match between CPD2 to CPD0 in the input control register upper (IPCUR:CPD2 to CPD0) and RDA2 to RDA0 in output data register upper (OPDUR:RDA2 to RDA0)**
- **Compare Clear is generated by the 16-bit Timer**
- **DTTI is changed to low signal level**

■ Multi-pulse Generator Interrupts

There are five interrupts generated from the multi-pulse generator as follows:

- Write Timing Interrupt
- Compare Clear Interrupt
- Position Detect Interrupt
- Compare Match Interrupt
- DTTI Interrupt

Write Timing Interrupt is multiplexed with Compare Clear Interrupt and Position Detect Interrupt is multiplexed with Compare Match Interrupt.

● Write Timing Interrupt

If the WTIE bit in the output control register upper (OPCUR) is set to "1", this Write Timing Interrupt is generated when the write timing is generated by the Data Write Control Circuit to make data transfer from one of 12 pairs of output data buffer registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0) to the output data register (OPDUR, OPDLR).

When this interrupt is generated, the write timing interrupt flag bit in the output control register upper (OPCUR:WTIF) is set to "1".

● Compare Clear Interrupt

If the ICRE bit in the timer control register (TCSR) is set to "1", this compare clear interrupt is generated when the compare value and the 16-bit timer value match.

When this interrupt is generated, the compare clear interrupt flag bit in the timer control register (TCSR:ICLR) is set to "1".

● Position Detect Timing Interrupt

If the PDIE bit in the output control register lower (OPCLR) is set to "1", this Position Detect Interrupt is generated when the write timing is output by the position detect circuit to make data transfer from one of 12 pairs of output data buffer registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0) to the output data register (OPDUR, OPDLR). This write timing output can be generated by either the compare match of the level of the position input (SNI2 to SNI0) with the RDA2 to RDA0 bits of the output data register upper (OPDUR), or a edge detected of the position input (SNI2 to SNI0) with one of 3 different kinds of edge setting.

When this interrupt is generated, the position detect interrupt flag bit in the output control register lower (OPCLR:PDIF) is set to "1".

● Compare Match Interrupt

If the CPIE bit in the input control register upper (IPCUR) is set to "1", this Compare Match Interrupt is generated when the RDA2 to RDA0 bits of the output data register upper (OPDUR) are matched with the CPD2 to CPD0 bits in the input control register upper (IPCUR).

When this interrupt is generated, the Compare match interrupt flag bit in the input control register upper (IPCUR:CPIF) is set to "1".

● DTTI Interrupt

If the DTIE bit in the output control register upper (OPCUR) is set to "1", this DTTI Interrupt is generated whenever a low input is detected at the DTTI pin.

When this interrupt is generated, the DTTI interrupt flag bit in the output control register upper (OPCUR:DTIF) is set to "1".

■ Multi-pulse Generator Interrupt Sources

IRQ04 : This interrupt is generated when a DTTI interrupt is happened.

DTTI interrupt is generated if OPCUR:DTIE is set to "1" when a low level input is detected at the DTTI pin.

IRQ16 : This interrupt is generated when either a Write Timing interrupt or Compare Clear interrupt is happened.

Write timing interrupt is generated if OPCUR:WTIE is set to "1" when a write timing signal is generated from the Data Write Control Circuit.

Compare clear interrupt is generated if TCSR:ICRE is set to "1" when the count value of 16-bit timer matches with the compare clear register (CPCUR, CPCLR).

IRQ17 : This interrupt is generated when either a Position Detect interrupt or Compare Match interrupt is happened.

Position detect interrupt is generated if OPCLR:PDIE is set to "1" when an effective edge at SNI2 to SNI0 is detected.

Compare match interrupt is generated if IPCUR:CPIE is set to "1" when the values of the CPD2 to CPD0 bits in the input control register upper (IPCUR) match with those of the RDA2 to RDA0 bits in the output data register upper (OPDUR).

MB95330H Series

■ Registers and Vector Table Addresses Related to Multi-pulse Generator Interrupts

Table 24.5-1 Registers and Vector Table Addresses Related to Multi-pulse Generator Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Lower	Upper
MPG (DTTI)	IRQ04	ILR1	L04	FFF3$_H$	FFF2$_H$
MPG (write timing/ compare clear)[*1]	IRQ16	ILR4	L16	FFDB$_H$	FFDA$_H$
MPG (position detection/compare match)[*2]	IRQ17	ILR4	L17	FFD9$_H$	FFD8$_H$

*1: The MPG (write timing/compare clear) shares the interrupt request number and vector table addresses mentioned in the table with 16-bit reload timer ch. 1 and I^{2}C.

*2: The MPG (position detection/compare match) shares the interrupt request number and vector table addresses mentioned in the table with 16-bit PPG timer ch. 1.

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

24.6 Operations of Multi-pulse Generator

The operations of the multi-pulse generator will be described in the following sections. According to the settings of the OPx1 and OPx0 bits in the output data register (OPDUR, OPDLR), the OPTx pin outputs the corresponding kind of waveforms ("H" or "L" or PPG output). See Table 24.6-1.

■ **Output Data Register Block Diagram**

Figure 24.6-1 Output Data Register Block Diagram
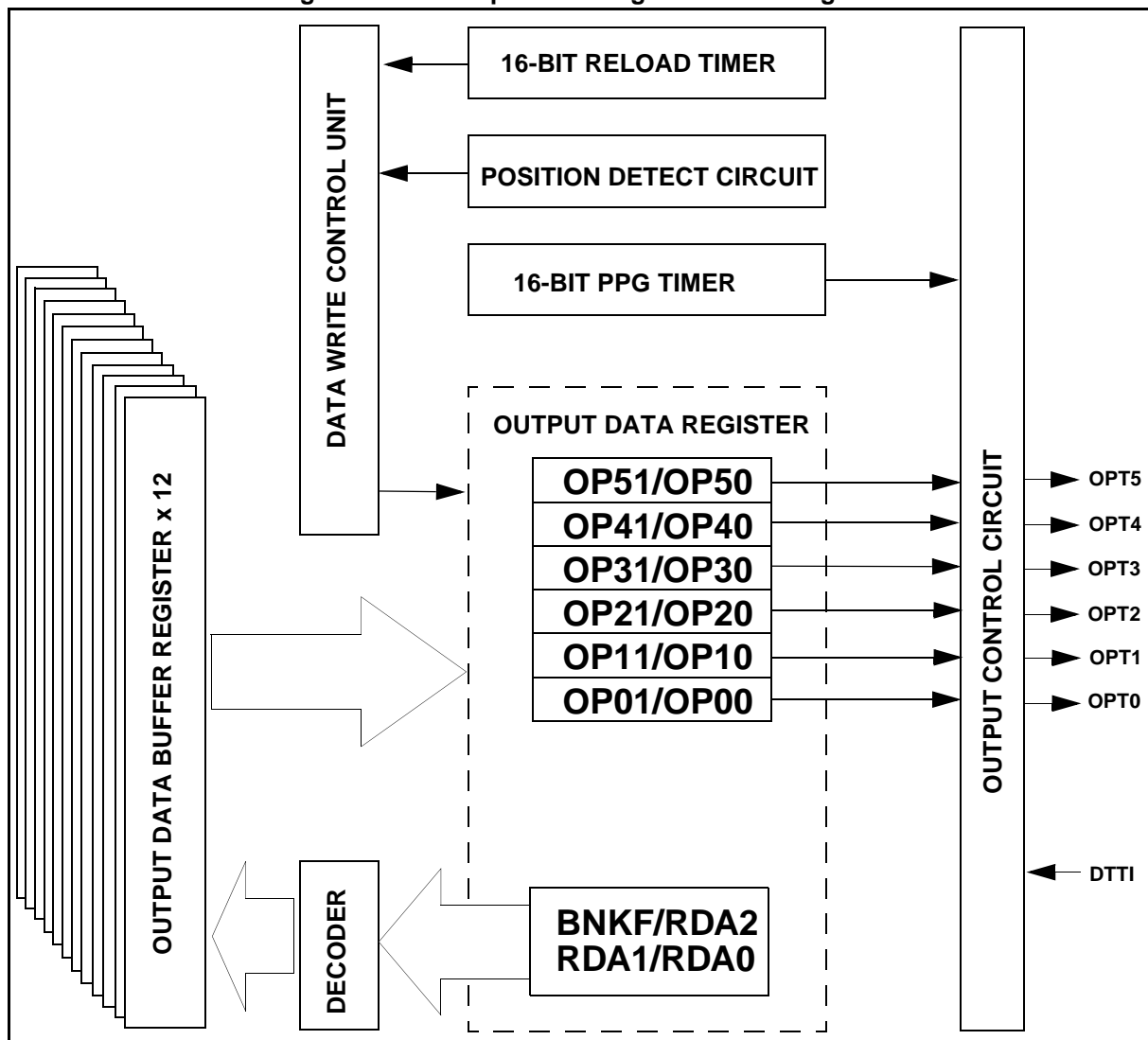
MB95330H Series

■ Output Data Register (OPDUR and OPDLR)

The content of the output data register (OPDUR, OPDLR) is sent from the output data buffer registers (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0) according to the write timing signal (WTO) generated by the Data Write Control Unit, and the OPTx output waveform is updated. Moreover, the output level can be compulsorily fixed by the DTTI pin input.

Table 24.6-1 Output Data Register (OPDUR and OPDLR)

OPx1,OPx0 Setting	OPTx Output
OPx1,OPx0 = 0,0	Low Level
OPx1,OPx0 = 0,1	16-bit PPG Timer Output
OPx1,OPx0 = 1,0	16-bit PPG Timer Inverted Output
OPx1,OPx0 = 1,1	High Level

The OPTx output waveform timing diagram is shown in Figure 24.6-2 and the operation is explained in following sections.

■ OPTx Output Waveform Timing Diagram (WTS1,WTS0 = 00$_B$)

Figure 24.6-2 OPTx Output Waveform Timing Diagram (WTS1,WTS0 = 00$_B$)
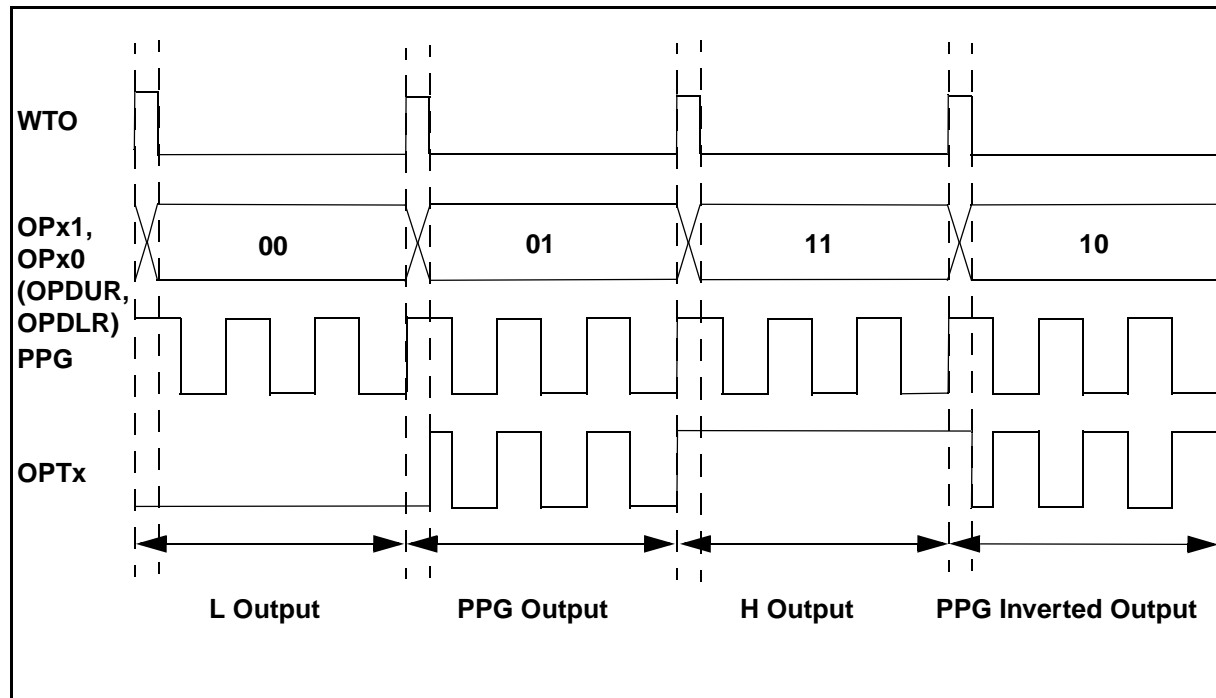
24.6.1   Operation of Position Detection

This section describes the operation of the Position Detection Circuit. When the effective position is detected, a Data Write Timing Output (WTIN1) will be generated to the Data Write Control Unit and a Position Detect Interrupt is generated if the OPCLR:PDIE is set to "1".

■ Operation of Position Detection

The WTIN1 signal is generated by the Position Detection Circuit under the following conditions:

- A comparison match between SNI2 to SNI0 and RDA2 to RDA0, which is triggered by any effective edge of SNI2 to SNI0.

- A detection of effective edge at SNIx which is enabled by the corresponding SEEx bit.

When the CMPE bit in the input control register upper (IPCUR) is set to "0", only the edge detection of SNIx pins enabled by the SEE2 to SEE0 bits will engage in the edge detection operation for the position detection. For instance, when only the SEE0 bit is set to "1", the input edge to the pin SNI0 is in effect, the data write output signal is generated only when an effective edge is detected at the SNI0 pin. See Figure 24.6-3 for the timing diagram of the edge detection when CMPE = 0.

When the CMPE bit in the input control register upper (IPCUR) is set to "1", the SNI2-SNI0 pins will be engaged in the comparison operation with the RDA2 to RDA0 bits. The comparison is triggered by any edge change at the SNI2-SNI0 pins. See Figure 24.6-4 for the timing diagram of the edge detection when CMPE = 1.

■ Edge Detection Timing Diagram (CMPE = 0)

Figure 24.6-3 Edge Detection Timing Diagram (CMPE = 0)
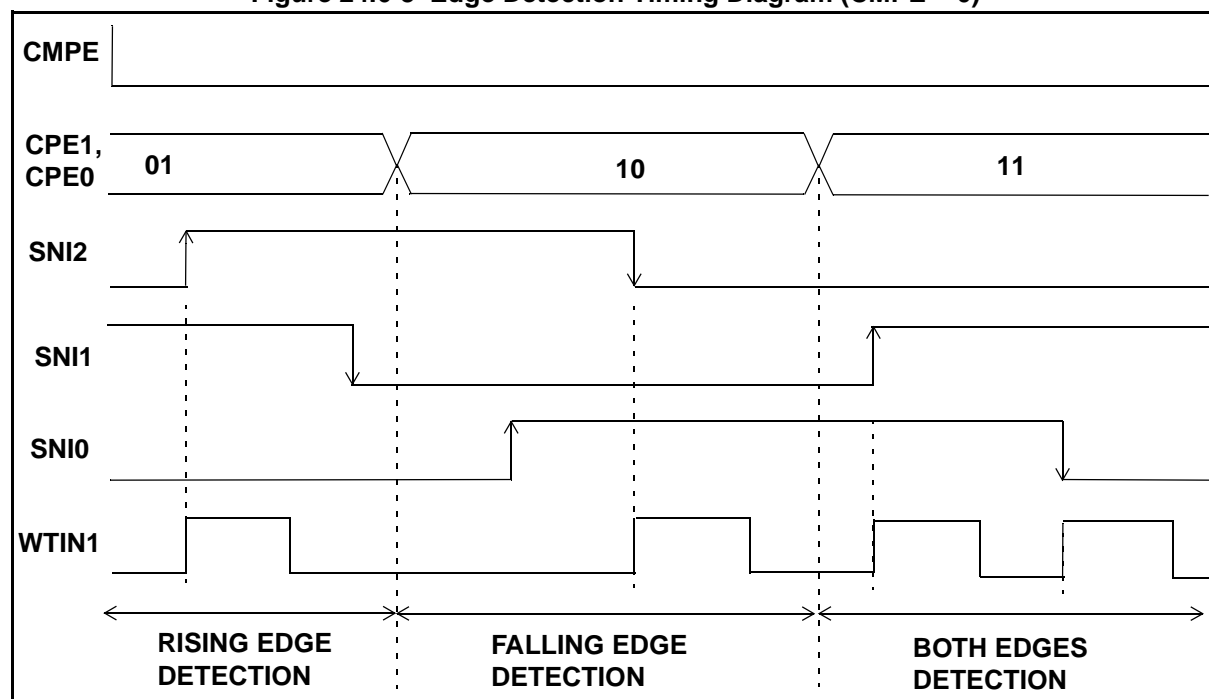
MB95330H Series

■ Both Edges Detection and SNIx/RDAx Comparison Timing Diagram (CMPE = 1)

Figure 24.6-4 Both Edges Detection and SNIx/RDAx Comparison Timing Diagram (CMPE = 1)
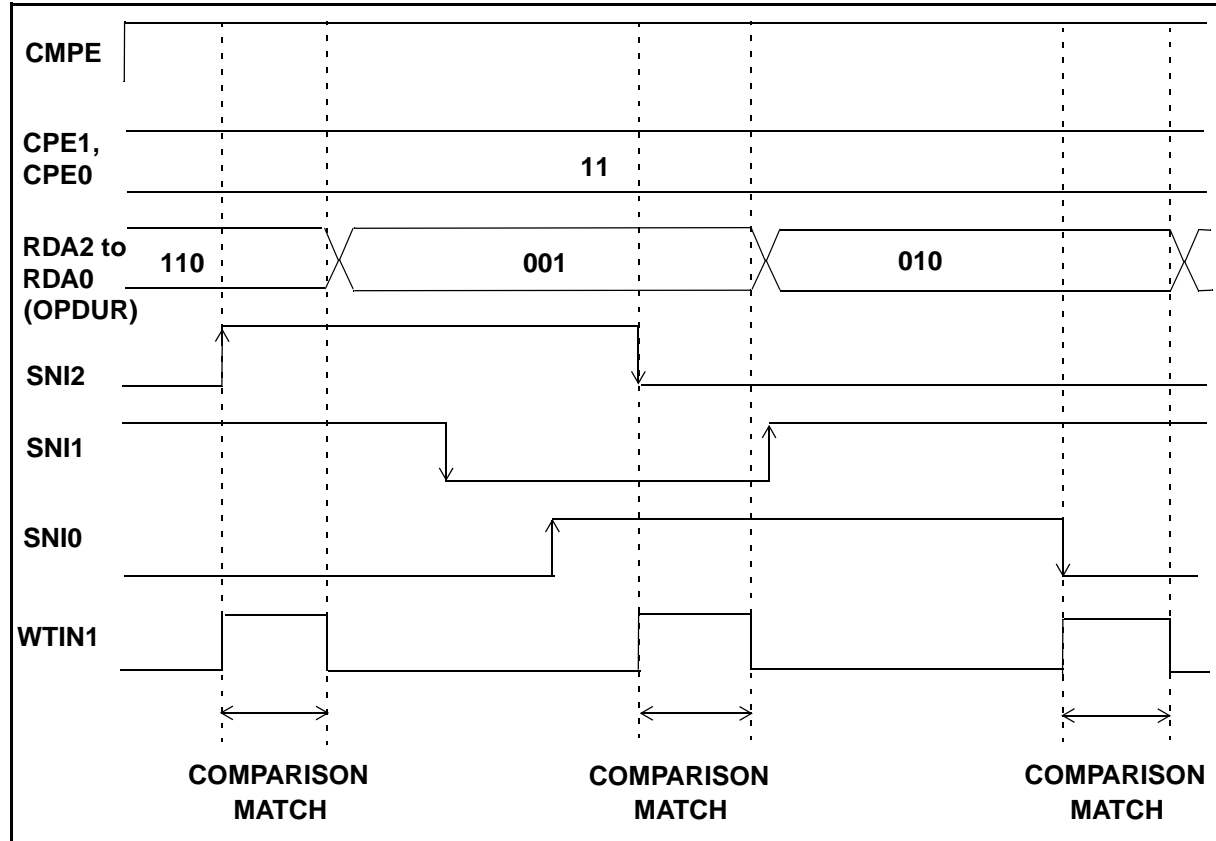
■ WTIN1 Output Condition and Register Setting

Table 24.6-2 WTIN1 Output Condition and Register Setting

CMPE	CPE1	CPE0	SEEx	WTIN1 Output Condition
0	0	0	0	No output. (Initial value)
0	X	X	0	No output.
0	0	0	1	No output.
0	0	1	1	Detect SNIx rising edge.
0	1	0	1	Detect SNIx falling edge.
0	1	1	1	Detect SNIx both edges.
1	0	0	X	Prohibited.
1	0	1	X	Detect SNIx rising edge and SNIx/RDAx comparison match.
1	1	0	X	Detect SNIx falling edge and SNIx/RDAx comparison match.
1	1	1	X	Detect SNIx both edges and SNIx/RDAx comparison match.

Note:
When CMPE = 1, SEEx should be set to "0", setting SEEx = 1 is not recommended.

24.6.2 Operation of Data Write Control Unit

The Data Write Control Unit is used to generate the write timing output (WTO) for transferring data from the output data buffer registers (OPDBRHx,OPDBRLx) to output data register (OPDUR, OPDLR).

■ Operation of Data Write Control Unit

The Write Timing Output (WTO) can be generated by the following condition:

- After values are written to OPDBRH0 and OPDBRL0 by software.

- Triggered by the 16-bit reload timer underflow.

- Triggered by the 16-bit reload timer underflow. The 16-bit timer is started by the position detection comparison circuit.

- Triggered by the position detection input (SNI2 to SNI0) (16-bit reload timer acts as a delay).

- Triggered either by the 16-bit reload timer underflow, or by the position detection input.

At the mean time the cause of generation of WTO will be defined by setting different value of the OPS2 to OPS0 bits in the output control register upper (OPCUR).

■ Signal Flow Diagram for OPDBRH0 and OPDBRL0 by Setting OPS2 to OPS0 = 000$_B$

Figure 24.6-5 Signal Flow Diagram for OPDBRH0 and OPDBRL0 (OPS2 to OPS0 = 000$_B$)
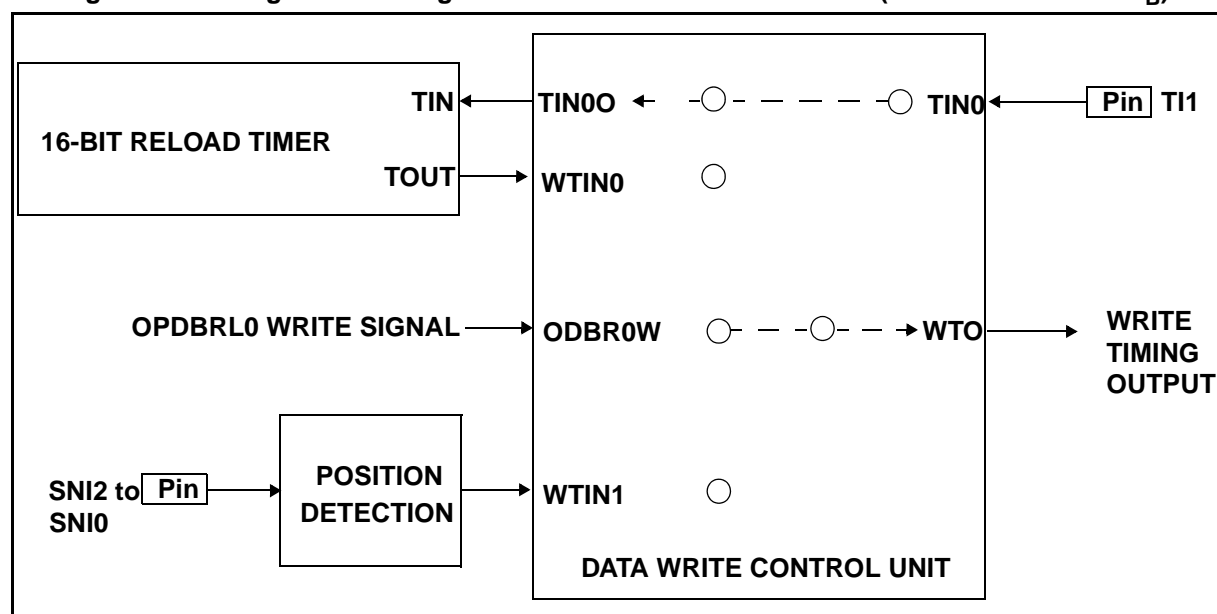
The write timing output signal is generated from the Data Write Control Unit whenever a value is written to OPDBRH0 and OPDBRL0, and the data in OPDBRH0 and OPDBRL0 is transferred to OPDUR and OPDLR one cycle later.

■ **OPDUR and OPDLR Write Timing Diagram (OPS2 to OPS0 = 000$_B$)**

Figure 24.6-6 OPDUR and OPDLR Write Timing Diagram (OPS2 to OPS0 = 000$_B$)
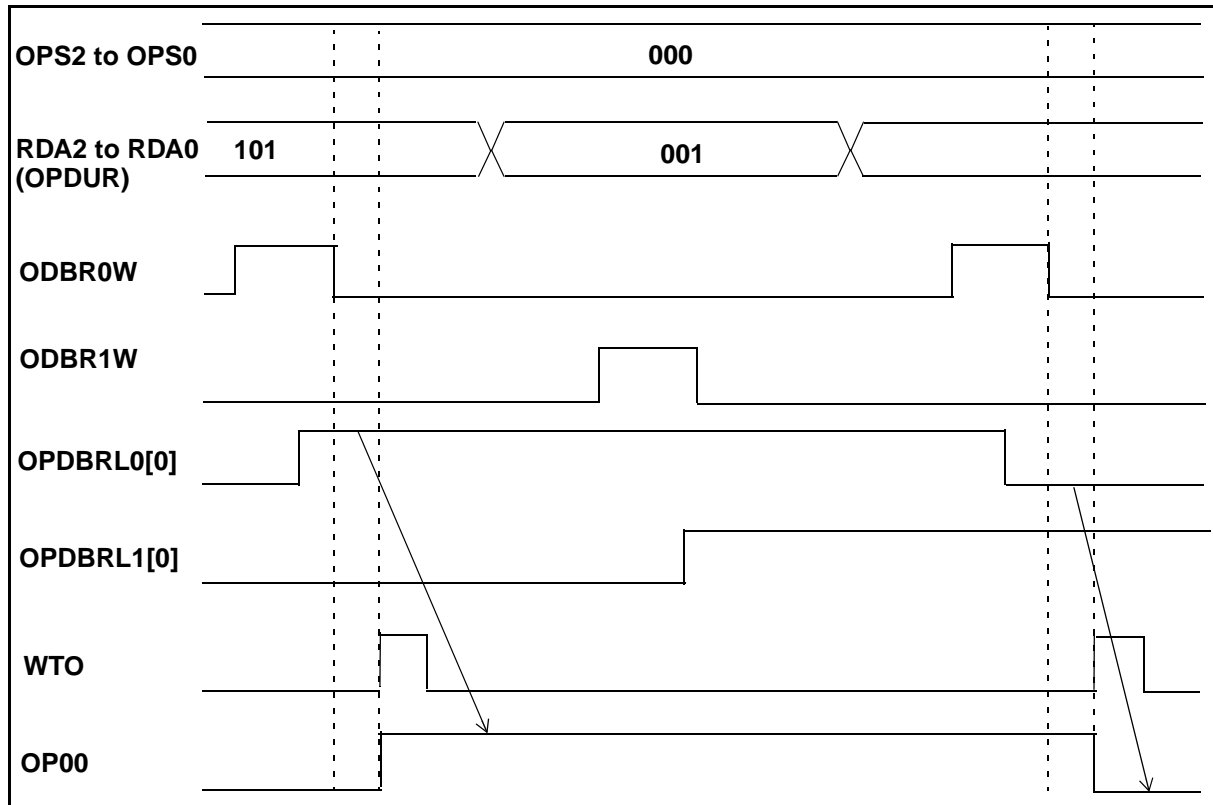
■ **Signal Flow Diagram for Reload Timer Underflow by Setting OPS2 to OPS0 = 001$_B$**

Figure 24.6-7 Signal Flow Diagram for Reload Timer Underflow (OPS2 to OPS0 = 001$_B$)
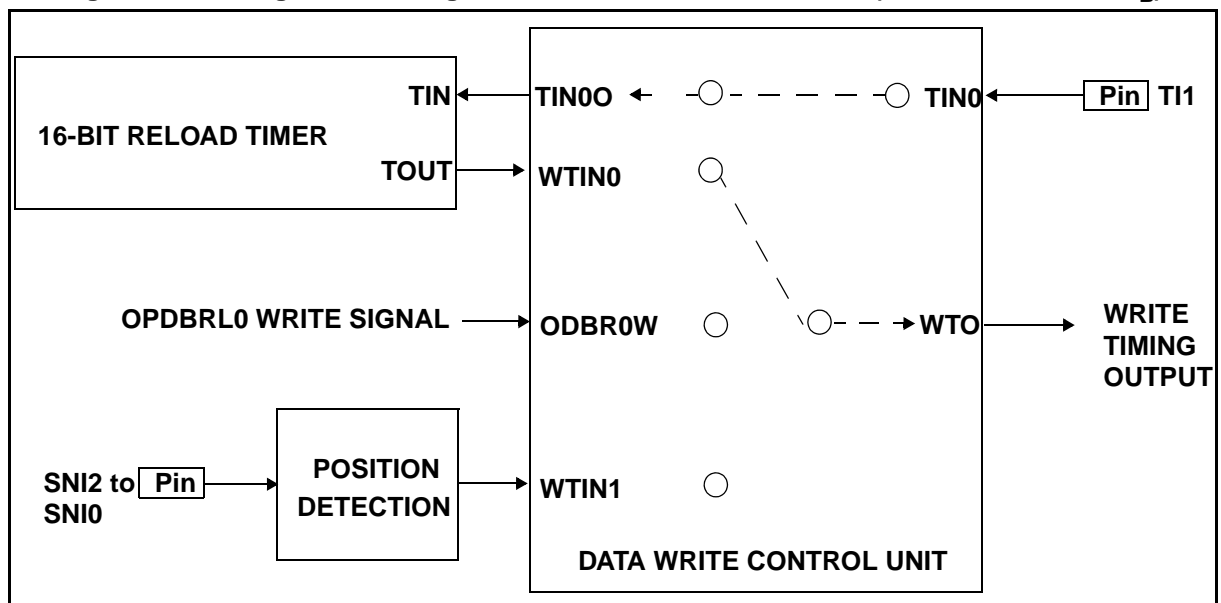
The 16-bit reload timer can be triggered by both TIN input and software to generate the write signal at this setting. The write signal is controlled by the 16-bit reload timer underflow.

■ **Signal Flow Diagram for Position Detection by Setting OPS2 to OPS0 = 010$_B$ or 110$_B$**

Figure 24.6-8 Signal Flow Diagram for Position Detection (OPS2 to OPS0 = 010$_B$ or 110$_B$)
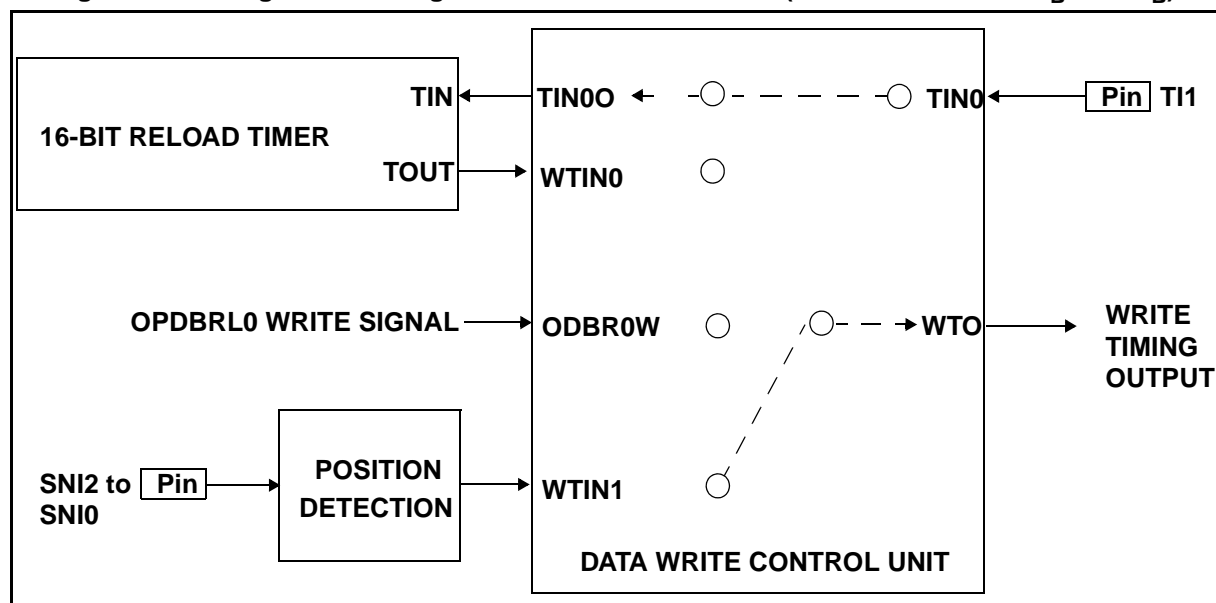
The write signal is generated by a comparison match or effective edge input of position detection.

■ **Signal Flow Diagram for Reload Timer and Position Detection by Setting OPS2 to OPS0 = 011$_B$ or 111$_B$**

Figure 24.6-9 Signal Flow Diagram for Reload Timer & Position Detect (OPS2 to OPS0 = 011$_B$ or 111$_B$)
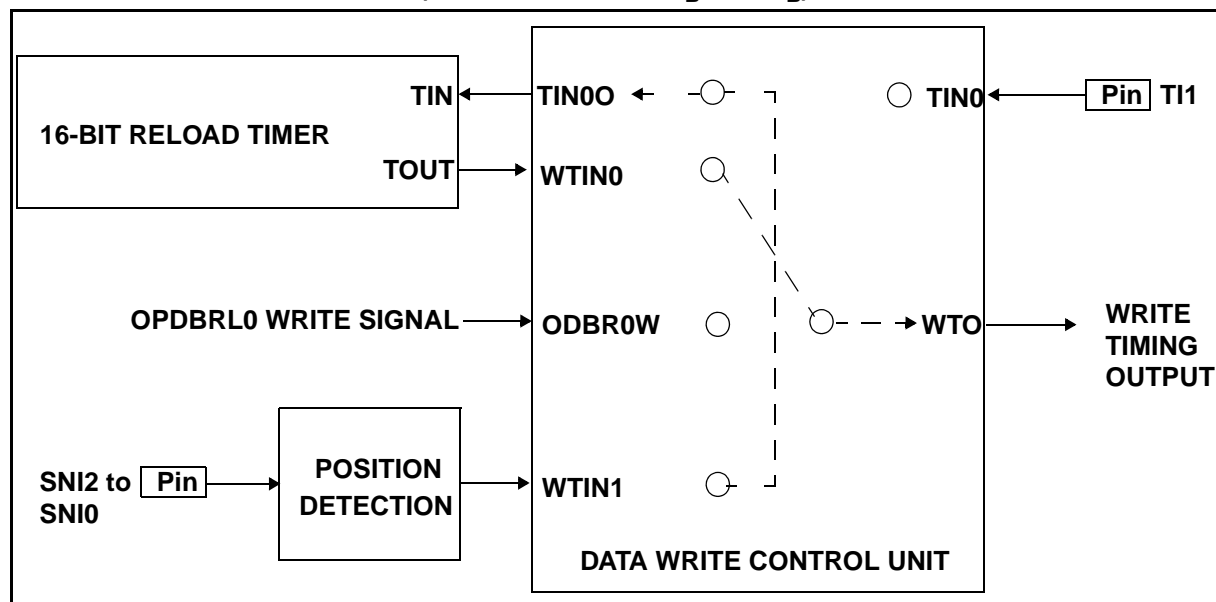
At this setting the16-bit reload timer is started by the compare match or effective edge input of the position detection circuit, write signal is then generated whenever the 16-bit reload timer is underflow. The compare match is triggered by any effective edge change in SNI2 to SNI0 pins.

■ Signal Flow Diagram for Reload Timer or Position Detection by Setting OPS2 to OPS0 = 100$_B$ or 101$_B$

Figure 24.6-10 Signal Flow Diagram for Reload Timer or Position Detect (OPS2 to OPS0 = 100$_B$ or 101$_B$)
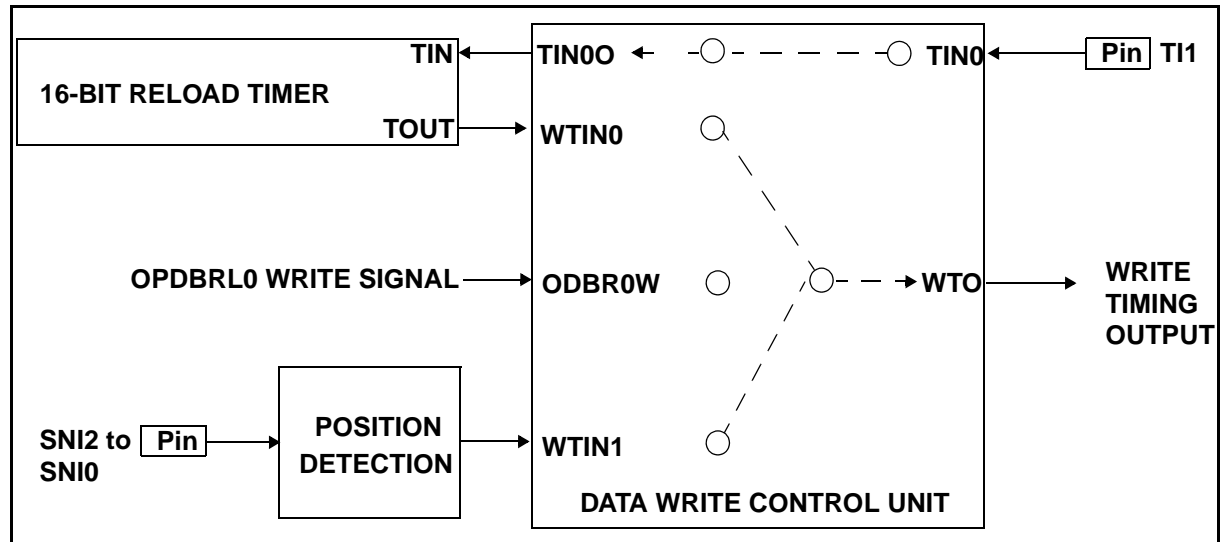
At this setting the write signal is generated by the compare match or effective edge input of the position detection or whenever the 16-bit reload timer is underflow. The compare match is triggered by any effective edge change in SNI2 to SNI0 pins.

■ OPDUR and OPDLR Write Timing Diagram
(OPS2 to OPS0 = 001$_B$, 010$_B$, 011$_B$, 100$_B$, 101$_B$, 110$_B$, 111$_B$)

Figure 24.6-11 OPDUR and OPDLR Write Timing Diagram
(OPS2 to OPS0 = 001$_B$, 010$_B$, 011$_B$, 100$_B$, 101$_B$, 110$_B$, 111$_B$)
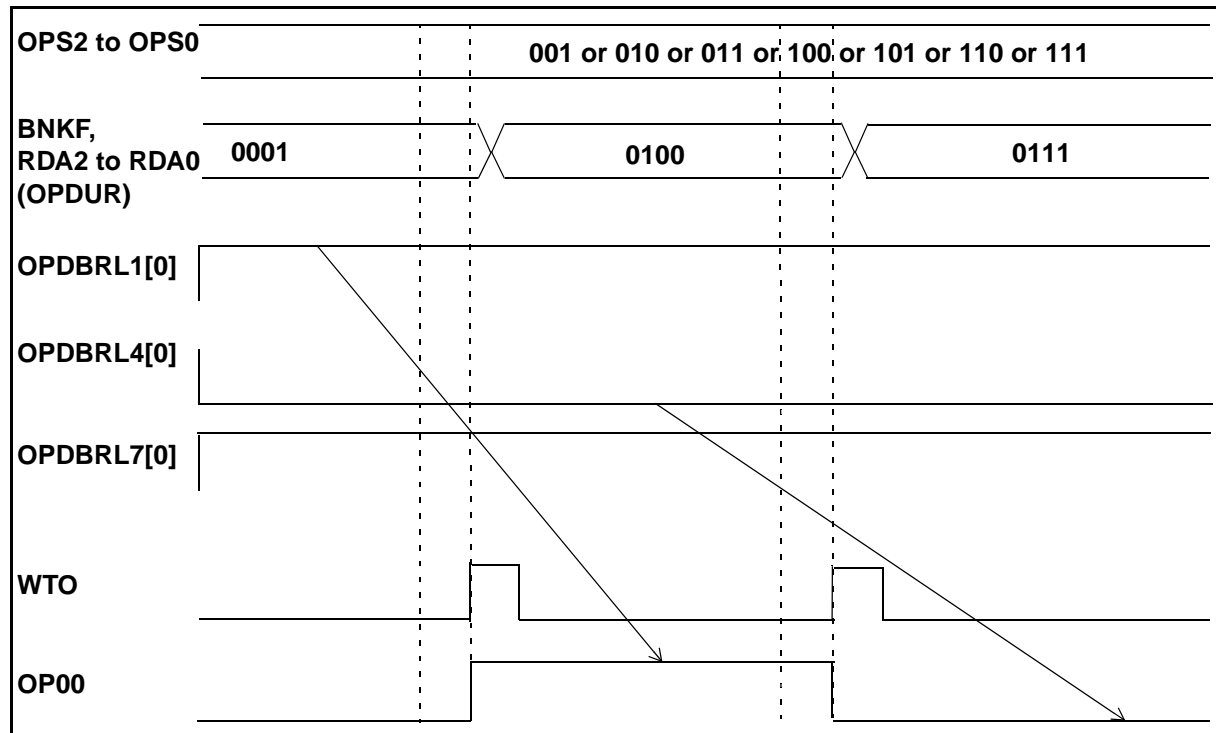
24.6.3   Operation of Output Data Buffer Register

The output data buffer registers (OPDBRH, OPDBRL) are composed of twelve pairs of registers. By loading different OPDBRH and OPDBRL registers into the output data register (OPDUR, OPDLR), various kinds of waveform are output at the multi-pulse generator output (OPT5 to OPT0).

■ Operation of Output Data Buffer Register

The data in the output data buffer registers (OPDBRH, OPDBRL) whose address specified by the BNKF, RDA2 to RDA0 bits is transferred to the output data register (OPDUR, OPDLR) at the write timing generated by the Data Write Control Unit.

The BNKF, RDA2 to RDA0 bits in the output data buffer register upper (OPDBRH) decide the order of data transfer to the output data register (OPDUR, OPDLR), and the OPx1/OPx0 bits decide the shape of the output waveform. The output waveform is updated automatically as long as the write timing (WTO) is generated.

An example of setting the output data buffer registers (OPDBRH, OPDBRL) is shown in Table 24.6-3.

Table 24.6-3 Output Data Buffer Registers (OPDBRH, OPDBRL)

No.	0	1	2	3	4	5	6	7	8	9	A
BNKF	0	0	0	0	0	1	0	X	X	0	1
RDA2	1	1	0	0	1	0	0	X	X	1	0
RDA1	0	0	1	0	1	1	1	X	X	0	1
RDA0	0	1	1	1	0	0	0	X	X	0	1
OP51	0	0	0	1	0	0	0	X	X	0	0
OP50	0	0	1	1	0	0	0	X	X	0	1
OP41	1	0	0	0	0	1	0	X	X	0	0
OP40	1	1	0	0	0	1	0	X	X	1	0
OP31	0	0	0	0	0	0	1	X	X	0	0
OP30	0	0	0	0	1	0	1	X	X	0	0
OP21	0	0	0	0	1	0	0	X	X	0	0
OP20	1	0	0	0	1	1	0	X	X	0	0
OP11	0	0	1	0	0	0	0	X	X	0	1
OP10	0	0	1	0	0	0	1	X	X	0	1
OP01	0	1	0	0	0	0	0	X	X	1	0
OP00	0	1	0	1	0	0	0	X	X	1	0
OPBDR No. Sequence	4	5	3	1	6	A	2	X	X	4	B
OPT5 Output	L	L	PPG	H	L	L	L	X	X	L	PPG
OPT4 Output	H	PPG	L	L	L	H	L	X	X	PPG	L
OPT3 Output	L	L	L	L	PPG	L	H	X	X	L	L
OPT2 Output	PPG	L	L	L	H	PPG	L	X	X	L	L
OPT1 Output	L	L	H	L	L	L	PPG	X	X	L	H
OPT0 Output	L	H	L	PPG	L	L	L	X	X	H	L

Setting the output data buffer register 0 (OPDBRH0, OPDBRL0) (No. 0) as shown in Table 24.6-3 initializes the value of the output data register (OPDUR, OPDLR). The following sequence begins to operate according to the write timing generated:

No. 4 -> No. 6 -> No. 2 -> No. 3 -> No. 1 -> No. 5 -> No. A -> No. B -> No. 9 -> No. 4 and recycle.

The data is transferred to the output data register (OPDUR, OPDLR) sequentially. The output data buffer registers (OPDBRH, OPDBRL) are not used if it is not set, e.g. No. 7 and No. 8 in Table 24.6-3.

24.6.4 Operation of Data Transfer of Output Data Register

Eight methods can be used to transfer data from the output data buffer register (OPDBRHx, OPDBRLx) to the output data register (OPDUR, OPDLR) automatically, which are described in the following section. Each method is selected by setting the OPS2 to OPS0 bits in the output control register upper (OPCUR).

■ Operation of Data Transfer of Output Data Register

There are eight methods of data transfer from output data buffer register (OPDBRHB and OPDBRLB - OPDBRH0 and OPDBRL0) to the output data register (OPDUR, OPDLR):

- Write values to OPDBRH0 and OPDBRL0
- 16-bit Reload Timer Underflow
- Position Detection
- Position Detection and 16-bit Reload Timer Underflow
- Position Detection or 16-bit Reload Timer Underflow
- One-shot Position Detection
- One-shot Position Detection and 16-bit Reload Timer Underflow
- One-shot Position Detection or 16-bit Reload Timer Underflow

The value of the output data buffer register (OPDBRHx, OPDBRLx) that is selected by the BNKF, RDA2 to RDA0 bits in the output data register upper (OPDUR) is transferred to the output data register (OPDUR, OPDLR) when the write signal is generated from the Data Write Control Circuit. However, at the time when OPS2 to OPS0 = 000_B, the value of OPDBRH0 and OPDBRL0 is always transferred to the output data register (OPDUR, OPDLR) in spite of the value of BNKF, RDA2 to RDA0 bits. Figure 24.6-2 shows structure between OPDBRHx, OPDBRLx and OPDUR, OPDLR.

Note:

When the data transfer method is changed, the next Data Buffer Register to be selected is always specified by the BNKF, RDA2 to RDA0 bits in the Data Output Register. This does not apply to the OPDBRH0 and OPDBRL0 Write method. In this Write method, BNKF, RDA2 to RDA0 bits are ignored. To access the output data register, the word access instruction must be used.

Figure 24.6-12 Structure between OPDBRHx, OPDBRLx and OPDUR, OPDLR
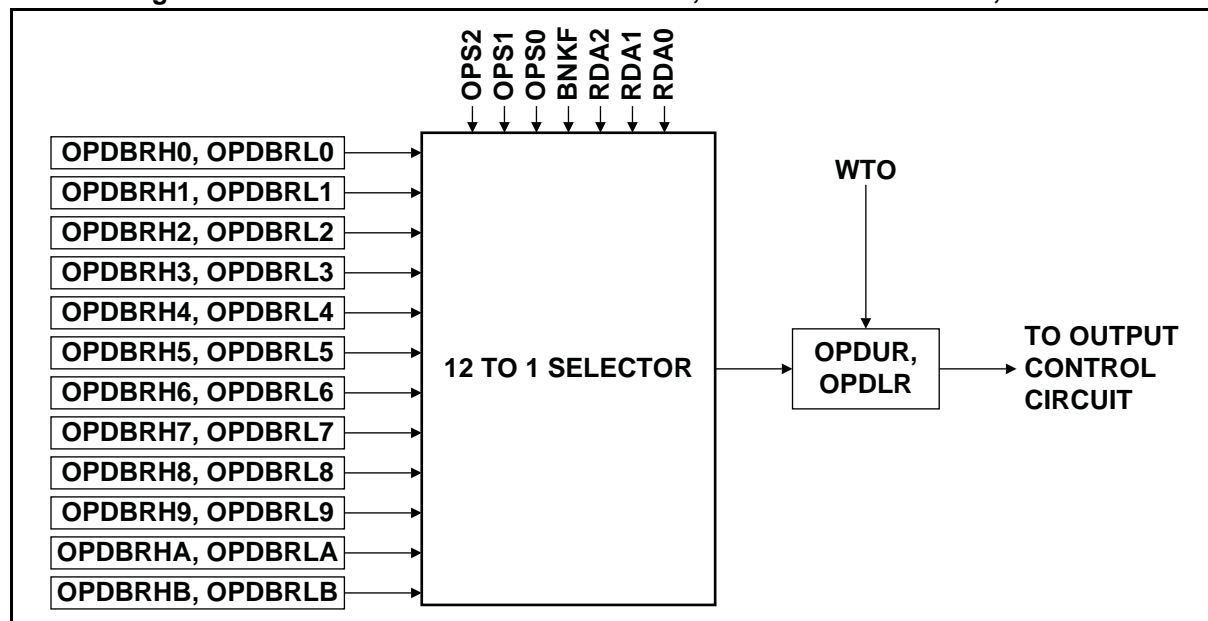
24.6.4.1   At OPDBRH0 and OPDBRL0 Write

The timing change of the output pin OPTx, which is triggered by OPDBRH0 and OPDBRL0 write, is shown in Figure 24.6-13.

■ **Timing Generated by OPDBRH0 and OPDBRL0 Write (OPS2 to OPS0 = 000$_B$)**

Note:

Word access to the output data buffer register 0 must be used in this operation, byte access to either lower register or upper register does not start any transfer operation. The reload timer is free to be used in this operation mode.

Figure 24.6-13 Timing Generated by OPDBRH0 and OPDBRL0 Write (OPS2 to OPS0 = 000$_B$)

MB95330H Series
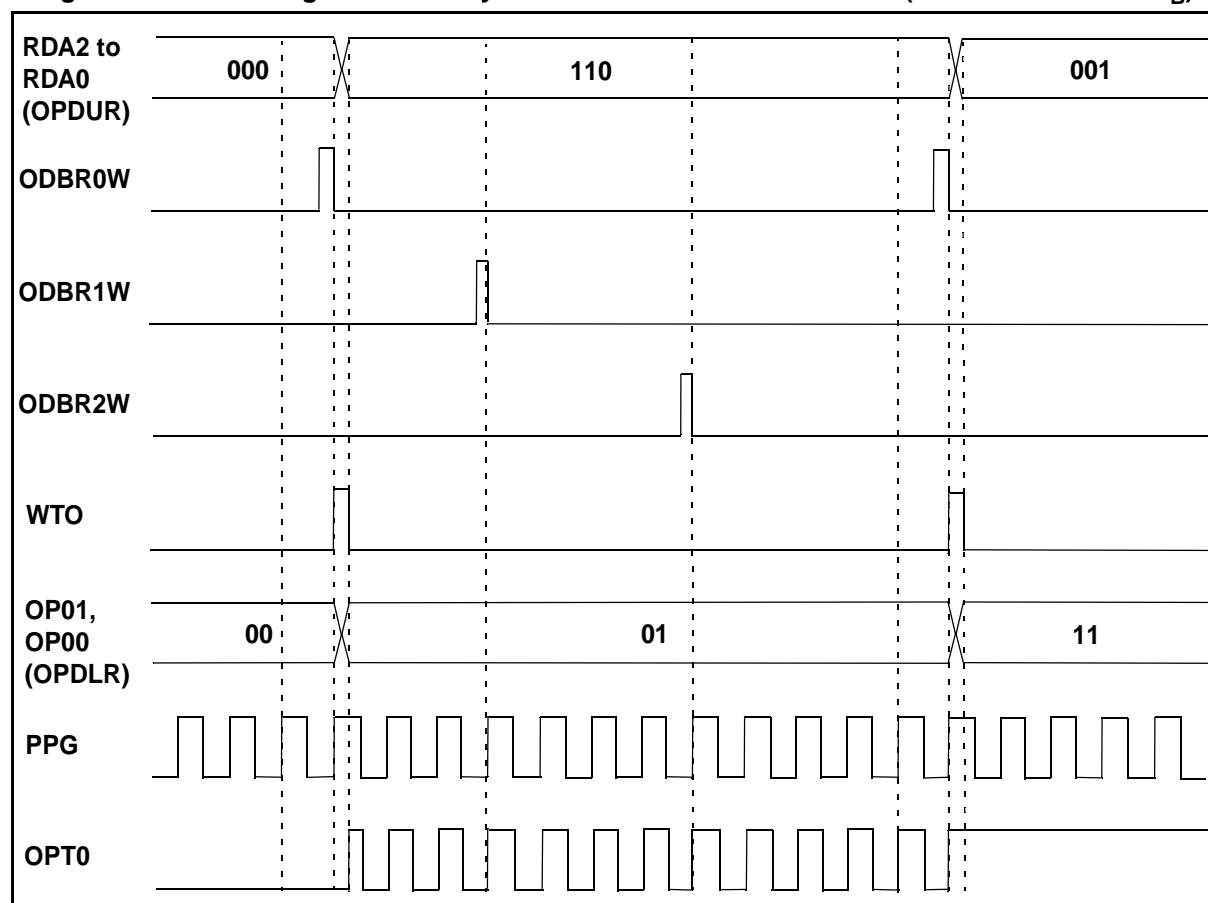
24.6.4.2 At 16-bit Reload Timer Underflow

The timing change of the output pin OPTx, which is triggered by the 16-bit reload timer underflow, is shown in Figure 24.6-14 and Figure 24.6-15.

■ **Timing Generated by Reload Timer Underflow**

Figure 24.6-14 Timing Generated by Reload Timer Underflow
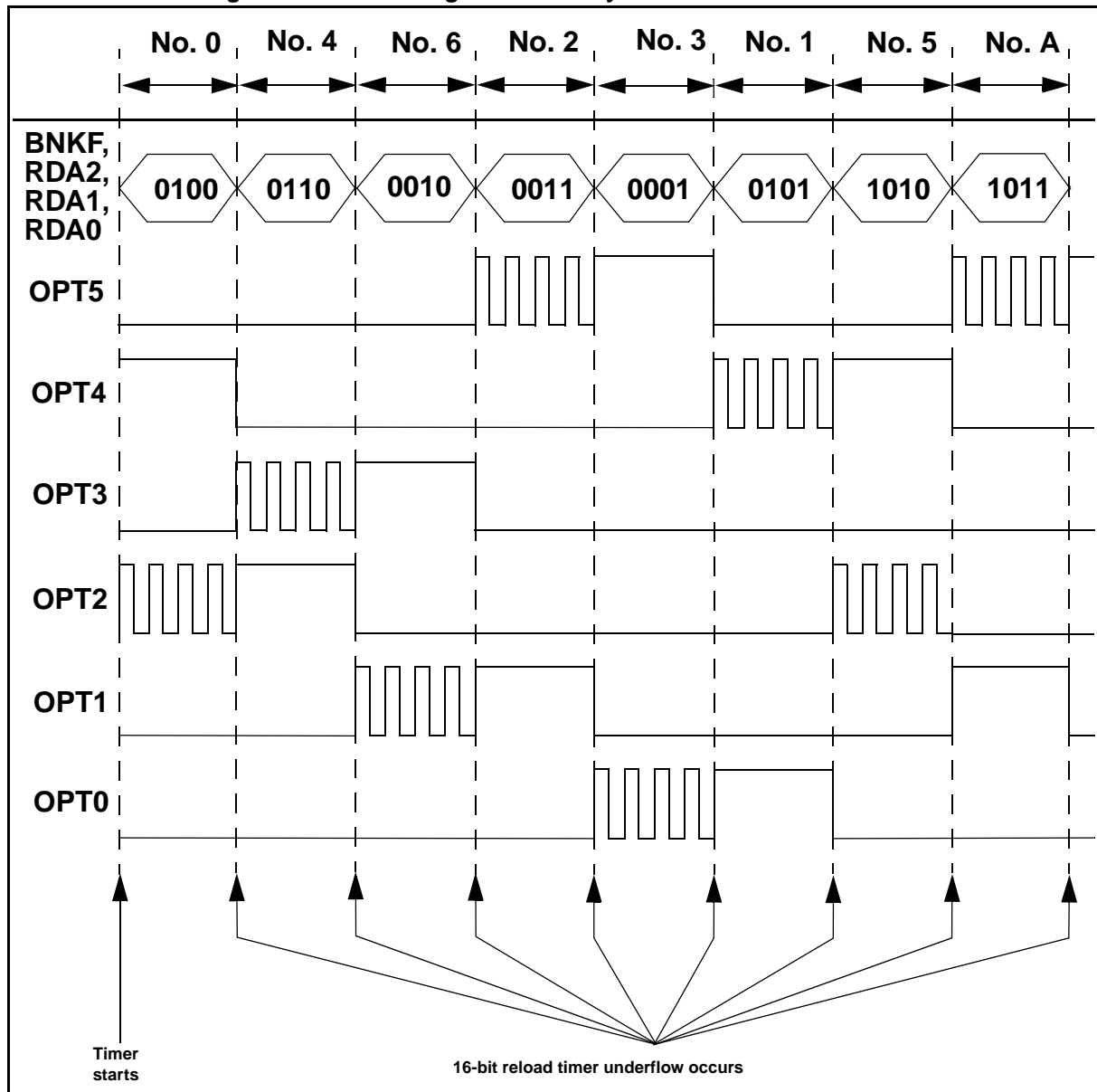
The data transfer from the output data buffer register (OPDBRHx, OPDBRLx) specified by the BNKF, RDA2 to RDA0 bits to the output data register (OPDUR, OPDLR) is updated automatically whenever a 16-bit reload timer underflow is generated as shown in Figure 24.6-15.

In order to use this method, the reload timer should be used in "Reload Mode". Software trigger is needed to be used for the startup of the reload timer. The 16-bit reload timer is needed for setting the update time in advance and executing the continuous control action.

■ **Timing Generated by Reload Timer Underflow (OPS2 to OPS0 = 001$_B$)**

Figure 24.6-15 Timing Generated by Reload Timer Underflow (OPS2 to OPS0 = 001$_B$)
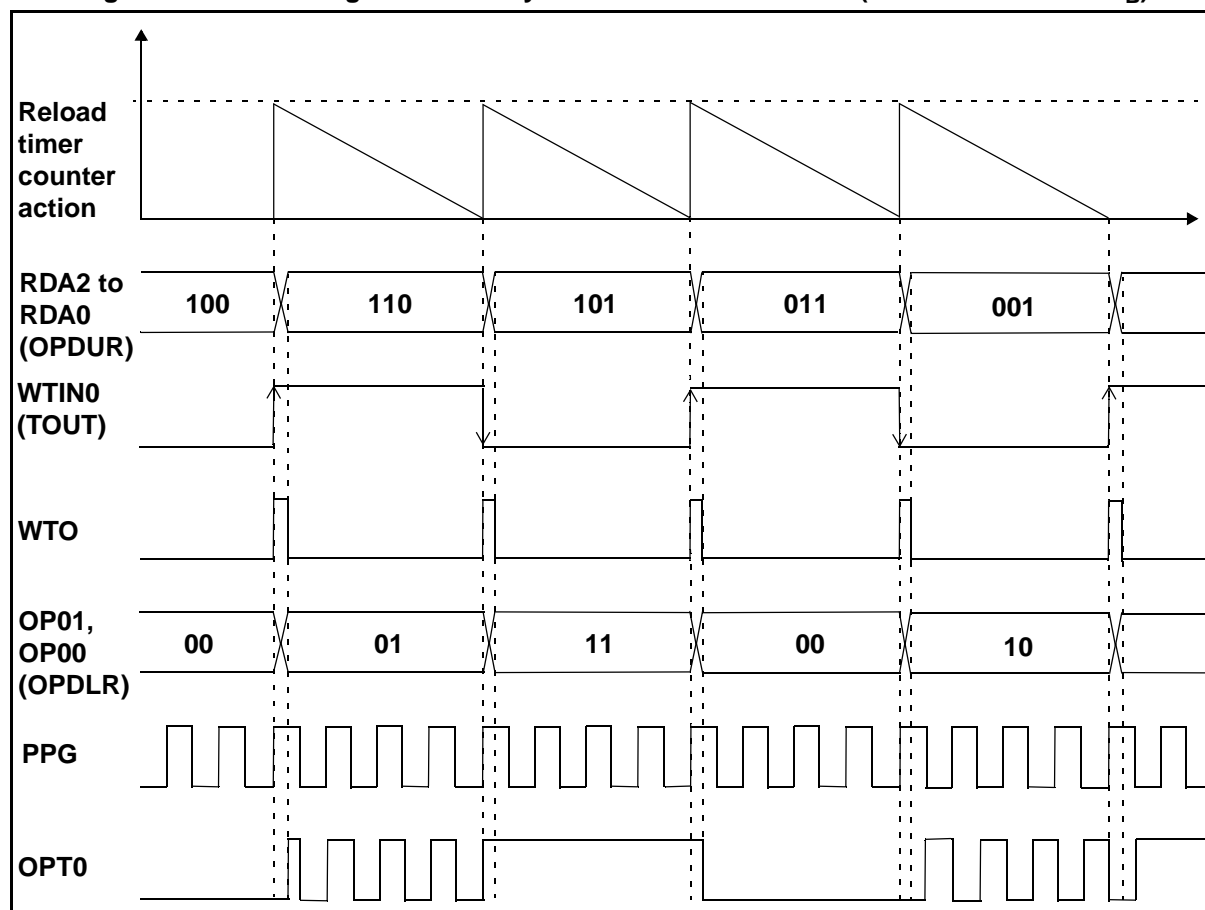
MB95330H Series

24.6.4.3   At Position Detection

The output timing change, which is triggered by the input pin SNIx for the position detection, is shown in Figure 24.6-16 and Figure 24.6-17.

■ **Timing Generated by Position Detection**

Figure 24.6-16 Timing Generated by Position Detection
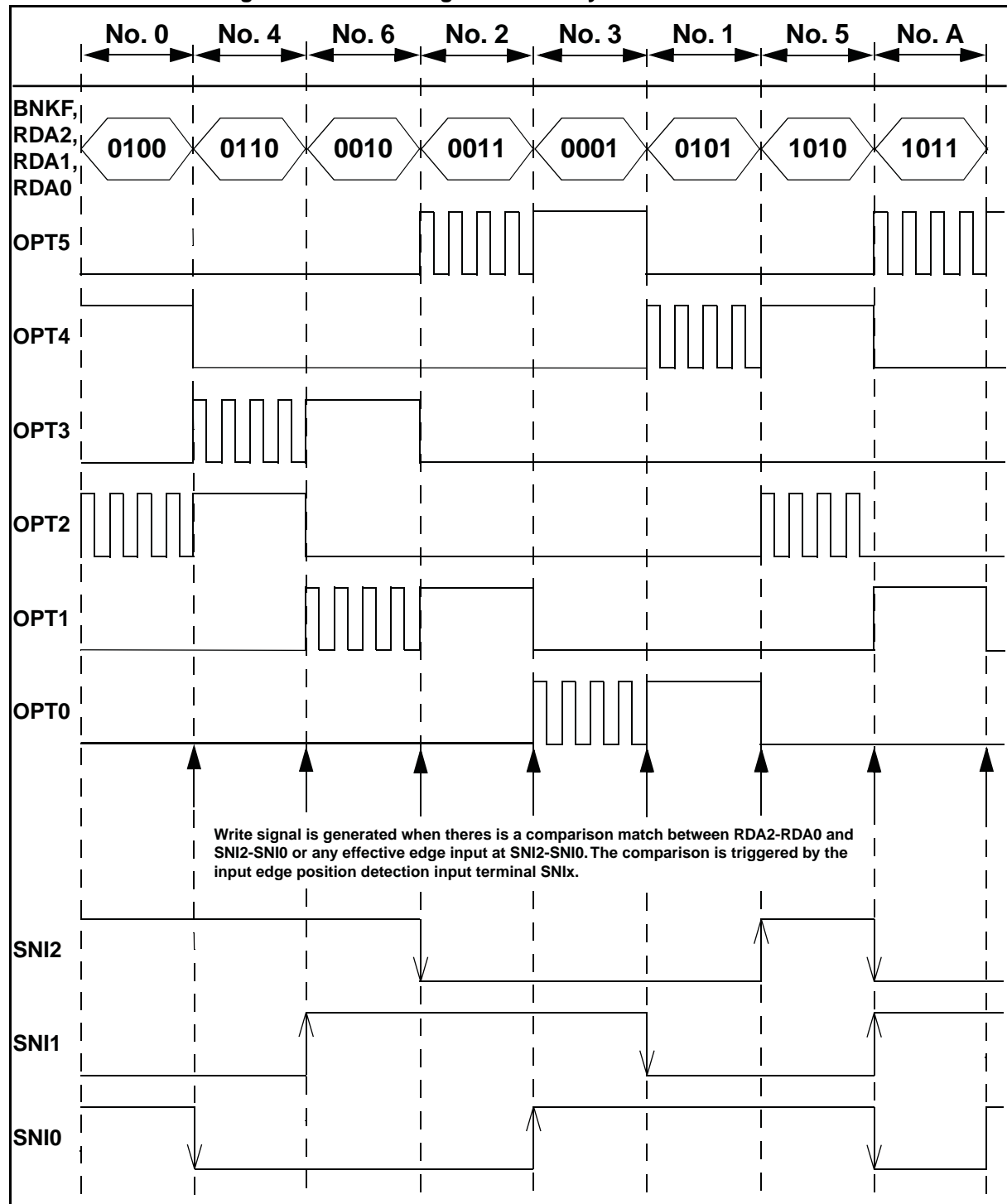
The comparisons between pin SNI2 and RDA2 bit, pin SNI1 and RDA1 bit, pin SNI0 and RDA0 bit are done for each position detection.

The OPTx output waveform is updated according to the effective edge input to pin SNIx as shown in Figure 24.6-17. The data of the output data buffer register (OPDBRHx, OPDBRLx) specified by the BNKF, RDA2 to RDA0 bits is transferred to the output data register (OPDUR, OPDLR), and the output data is renewed automatically when pins SNI2 to SNI0 are compared with the value of the RDA2 to RDA0 bits and matches.

The reload timer can be used in this operation mode.

■ **Timing Generated by Position Detection (OPS2 to OPS0 = 010$_B$)**

Figure 24.6-17 Timing Generated by Position Detection (OPS2 to OPS0 = 010$_B$)

MB95330H Series
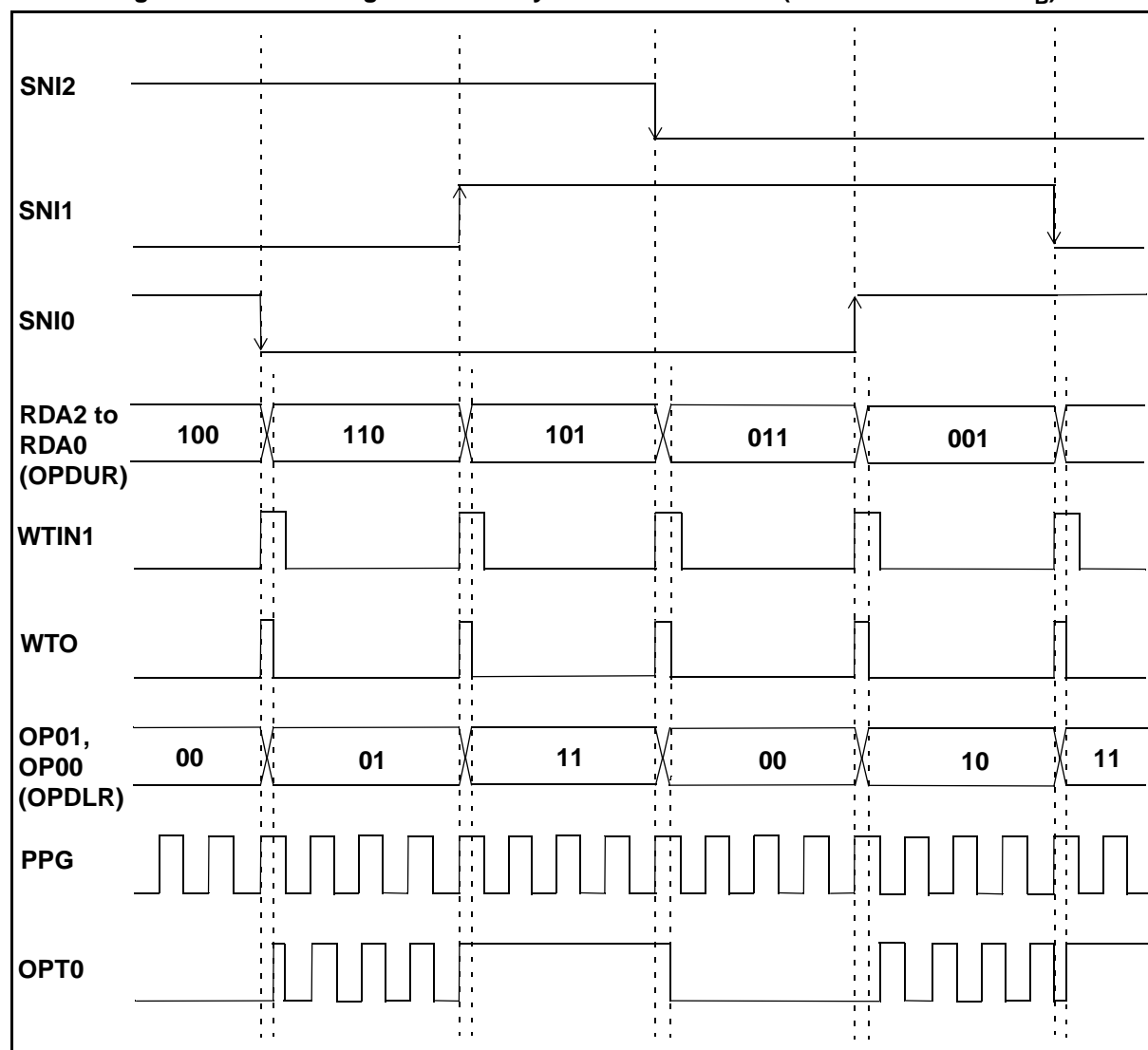
24.6.4.4   At Position Detection and Timer Underflow

The output timing change of the operation of the Position Detection and Reload Timer underflow is shown in Figure 24.6-18 and Figure 24.6-19.

■ **Timing Generated by Position Detection and Timer Underflow**

Figure 24.6-18 Timing Generated by Position Detection and Timer Underflow
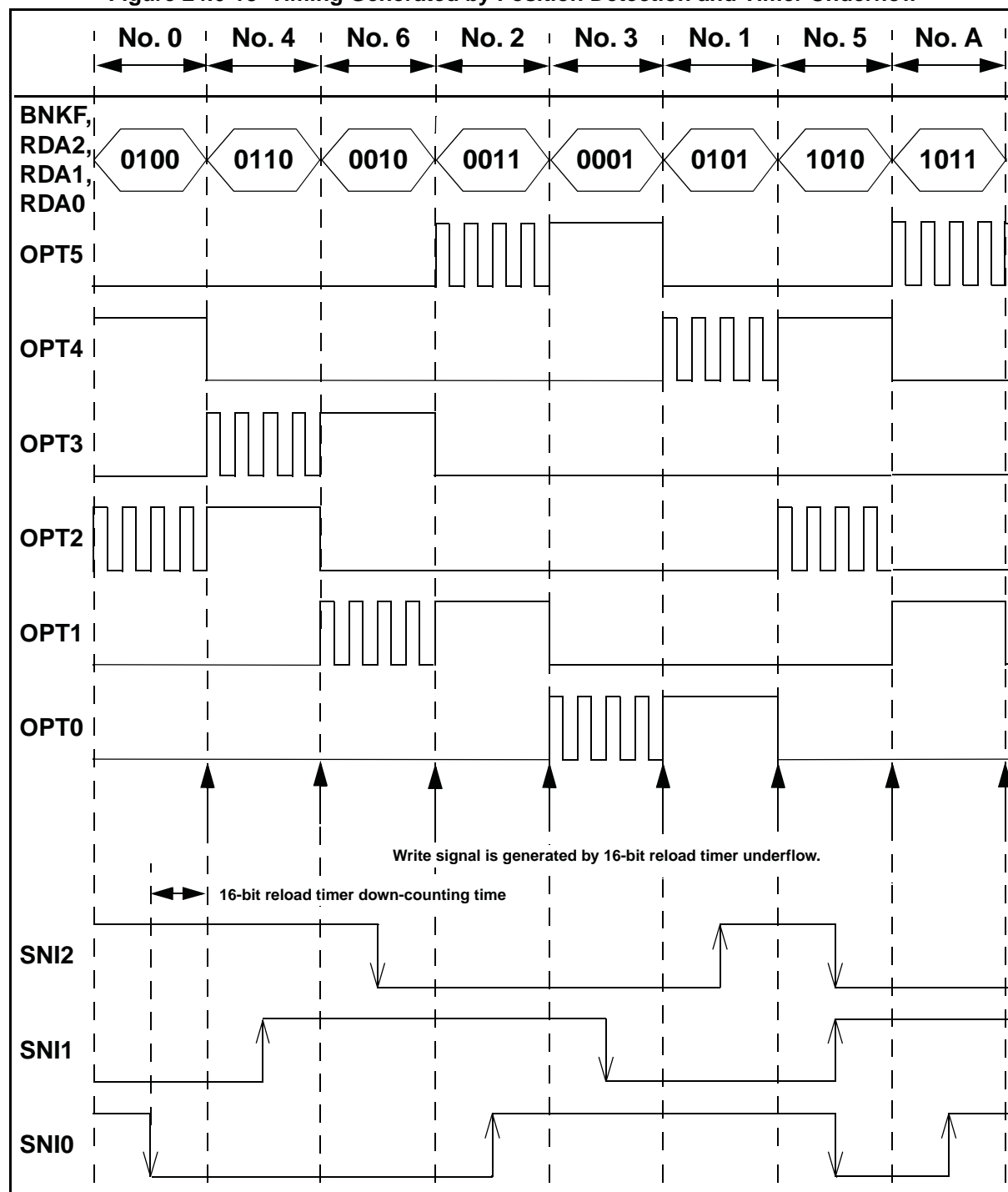
The comparison for the position detection is done in pair for each SNIx pin and RDAx bit (SNI2 and RDA2, SNI1 and RDA1, SNI0 and RDA0), a comparison match starts the 16-bit reload timer. The write signal is generated by the16-bit reload timer underflow.

Pin OPTx output waveform according to the effective edge input to pin SNIx is shown as in Figure 24.6-19. The 16-bit reload timer is started when the pins SNI2 to SNI0 are compared with the value of the RDA2 to RDA0 bits and matches. Data transfer from the output data buffer register (OPDBRHx, OPDBRLx) specified by the RDA2 to RDA0 bits to the output data register (OPDUR, OPDLR) is triggered by the underflow of the 16-bit reload timer. The operation of output data is renewed automatically.

In order to use this method, the reload timer should be used in " Single Shot Mode". TIN0O must be longer than two machine cycles.

MB95330H Series

■ Timing Generated by Position Detection and Timer Underflow
(OPS2 to OPS0 = 011_B)

Figure 24.6-19 Timing Generated by Position Detection and Timer Underflow (OPS2 to OPS0 = 011_B)
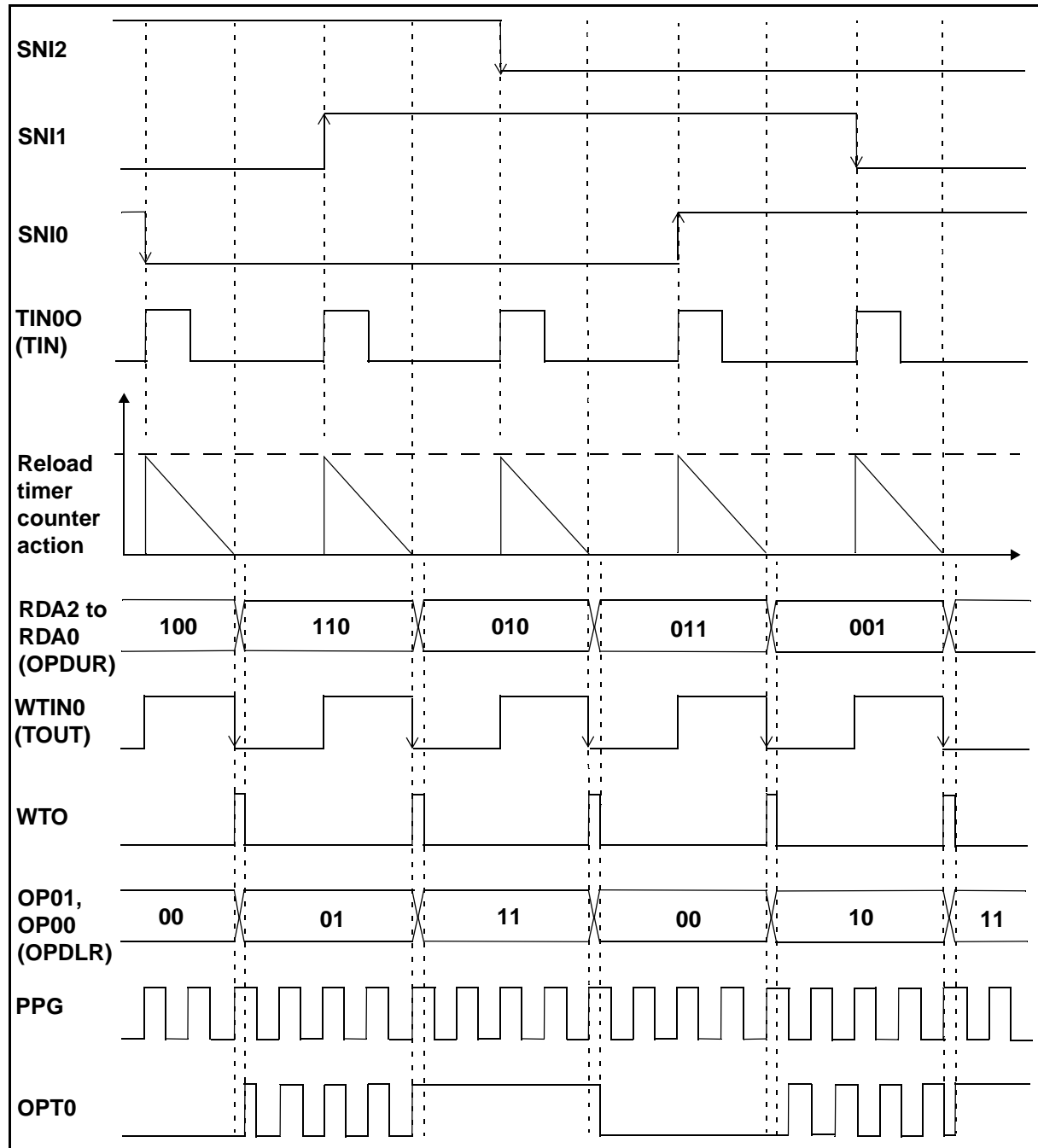
24.6.4.5 At Position Detection or Timer Underflow

The output timing changes of the operation of the Position Detection or Reload Timer underflow are shown in Figure 24.6-20 and Figure 24.6-21. This operation mode is selected by setting the OPS2 to OPS0 = 100_B.

■ **Timing Generated by Position Detection or Timer Underflow**

Figure 24.6-20 Timing Generated by Position Detection or Timer Underflow
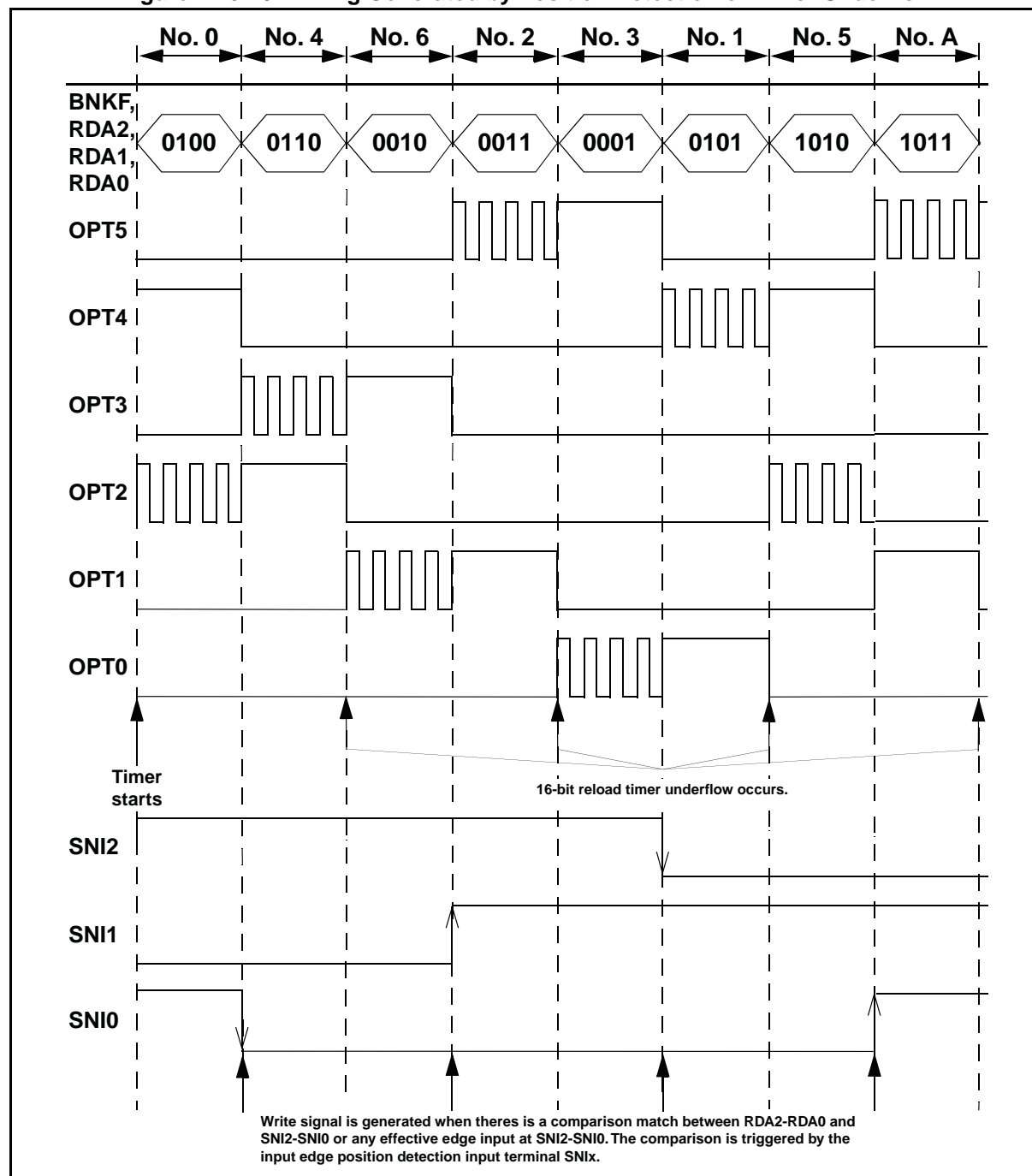
MB95330H Series

■ Timing Generated by Position Detection or Timer Underflow (OPS2 to OPS0 = 100$_B$)

Figure 24.6-21 Timing Generated by Position Detection or Timer Underflow (OPS2 to OPS0 = 100$_B$)
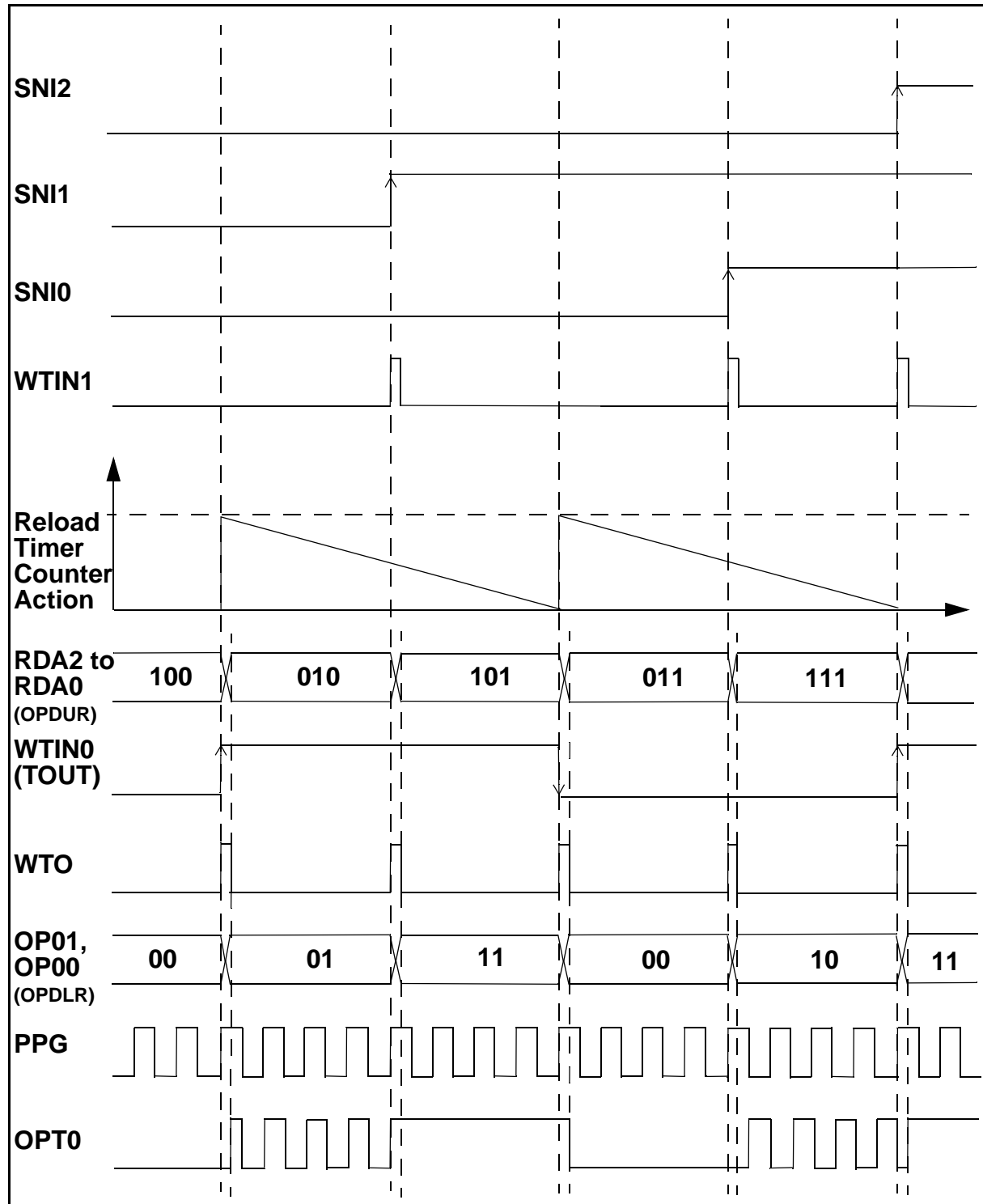
24.6.4.6 At One-shot Position Detection

The output timing change, which is triggered by the input pin SNIx for the one-shot position detection, is shown in Figure 24.6-22.

■ **When One-shot Position Detection**

Same as operation of position detection except that no further position detection will be recognized after the first valid detection until it is changed to ANY OTHER operation mode. The OPTx output waveform is shown in Figure 24.6-22.

The reload timer is free to be used in this operation mode.

■ **Timing Generated by One-shot Position Detection (OPS2 to OPS0 = 110$_B$)**

Figure 24.6-22 Timing Generated by One-shot Position Detection (OPS2 to OPS0 = 110$_B$)

MB95330H Series
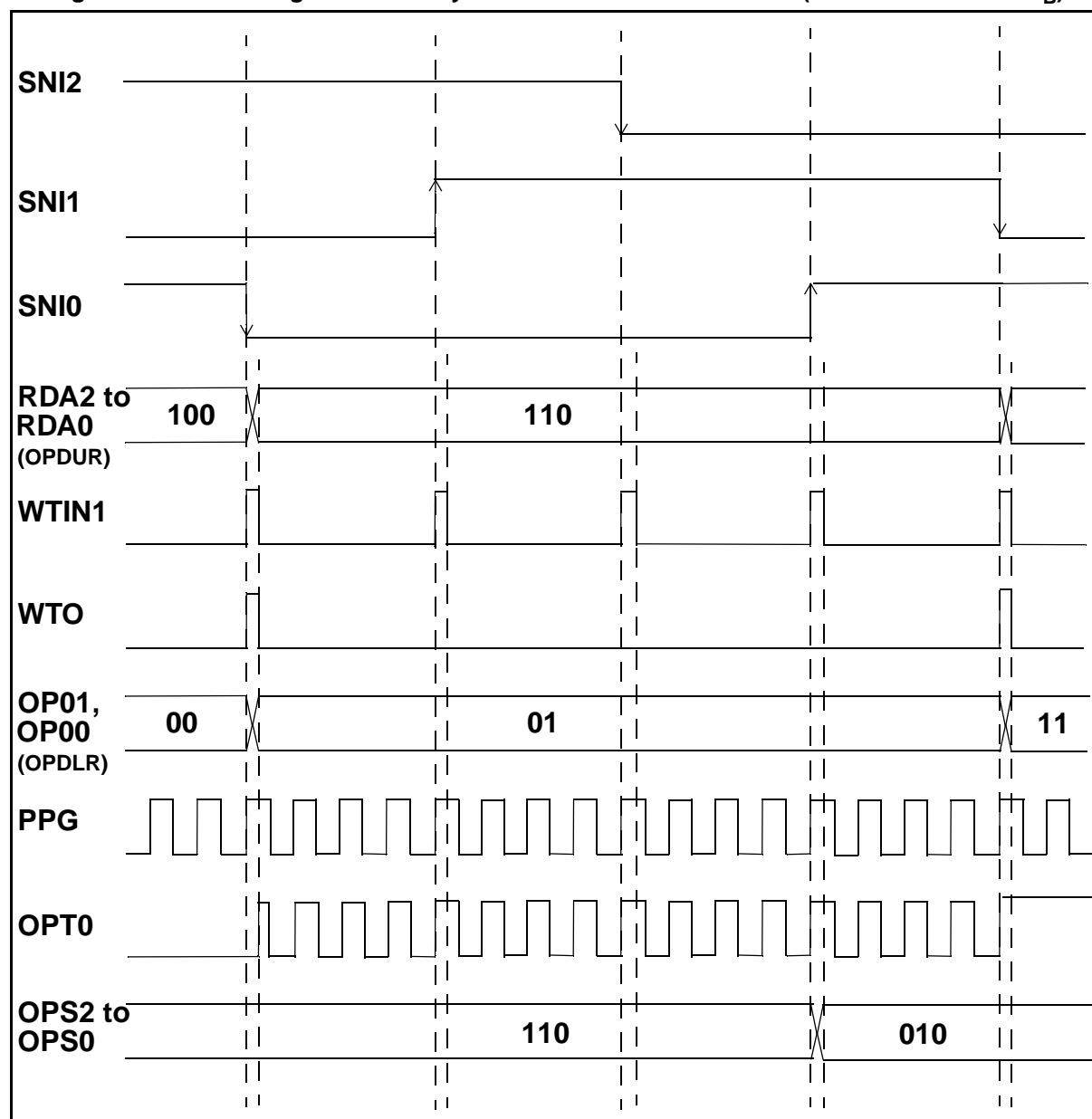
24.6.4.7 When One-shot Position Detection and Timer Underflow

---

The output timing change of the operation of the One-shot Position Detection and Reload Timer underflow is shown in Figure 24.6-23.

■ When One-shot Position Detection and Timer Underflow

Same as operation of position detection and timer underflow except that no further position detection will be recognized after first valid position detection until it is changed to ANY OTHER operation mode. Pin OPTx output waveform is shown as in Figure 24.6-23.

In order to use this method, the reload timer should be used in " Single Shot Mode". TIN0O must be longer than two machine cycles.

■ Timing Generated by One-shot Position Detection and Timer Underflow (OPS2 to OPS0 = 111$_B$)

Figure 24.6-23 Timing Generated by One-shot Position Detection and Timer Underflow (OPS2 to OPS0 = 111$_B$)
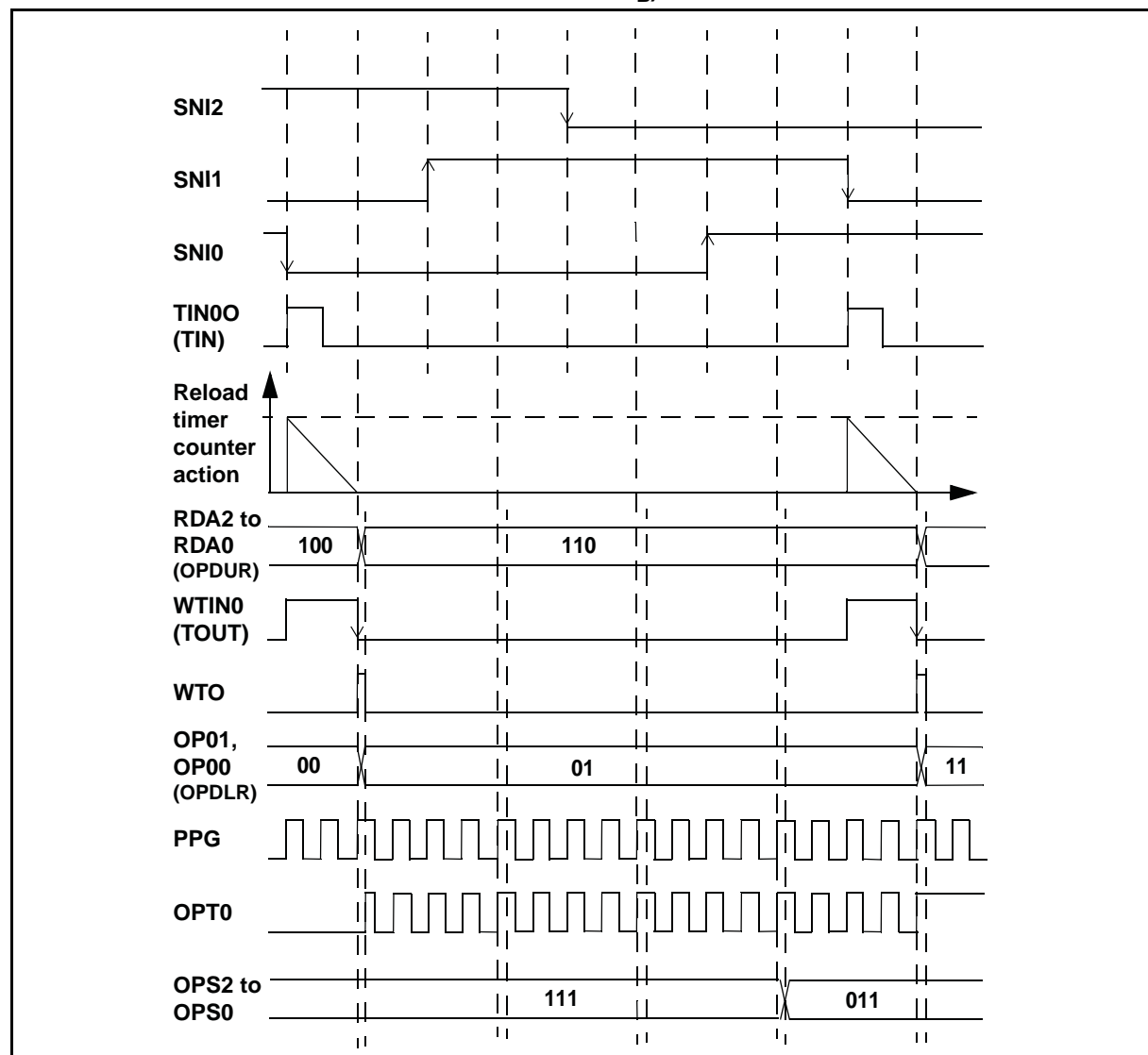
24.6.4.8   When One-shot Position Detection or Timer Underflow

The output timing change of the operation of the One-shot Position Detection or Reload Timer underflow is shown in Figure 24.6-24. This operation mode is selected by setting the OPS2 to OPS0 = 101$_B$.

■ When One-shot Position Detection or Timer Underflow

Same as operation of position detection or timer underflow except that no further position detection will be recognized after first valid position detection until it is changed to ANY OTHER operation mode. Pin OPTx output waveform is shown as in Figure 24.6-24.

■ Timing Generated by One-shot Position Detection or Timer Underflow (OPS2 to OPS0 = 101$_B$)

Figure 24.6-24 Timing Generated by One-shot Position Detection or Timer Underflow (OPS2 to OPS0 = 101$_B$)

MB95330H Series
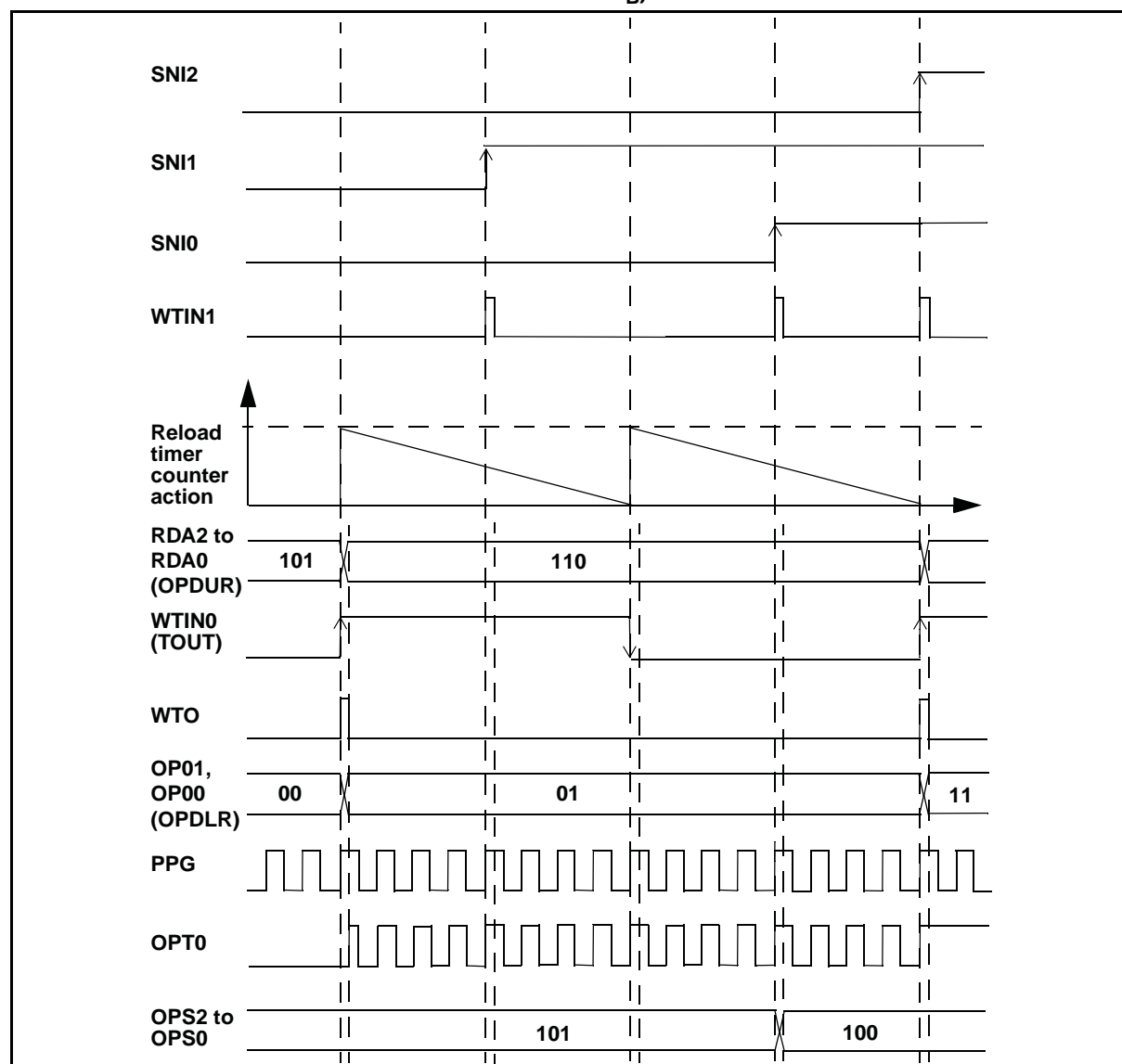
24.6.5 Operation of DTTI Input Control

This section describes the operation of the DTTI Input Control Circuit.

■ Operation of DTTI Input Control

The DTTI circuit controls the output of the value of PDRx (PORTx Data Register) to the pin OPTx which is multiplexed with the PORTx where OPTx is enable by setting OPEx = 1. The operation mode is enabled by the DTIE bit in the output control register upper (OPCUR).

Note:

Before the DTTI circuit is in effect, make sure that the PORTx which is multiplexed with the OPTx is configured as an output port by setting its Data Direction Register.

When the DTIE bit in the output control register upper (OPCUR) is set to "1", the waveform output at OPT5 to OPT0 pins are enabled by the valid level of the DTTI pin. When the low input level is placed at the DTTI pin, the output of OPTx is fixed at the inactive level. The software can set the inactive level for each OPTX pin in PDRx of PORTx, the OPTx pin is then driven by the data written in the PDRx of PORTx.

Even while the output is fixed at the inactive level by the input of the DTTI pin, the timer keeps running, the position detection function does not stop and the data transfer from the output data buffer register (OPDBRHx, OPDBRLx) to the output data register (OPDUR, OPDLR) is continued for waveform generation, but no waveform is output to the OPT5 to OPT0 pins.

Figure 24.6-25 shows the DTTI circuit block diagram and Figure 24.6-26 shows the DTTI circuit timing diagram when D1,D0 is set to "00_B".

■ DTTI Circuit Block Diagram

Figure 24.6-25 DTTI Circuit Block Diagram
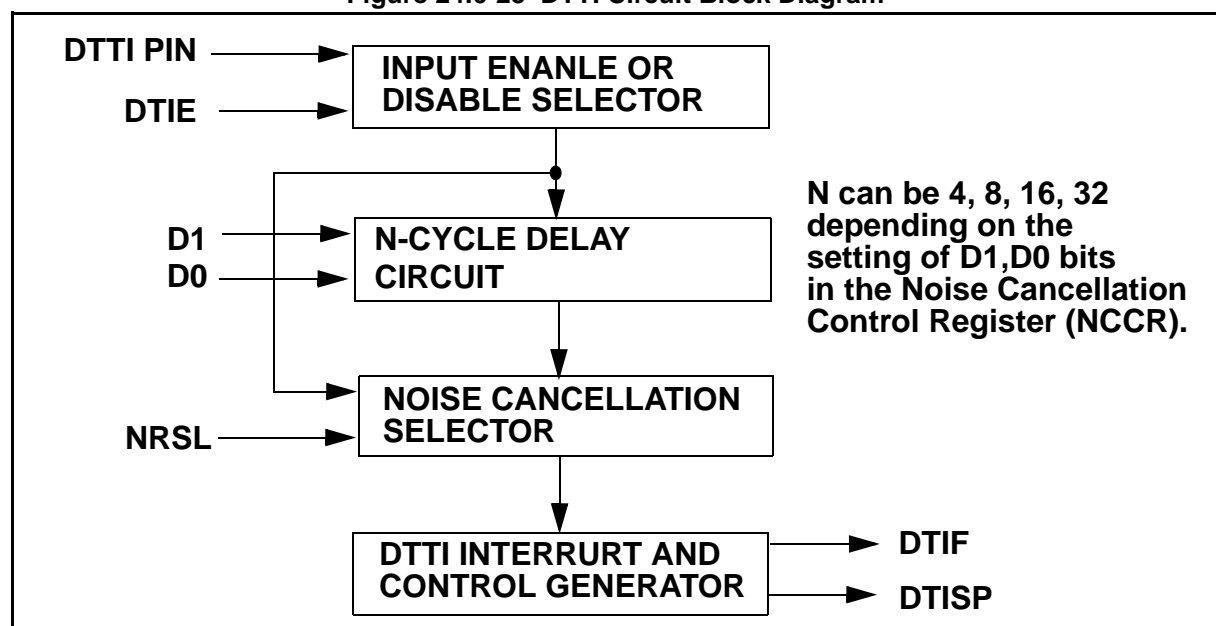
■ **DTTI Circuit Timing Diagram (D1,D0 = 00$_B$)**

Figure 24.6-26 DTTI Circuit Timing Diagram (D1,D0 = 00$_B$)

* DTIF goes to low only by writing a "0" to it.
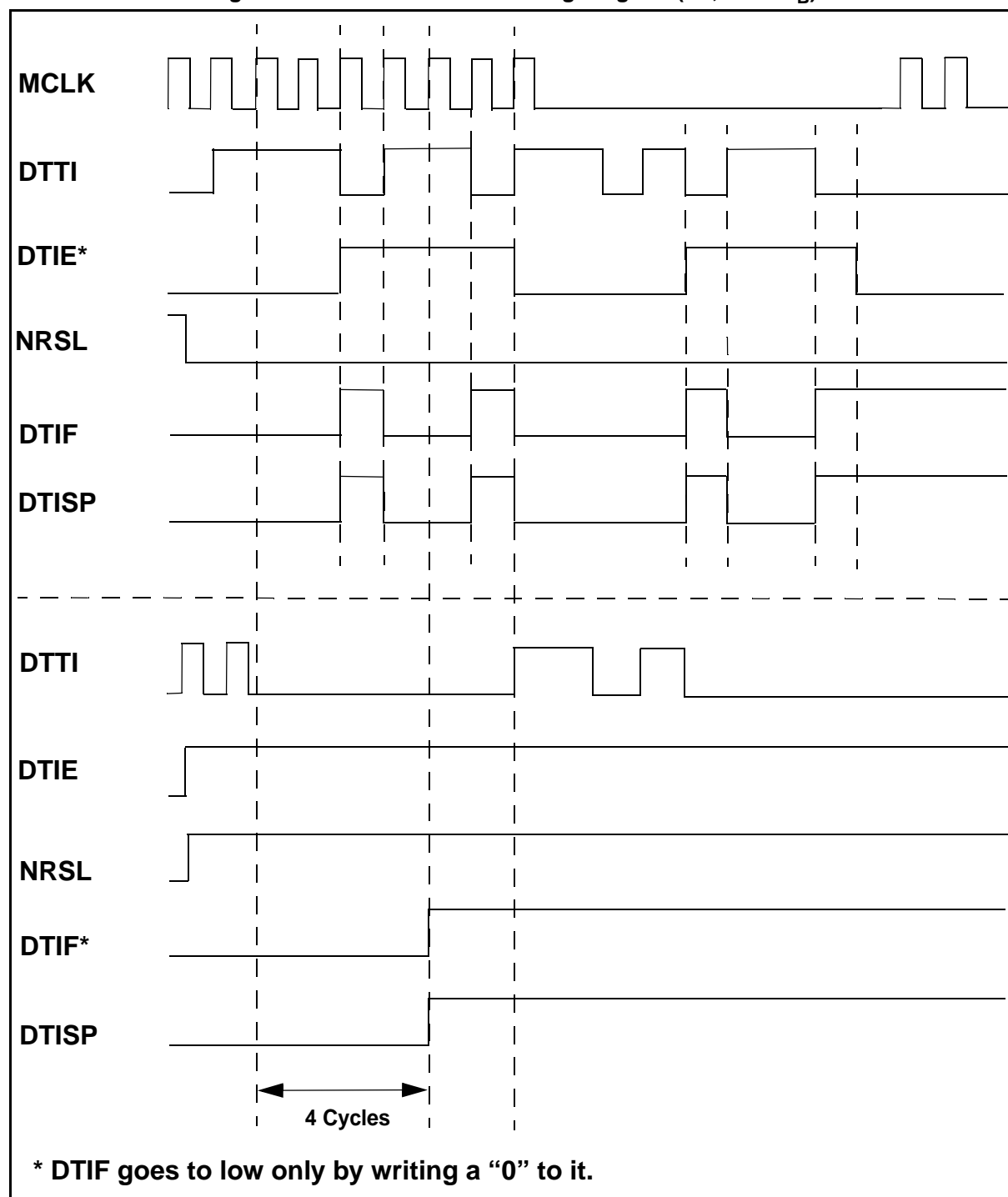
Note:

In the worst case the time from DTTI being recognized (after noise cancellation) to DTISP in effect takes 2 cycles, in best case it takes 1 cycle.

MB95330H Series

■ Relationship between DTTI and OPTx Output

Table 24.6-4 Relationship between DTTI and OPTx Output

NRSL	DTIE	DTTI	Function
X	0	X	DTTI has no effect on OPTx. (Initial value)
0	1	0	DTTI takes effect. Noise filter is not enabled. An "L" input at DTTI pin triggers the output of the inactive level set in PDRx. The DTTI interrupt is generated.
0	1	1	DTTI has no effect on OPTx.
1	1	0	DTTI takes effect. Noise filter is enabled. An "L" input at DTTI pin triggers the output of the inactive level set in PDRx. The DTTI interrupt is generated.
1	1	1	DTTI has no effect on OPTx.

24.6.6   Operation of Noise Cancellation Function

This section describes the noise cancellation function for the SNIx and DTTI pins.

■ **Operation of Noise Cancellation Function**

● DTTI Pin Noise Cancellation Function

When the NRSL bit in the output control register upper (OPCUR) is set to "1", the noise cancellation function for DTTI pin input can be used. When the noise cancellation function is selected, the time for fixing an output pin at the inactive level is delayed for about 4, 8, 16 or 32 machine clocks by the noise cancellation circuit.

Note:

Since the DTTI Input Control Circuit uses a peripheral clock, input is invalidated even if the DTTI input is enabled in a mode such as STOP mode in which the oscillator stops.

● SNI2 to SNI0 Pins Noise Cancellation Function

When SNC2 to SNC0 bits in the input control register lower (IPCLR) are set to "1", the noise cancellation function for SNI2 to SNI0 pins input can be used. When the noise cancellation function is selected, the input is delayed for about four machine clocks by the noise cancellation circuit. Since the noise cancellation circuit uses a peripheral clock, input is invalidated in a mode such as STOP mode in which the oscillator stops even if the SNIx input is enabled.

● Programmable Noise Cancellation Circuit

Noise to be cancelled is programmable to have pulse width less than 4, 8, 16 and 32 machine cycles, i.e. for 16 MHz machine clock, the circuit can filter 0.25 μs to 2 μs width pulses. The control for the programming of the noise cancellation circuit of the SNIx and DTTI pins are separated. Figure 24.4-13 shows the noise cancellation control register.

MB95330H Series

24.6.7 Operation of 16-bit Timer

The 16-bit timer has a buffer and compare clear function, which is used for motor speed checking and abnormal detection timeout. The 16-bit timer starts counting up from counter value "0000$_H$" after a reset has been completed and counting enable bit is set.

■ 16-bit Timer Operation

The counter value is cleared in the following conditions:

- When an overflow has occurred.
- When a match with the compare clear register (CPCUR, CPCLR) is detected.
- When "1" is written to the TCLR bit in the TCSR register during operation.
- When a write timing signal is generated and MODE bit in TCSR is "0".
- When a position detection signal is generated and MODE bit in TCSR is "1".
- Reset

An interrupt can be generated when the counter is cleared due to a match with the compare clear register. There is no interrupt when an overflow occurs.

Note:

To access the compare clear register and the timer buffer register, the word access instruction must be used.

Figure 24.6-27 Clearing the Counter by an Overflow
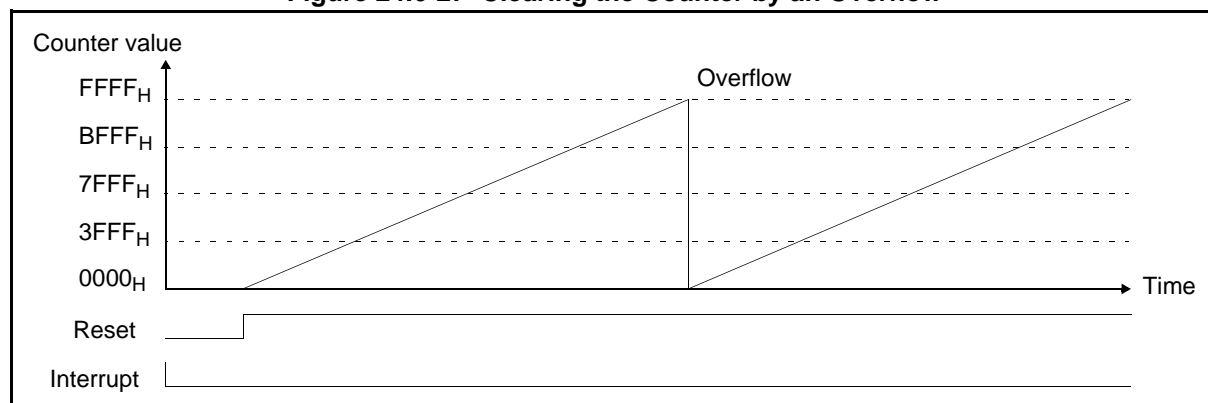
Figure 24.6-28 Clearing the Counter upon a Match with Compare Clear Register

MB95330H Series
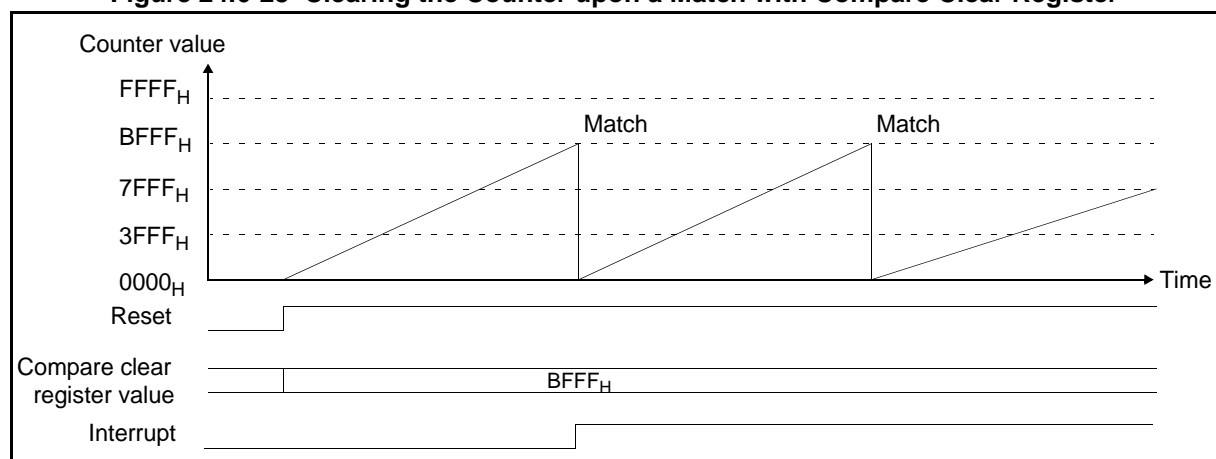
■ 16-bit Timer Timing

The 16-bit timer increases its value at timing according to the prescaler clock and counts up at a rising edge.

Note:

Before the prescaler clock is changed, the Timer Counter should be disabled first by setting the TMEN bit to "0".

Figure 24.6-29 16-bit Timer Count Timing
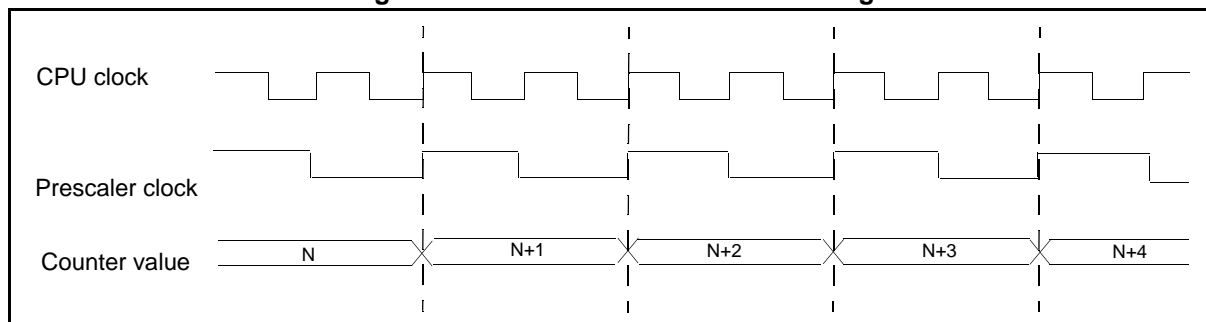
The counter can be cleared upon a reset, software clear (TCLR), a match with the compare clear register, the Write Timing signal or the Position Detection signal. By a reset, the counter is immediately cleared. By a match with the compare clear register, software clear (TCLR), the Write Timing signal or the Position Detection signal, the counter is cleared in synchronization with the count timing.

Figure 24.6-30 16-bit Timer Clear Timing
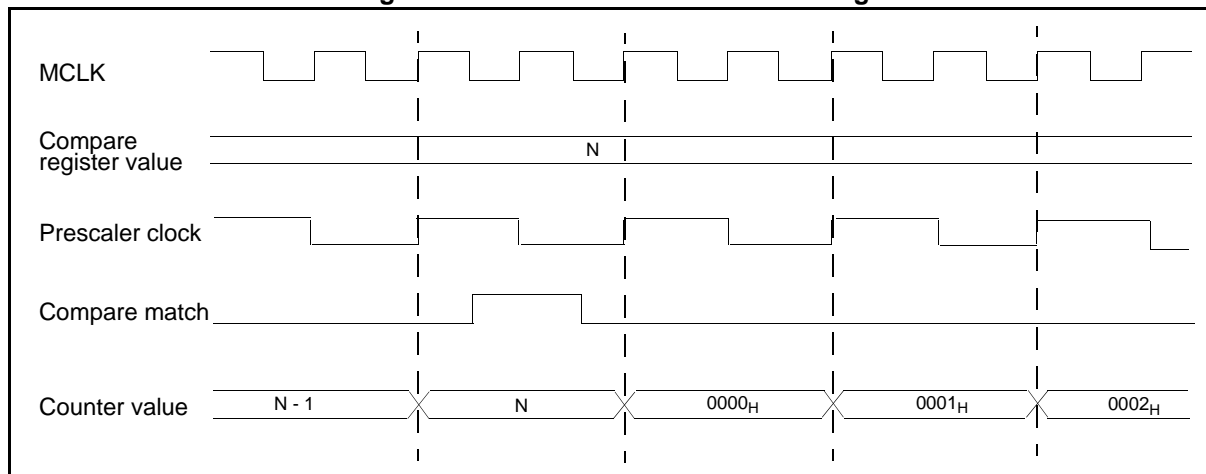
■ **16-bit Timer Buffer Operation Timing Diagram**

Figure 24.6-31 16-bit Timer Buffer Operation Timing Diagram

MB95330H Series
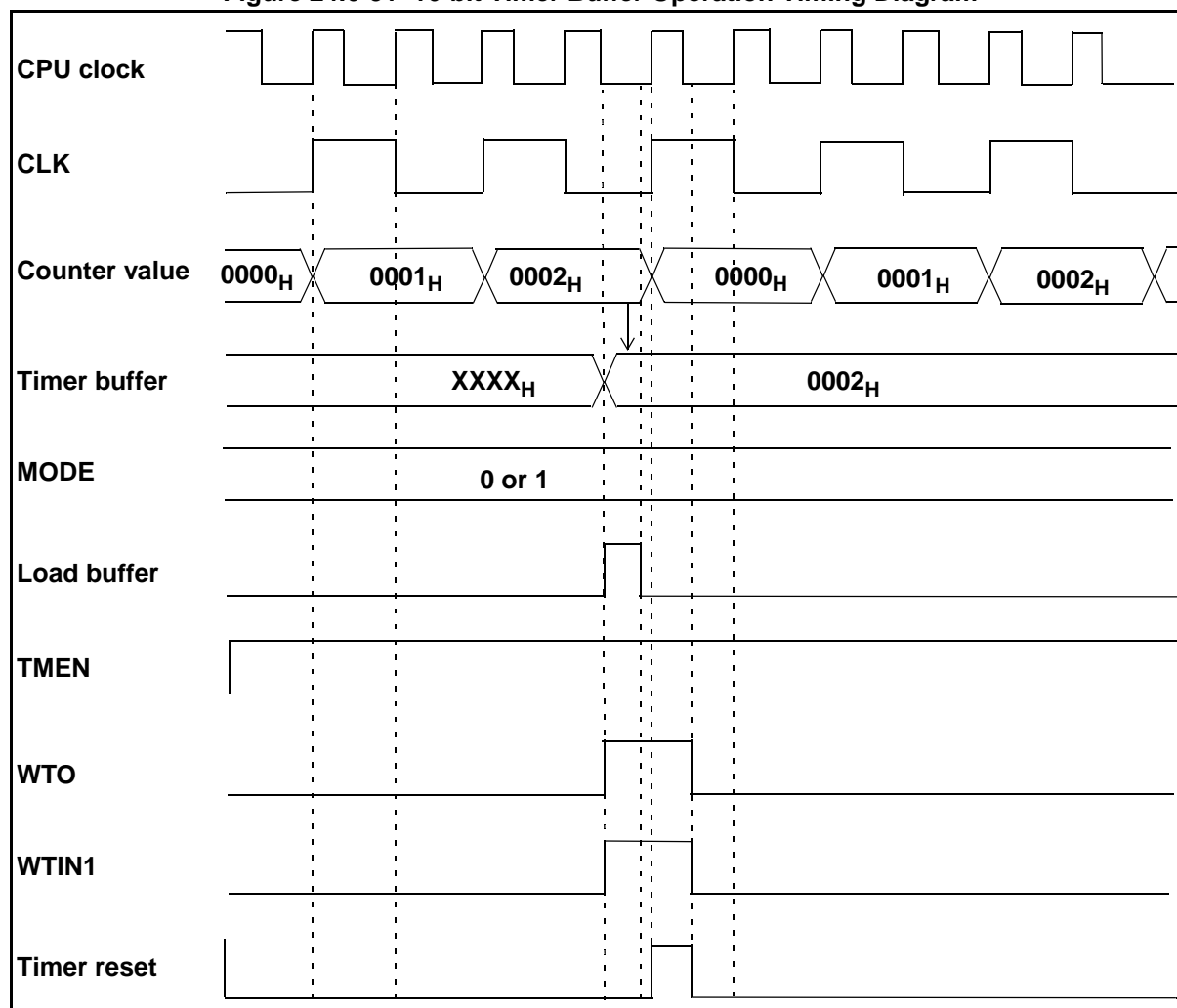
■ **Using 16-bit Timer of Multi-pulse Generator**

The timer is reset when write timing or position detection interrupt flag is set, which is selectable by the MODE bit in the timer control status register (TCSR).

The timer can be started or stopped by setting the TMEN bit in the timer control status register (TCSR). There is no timer overflow interrupt. Whenever the timer is restarted, the current counter value is latched to a buffer for speed calculation.

If the counter value matches the compare clear register (CPCUR, CPCLR), it interrupts the CPU and the timer is reset.

Note:

If the values loaded to the compare clear register upper (CPCUR) and the compare clear register lower (CPCLR) are the same as the timer counter value, the comparison operation will NOT be performed until the next occasion in which the values of CPCUR and CPCLR are the same as the timer counter value.

The Compare Clear interrupt shares the same interrupt vector with the Write Timing interrupt while Compare Match interrupt shares the same vector as that of the Position Detect interrupt.

■ **16-bit Timer in Multi-pulse Generator Operation Diagram**

Figure 24.6-32 16-bit Timer in Multi-pulse Generator Operation Diagram
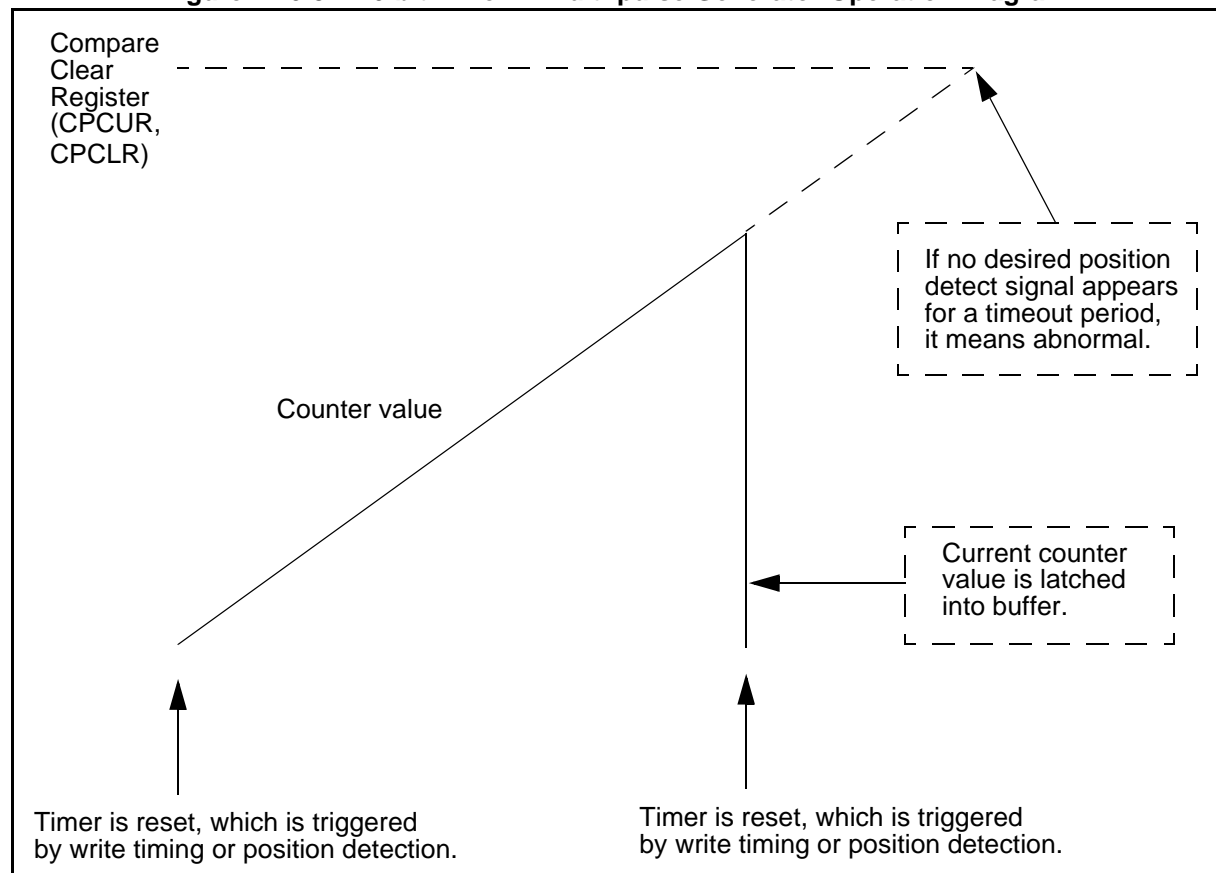
24.7 Notes on Using Multi-pulse Generator

This section provides notes on using the multi-pulse generator.

■ Notes on Using Waveform Sequencer

● Notes on using a program for setting

- Directly changing from one PPG synchronization mode to another PPG synchronization mode (e.g. from rising-edge synchronization (IPCUR:WTS1,WTS0 = 01_B) to falling-edge synchronization (IPCUR:WTS1,WTS0 = 10_B) or vice versa) is prohibited. To change from one PPG synchronization mode to another PPG synchronization mode, disable PPG edge synchronization (IPCUR:WTS1,WTS0 = 00_B) temporarily before changing to another PPG synchronization mode.

- When the data transfer method is changed, the next data buffer register to be selected is always specified by the BNKF, RDA2 to RDA0 bits in the data output register upper (OPDUR). This does not apply to the OPDBRH0 and OPDBRL0 write method (OPCUR:OPS2 to OPS0 = 000_B). In the OPDBRH0 and OPDBRL0 write method, BNKF, RDA2 to RDA0 bits are ignored.

- To access the output data register (OPDUR, OPDLR), the word access instruction must be used. Use the "MOVW" instruction to access OPDUR and OPDLR, or use the "MOV" instruction to access OPDUR first and then OPDLR.

- When using the OPDBRH0 and OPDBRL0 write method for data transfer (OPCUR:OPS2 to OPS0 = 000_B), word access to output data buffer register 0 must be used, byte access to either lower register or upper register does not start any transfer operation.

- In order to use the 16-bit reload timer underflow transfer method (OPCUR:OPS2 to OPS0 = 010_B), the reload timer should be used in "Reload Mode". Software trigger is needed to be used for the startup of the reload timer. The 16-bit reload timer is needed for setting the update time in advance and executing the continuous control action.

- In order to use the position detection and timer underflow transfer method (OPCUR:OPS2 to OPS0 = 011_B or 111_B), the reload timer should be used in "Single Shot Mode". TIN0O must be longer than two machine cycles.

- Before DTTI circuit is in effect (OPCUR:DTIE = 1), make sure that the PORTx which is multiplexed with the OPTx is configured as an output port by setting its data direction register (DDRx).

- Since the DTTI input control circuit uses a peripheral clock, input is invalidated even if the DTTI input is enabled (OPCUR:DTIE = 1) in a mode such as STOP mode in which the oscillator stops.

- In the worst situation, the time from DTTI being recognized (after noise cancellation) to DTISP in effect takes 2 cycles; in the best situation, it takes 1 cycle.

- Always change the D1 and D0 bits of noise cancellation control register (NCCR) when the noise cancellation function is disabled (OPCUR:NRSL = 0).

- Always change the S21, S20, S11, S10, S01 and S00 bits of noise cancellation control register (NCCR) when the noise cancellation function is disabled (IPCLR:SNC2 to SNC0 = 000_B).

MB95330H Series

● Notes on interrupts

- When the DTIF bit of the output control register upper (OPCUR) is set to "1", control cannot be returned from interrupt processing. Always clear the DTIF bit.

- When the WTIF bit of the output control register upper (OPCUR) is set to "1", control cannot be returned from interrupt processing. Always clear the WTIF bit.

- When the PDIF bit of the output control register lower (OPCLR) is set to "1", control cannot be returned from interrupt processing. Always clear the PDIF bit.

- When the CPIF bit of the input control register upper (IPCUR) is set to "1", control cannot be returned from interrupt processing. Always clear the CPIF bit.

- Since the above interrupts share an interrupt vector with other resource, interrupt causes must be checked carefully by the interrupt processing routine when interrupts are used.

■ Notes on Using 16-bit Timer

● Notes on using a program for setting

- To access the compare clear register (CPCUR, CPCLR) and the timer buffer register (TMBUR, TMBLR), the word access instruction must be used.

- Before the prescaler clock is changed, the timer counter should be disable first by setting the TMEN bit to "0". Change the CLK2, CLK1 and CLK0 bits of the timer control status register (TCSR) only when the timer is not counting.

- If the values loaded to the compare clear register upper (CPCUR) and the compare clear register lower (CPCLR) are the same as the timer counter value, the comparison operation will NOT be performed until the next occasion in which the values of CPCUR and CPCLR are the same as the timer counter value.

● Notes on interrupts

- When the ICLR bit of the timer control status register (TCSR) is set to "1" and an interrupt request is enabled (TCSR:ICRE = 1), control cannot be returned from interrupt processing. Always clear the ICLR bit.

- Since the 16-bit timer shares an interrupt vector with other resource, interrupt causes must be checked carefully by the interrupt processing routine when interrupts are used.

● Notes on pin occupancy

- P66 is used as MPG output when the MPG is enabled regardless of the enable state of the 16-bit PPG. P17 is shared between MPG input and the 16-bit reload timer. Therefore, it is important to ensure that only one of the three modules mentioned above is enabled to prevent their resource output from clashing. When the MPG is enabled, disable the resource output of the 16-bit PPG (PCNTL1:POEN = 0) and also that of the 16-bit reload timer (TMCSRL.OUTE = 0).

● Notes on function conflict

- The 16-bit PPG and the 16-bit reload timer form part of the MPG. When the MPG is enabled, the two modules are used for the MPG and cannot work independently of the MPG. When the 16-bit PPG or the 16-bit reload timer is needed for other applications, disable the MPG first before using them for other applications.

24.8 Sample Program for Multi-pulse Generator

This section provides a sample program for the multi-pulse generator.

■ Sample Program for Multi-pulse Generator

● Processing

- An output in PPG is directed to OPT0 and an inverted output in PPG is directed to OPT1 when write timing interrupt is generated.

- The OPDBRH0 and OPDBRL0 write method is used for data transfer to output data register (OPDUR, OPDLR).

- The 16-bit PPG timer is used in PWM and is started with a software trigger.

- 16 MHz is used for the machine clock, and 62.5 ns is used for the count clock of the 16-bit PPG timer.

● Coding example

```
;-------A demo program------------------------------------------------------------------------
ILR4       EQU    007DH        ;Interrupt control register for the waveform sequencer
PCSR1      EQU    0FB2H        ;16-bit PPG cycle setting buffer register
PDUT1      EQU    0FB4H        ;PPG duty setting register
PCNT1      EQU    0044H        ;PPG control status register
OPCUR      EQU    0066H        ;Output control register upper
OPCLR      EQU    0067H        ;Output control register lower
OPCR       EQU    OPCUR        ;Output control register upper+lower
                               ; ,for word access.
OPDBRH0    EQU    0FC4H        ;Output data buffer register 0 upper
OPDBRL0    EQU    0FC4H        ;Output data buffer register 0 lower
OPDBR0     EQU    OPDBRH0      ;Output data buffer register 0 upper+lower
                               ; ,for word access.
WTIF       EQU    OPCUR:1      ;Interrupt request flag bit
;-------Main program-------------------------------------------------------------------------
CODE       CSEG   ABS
START:
;          :                   ;Assumes that stack pointer (SP) has already been
           CLRI                ;Interrupt disable
           MOV    ILR4,#00H     ;Interrupt level 0 (strongest)
           MOVW   A,#0064H
           MOVW   PCSR1,A       ;Sets the period of the PPG output
           MOVW   A,#003CH
           MOVW   PDUT1,A       ;Sets the duty ratio of the PPG output
           MOVW   A,#0110000000000110B
```

```
                MOVW    PCNT1,A         ;Enables PPG output in normal polarity
                                        ;Enables 16-bit PPG timer
                                        ;Software triggers PPG
                                        ;Select PWM mod
                                        ;Clears interrupt flag, and starts counter
                MOVW    A,#0103H
                MOVW    OPCR,A          ;Enable OPT0 and OPT1 output
                                        ;Sets OPDBRH0 and OPDBRL0 write method for
                                         data transfer
                                        ;Enable write timing interrupt
                                        ;Clears interrupt flag
                MOVW    A,#0009H
                MOVW    OPDBR0,A        ;Sets OPT0 pin as PPG output
                                        ;Sets OPT1 pin as inverted PPG output
                                        ;Starts data transfer
                SETI                    ;Interrupt enable
LOOP:           MOV     A,#00H          ;Endless loop
                MOV     A,#01H;
                JMP     LOOP;
;-------Interrupt program-------------------------------------------------------------------
WARI:
                CLRB    WTIF            ;Clears interrupt request flag
                ;  ;
                ;   User processing
                ;  :
                RETI                    ;Returns from interrupt
CODE ENDS
;-------Vector setting----------------------------------------------------------------------
VECT            CSEG    ABS
                ORG     0FFDAH          ;Sets vector for interrupt #16 (10H)
                DW WARI
                ORG     0FFFCH          ;Sets reset vector
                DW 0000H                ;Sets single-chip mode
                DW START
VECT ENDS
                END     START
                END
```

FUJITSU SEMICONDUCTOR LIMITED

CHAPTER 25

UART/SIO

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

25.1   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 25.1-1.)
- 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 25.1-1 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	-		

MB95330H Series

25.2 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 25.2-1 Block Diagram of UART/SIO
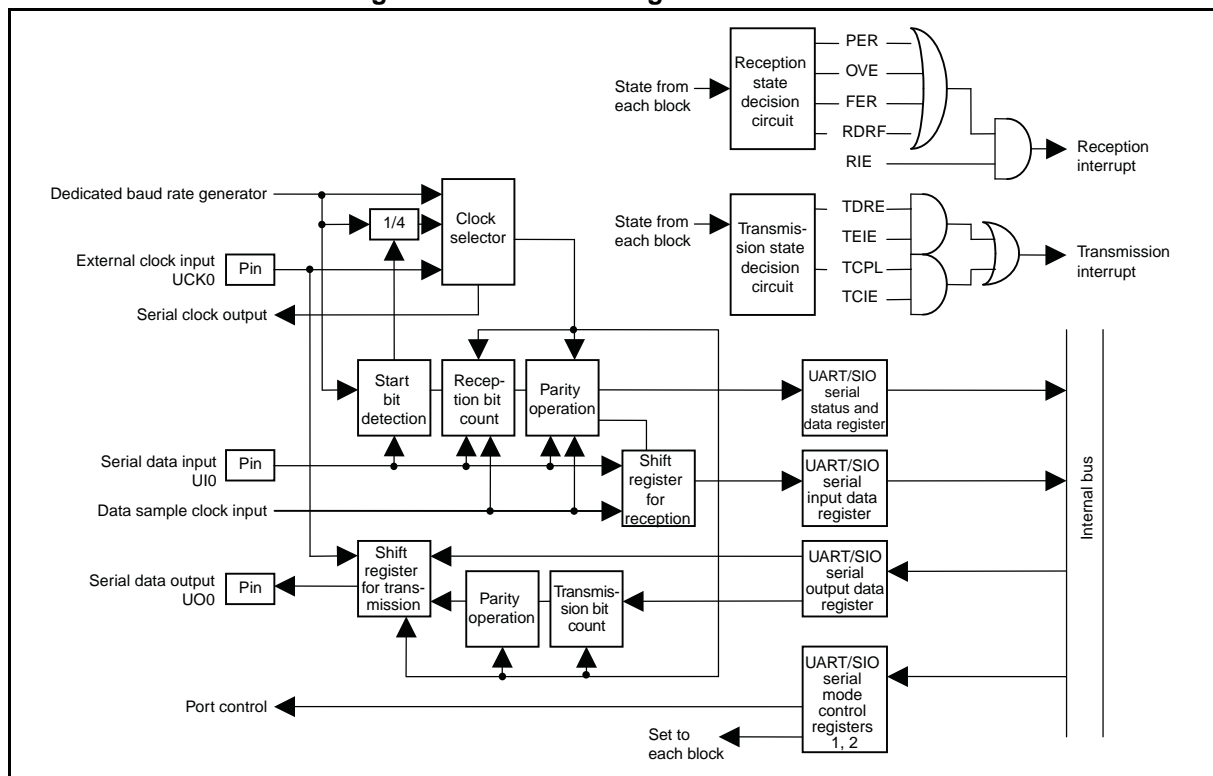
● UART/SIO serial mode control register 1 (SMC10)

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 UCK0 pin as its input clock (serial clock).

MB95330H Series

25.3   Channels of UART/SIO

This section describes the channels of UART/SIO.

■ Channels of UART/SIO

The MB95330H Series has one channel of UART/SIO.

Table 25.3-1 and Table 25.3-2 show the pins and registers of UART/SIO respectively.

Table 25.3-1 Pins of UART/SIO

Channel	Pin name	Pin function
0	UCK0	Clock input/output
	UO0	Data output
	UI0	Data input

Table 25.3-2 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

25.4　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 (UCK0), serial data output pin (UO0) and serial data input pin (UI0).

● UCK0

Clock input/output pin for UART/SIO.

When the clock output is enabled (SMC20:SCKE=1), it serves as a UART/SIO clock output pin (UCK0) regardless of the value of the corresponding port direction register. At this time, do not select the external clock (set SMC10:CKS = 0).

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

● UO0

Serial data output pin for UART/SIO. When the serial data output is enabled (SMC20:TXOE = 1), it serves as a UART/SIO serial data output pin (UO0) regardless of the value of the corresponding port direction register.

● UI0

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

MB95330H Series

■ Block Diagrams of Pins of UART/SIO

Figure 25.4-1 Block Diagram of Pin UO0 (P15/UO0/PPG20) of UART/SIO
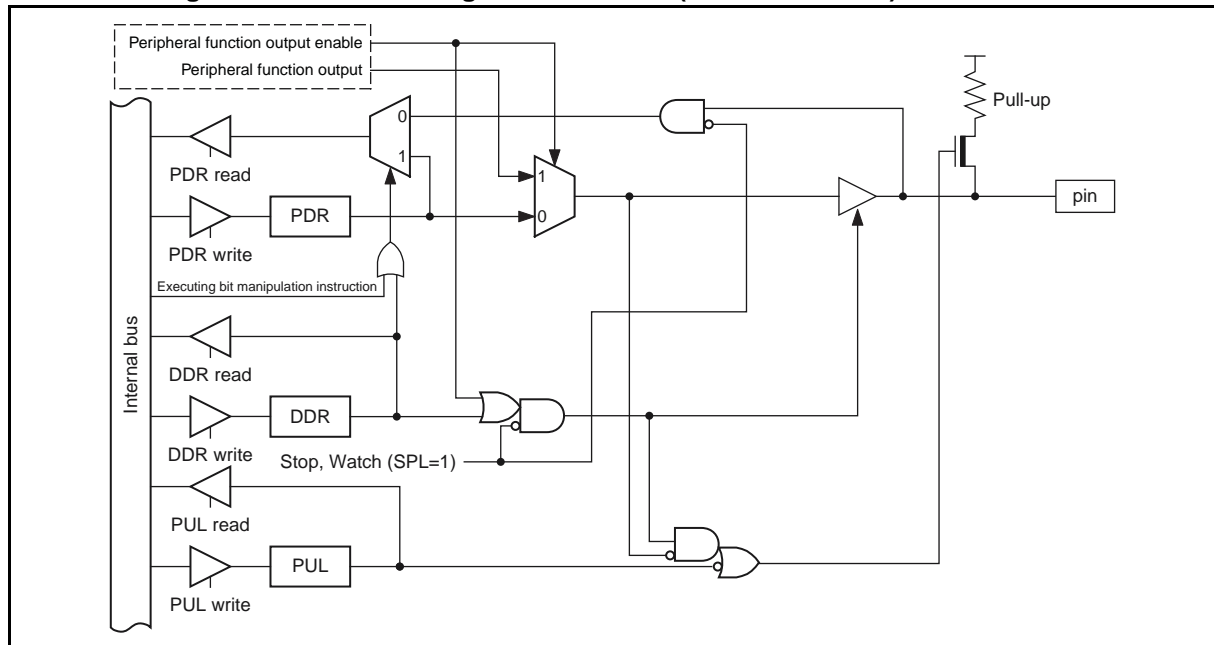
Figure 25.4-2 Block Diagram of Pin UCK0 (P14/UCK0/PPG01) of UART/SIO
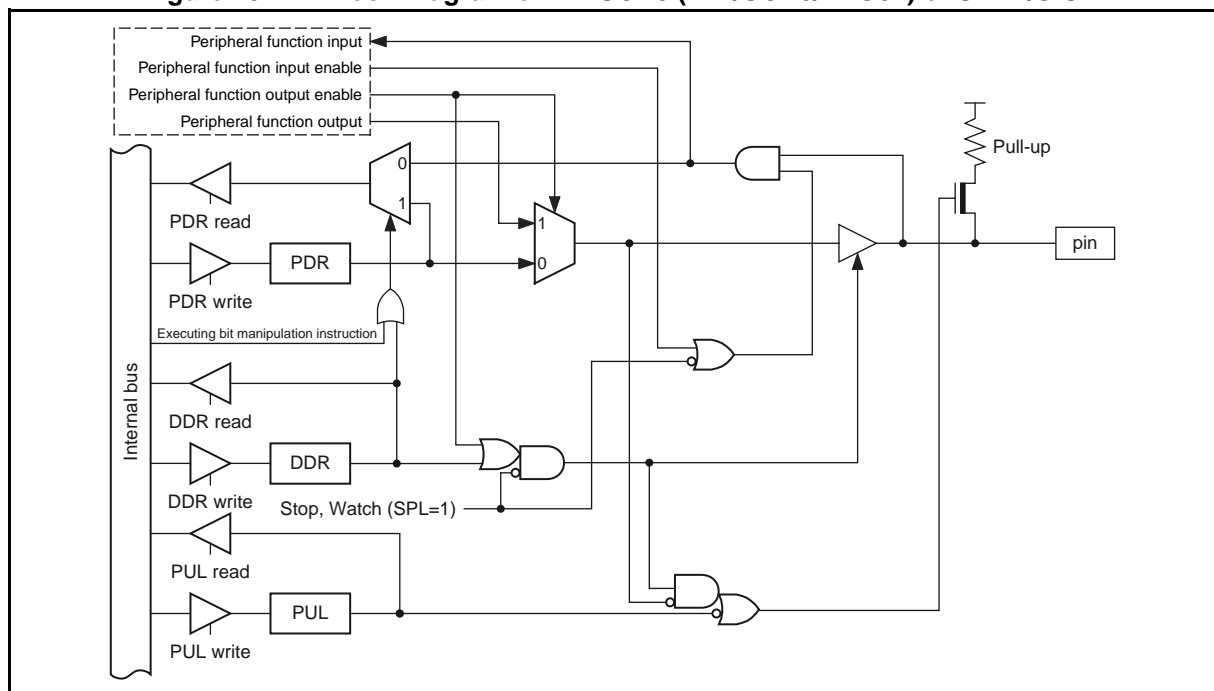
Figure 25.4-3 Block Diagram of Pin UI0 (P16/UI0/PPG21) of UART/SIO

MB95330H Series
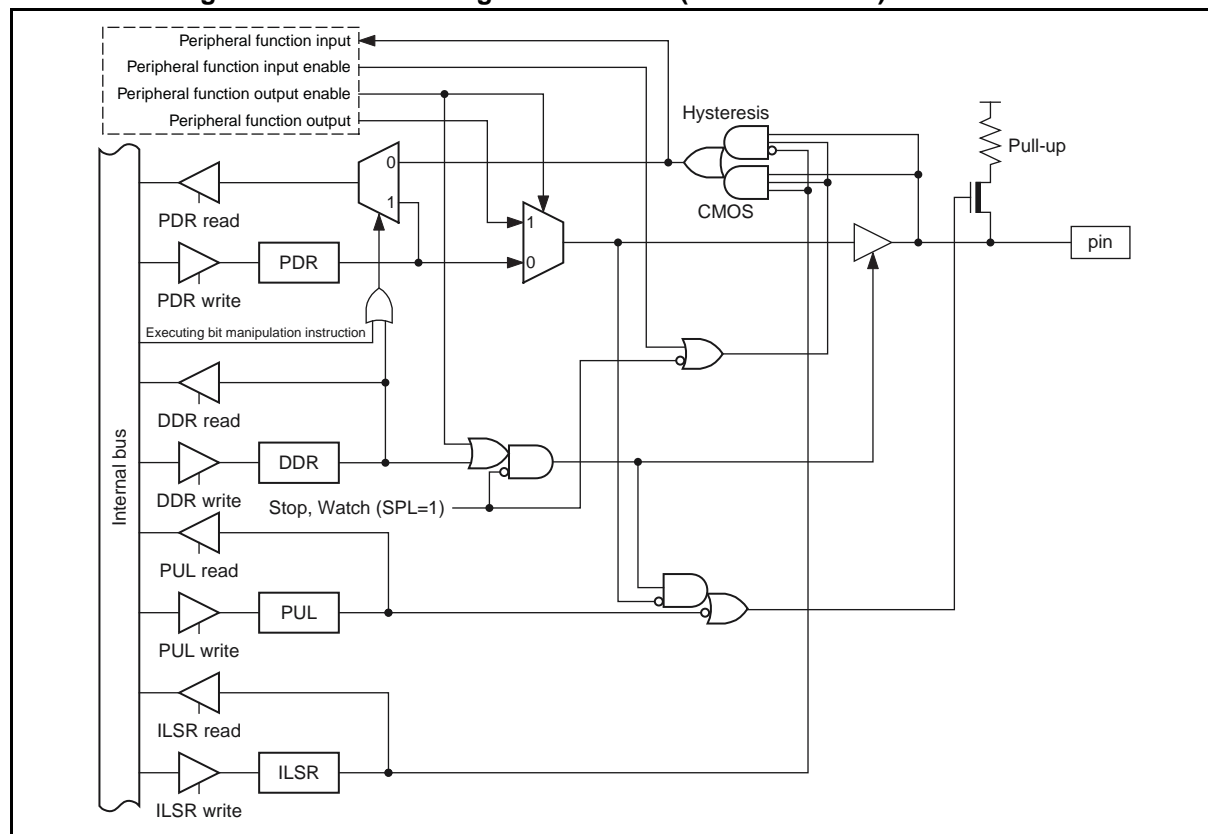
25.5 Registers of UART/SIO

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 25.5-1 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

25.5.1 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 25.5-2 UART/SIO Serial Mode Control Register 1 (SMC10)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 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 25.5-1 Functions of Bits in UART/SIO Serial Mode Control Register 1 (SMC10)

Bit name		Function
bit7	BDS: Serial data direction control bit	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.
bit6	PEN: Parity control bit	This bit enables or disables parity in clock asynchronous mode. **Writing "0":** no parity **Writing "1":** with parity
bit5	TDP: Parity polarity bit	This bit controls even/odd parity. **Writing "0":** specifies even parity **Writing "1":** specifies odd parity
bit4	SBL: Stop bit length control bit	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 set to "1" after detecting stop bit(1-bit) and completing the reception regardless of this bit.
bit3, bit2	CBL1, CBL0: Character bit length control bits	These bits select the character bit length as shown in the following table: CBL1 / CBL0 / Character bit length 0 / 0 / 5 0 / 1 / 6 1 / 0 / 7 1 / 1 / 8 The above setting is valid in both asynchronous and synchronous modes.
bit1	CKS: Clock select bit	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 UCK0 pin. The external clock cannot be used in clock asynchronous mode (UART).
bit0	MD: Operation mode select bit	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).

Note:

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

25.5.2 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 25.5-3 UART/SIO Serial Mode Control Register 2 (SMC20)

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

TEIE	Transmit data register empty interrupt enable bit
0	Disables transmit data register empty interrupts.
1	Enables transmit data register empty interrupts.

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

RIE	Receive interrupt enable bit
0	Disables receive interrupts.
1	Enables receive interrupts.

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 a general-purpose port).
1	Enables serial data output.

SCKE	Serial clock output enable bit
0	Disables serial clock output (usable as a general-purpose 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 25.5-2 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 (UCK0) 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 (UO0 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 error flags (PER, OVE, FER, 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 transmit data register empty (TDRE) bit will also be set to "1".
bit2	RIE: Receive interrupt enable bit	**Writing "0":** disables receive interrupts. **Writing "1":** enables receive interrupts. 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: Transmit data register empty interrupt enable bit	**Writing "0":** disables interrupts by the transmit data register empty. **Writing "1":** enables interrupts by the transmit data register empty. A transmit interrupt occurs immediately after the transmit data register empty (TDRE) bit is set to "1" with this bit set to "1" (enabled).

25.5.3 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 25.5-4 UART/SIO Serial Status and Data Register (SSR0)

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

TDRE	Transmit 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	Receive data register full flag
0	Receive data absent
1	Receive data present

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

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

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

R(RM1),W : Readable/writable (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 25.5-3 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: Transmit 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.

25.5.4 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 25.5-5 shows the bit configuration of the UART/SIO serial input data register.

Figure 25.5-5 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/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 (UI0 pin) is converted by the shift register and stored in this register.

When received data is set correctly in this register, the receive data register full (RDRF) bit is set to "1". At this time, an interrupt occurs if 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".

MB95330H Series

25.5.5   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 25.5-6 shows the bit configuration of the UART/SIO serial output data register.

Figure 25.5-6 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	

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 transmit data register empty bit (TDRE) 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 transmit data register empty bit (TDRE) is set to "0". Upon completion of transfer of transmit data to the transmission shift register, the transmit 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 transmit 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 and 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.

25.6 Interrupts of UART/SIO

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

■ Interrupts of UART/SIO

Table 25.6-1 lists the UART/SIO interrupt control bits and interrupt sources.

Table 25.6-1 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	Transmit 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 25.6-2 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

*: UART/SIO ch. 0 shares the interrupt request number and vector table addresses mentioned in the table with MPG (DTTI).

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

MB95330H Series

25.7 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 25.7-1).

Table 25.7-1 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)

25.7.1 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 26 "UART/SIO DEDICATED BAUD RATE GENERATOR".)

Figure 25.7-1 Baud Rate Calculation when Using Dedicated Baud Rate Generator

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

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

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

Table 25.7-2 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 (10 MHz / Total division ratio)	Baud rate (16 MHz / Total division ratio)	Baud rate (16.25 MHz / 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

MB95330H Series

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

Table 25.7-3 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 25.7-2 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 25.7-2, 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 25.7-2 Transfer Data Format
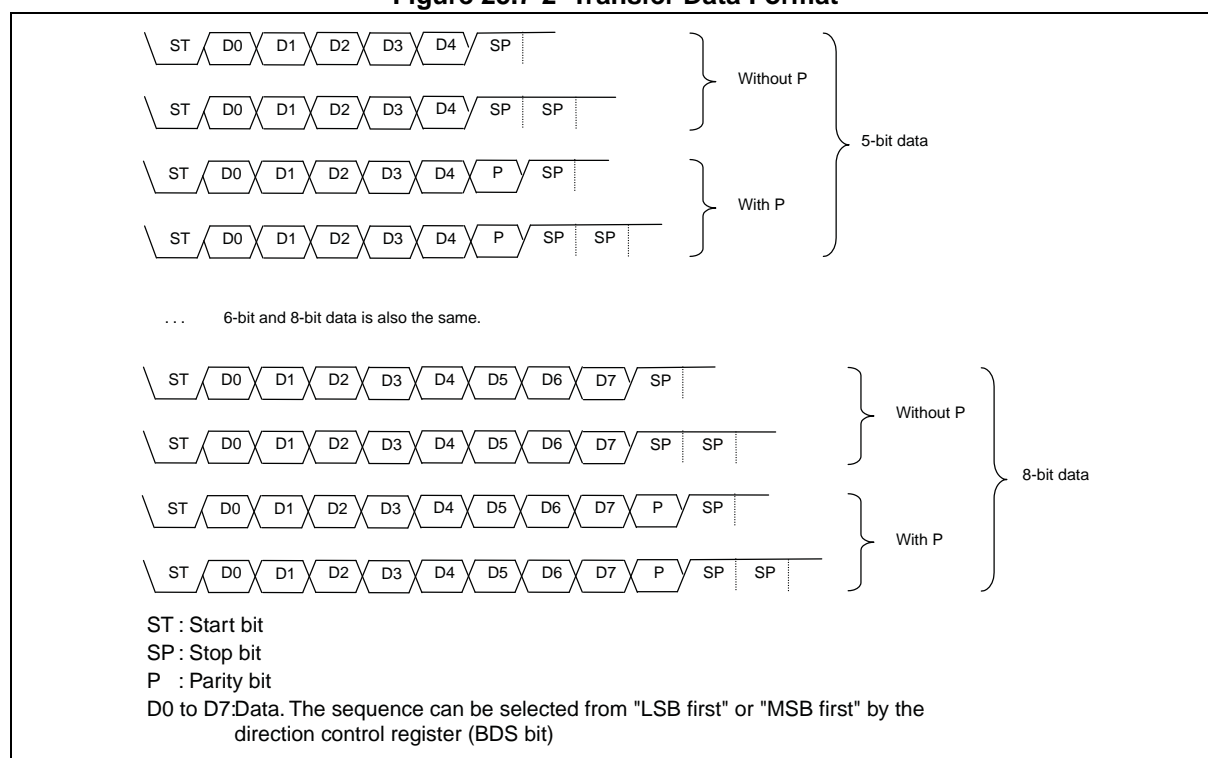
● Receiving operation in asynchronous clock mode (UART)

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 25.7-3 Receiving Operation in Asynchronous Clock Mode
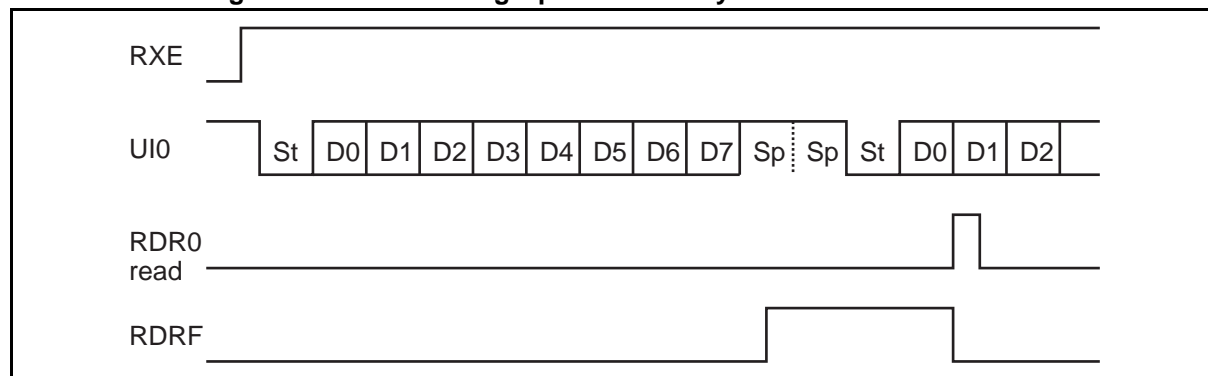
● Reception error in asynchronous clock mode (UART)

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 25.7-4 Setting Timing for Receiving Errors
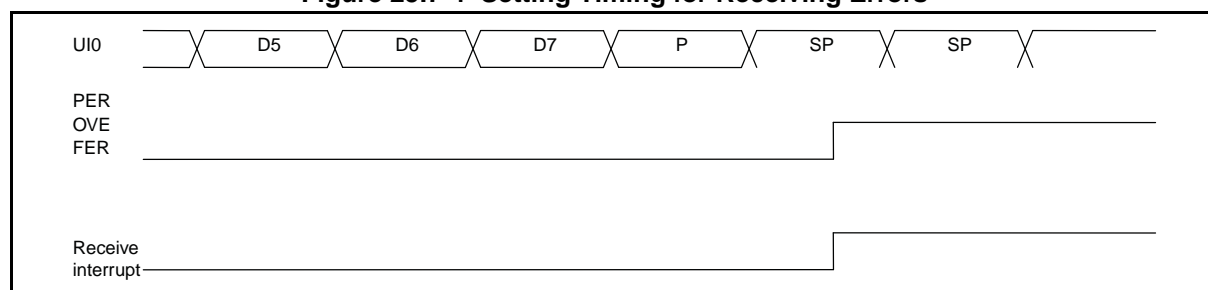
● Start bit detection and confirmation of receive data during reception

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 25.7-5 Start Bit Detection and Serial Data Input
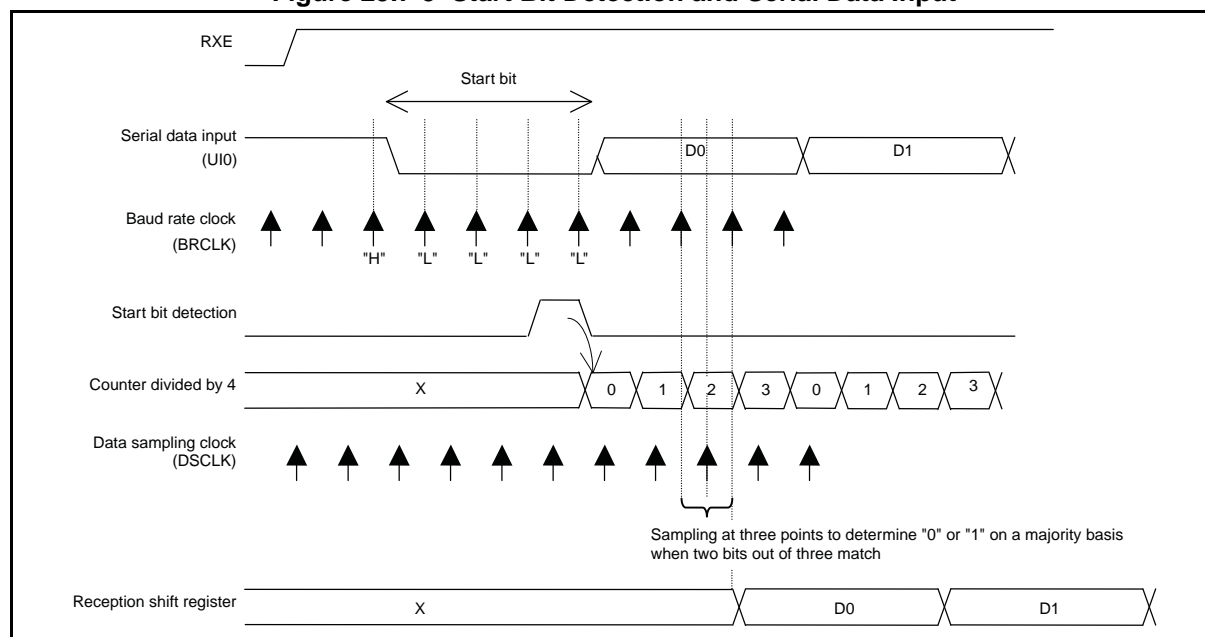
● Transmission in asynchronous clock mode

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 transmit 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 25.7-6 below.

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

Figure 25.7-6 Transmission in Asynchronous Clock Mode (UART)
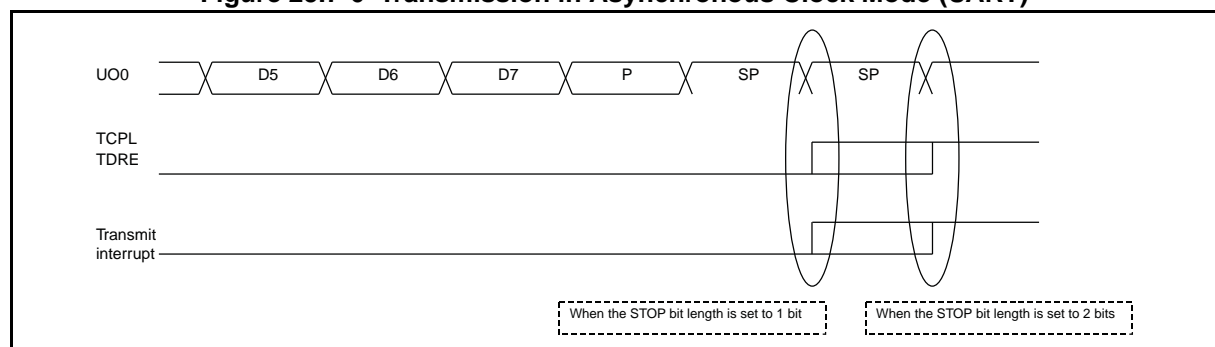
The TDRE flag is set at the point indicated in the following figure if the preceding piece of transmit data does not exist in the transmission shift register.

Figure 25.7-7 Setting Timing 1 for Transmit Data Register Empty Flag (TDRE) (When TXE is "1")
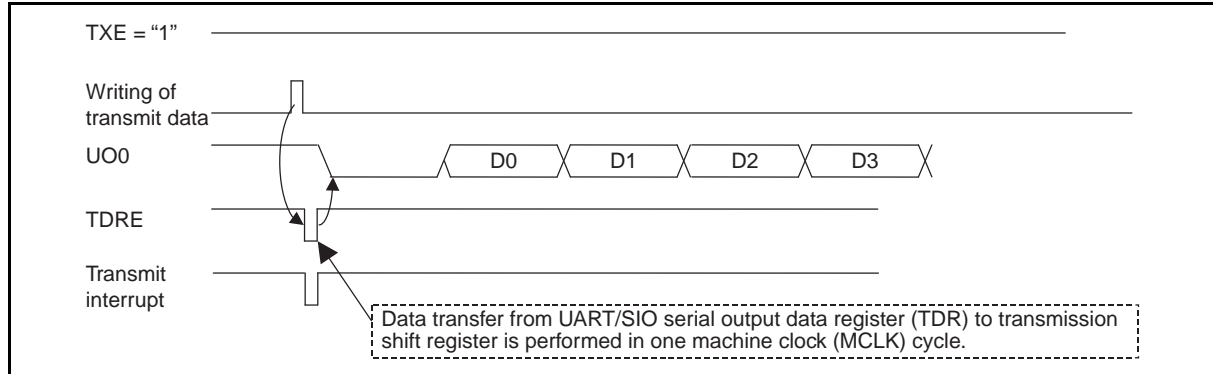
Figure 25.7-8 Setting Timing 2 for Transmit Data Register Empty Flag (TDRE) (When TXE Is Switched from "0" to "1")
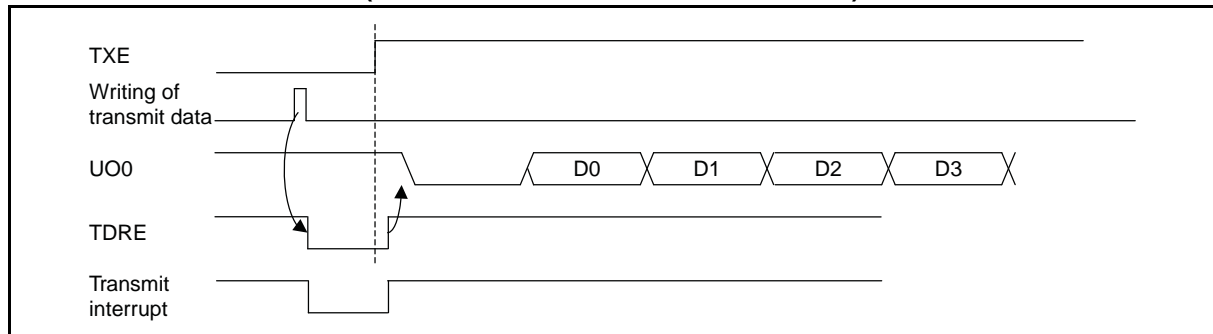
● Concurrent transmission and reception

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.

MB95330H Series

25.7.2   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 26 "UART/SIO DEDICATED BAUD RATE GENERATOR".)

Table 25.7-4 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 25.7-9 Calculating Baud Rate Based on External Clock

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

More than 4 machine clock

*: External clock

More than 4 machine clock
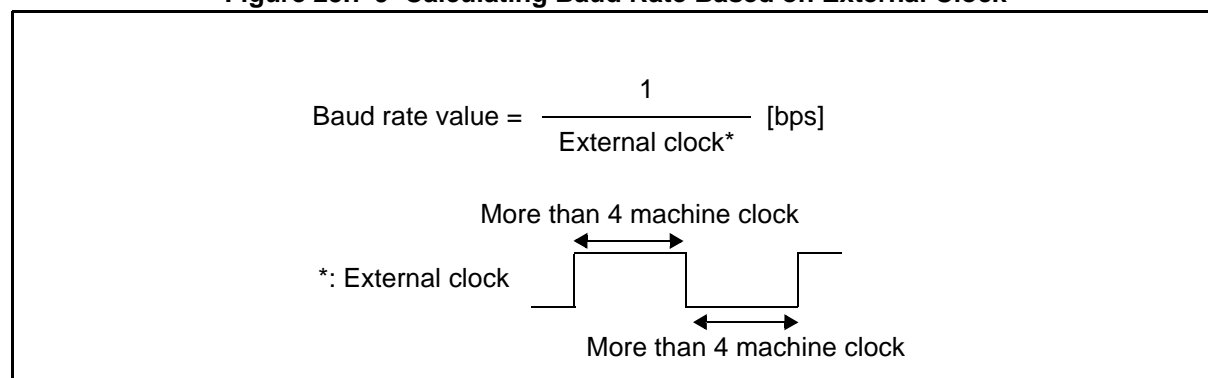
Figure 25.7-10 Baud Rate Calculation Formula for Using Dedicated Baud Rate Generator

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

UART prescaler select register (PSSR0)   UART baud rate setting register (BRSR0)
Prescaler select (PSS1, PSS0)   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 MB95330H Series for the UCK0 clock value.

● Reception in UART/SIO operation mode 1

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

MB95330H Series

Figure 25.7-11 Registers Used for Reception in Operation Mode 1

SMC10 (UART/SIO serial mode control register 1)

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

SMC20 (UART/SIO serial mode control register 2)

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

SSR0 (UART/SIO serial status and data register)

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

TDR0 (UART/SIO serial output data register)

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

RDR0 (UART/SIO serial input data register)

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

◎ : Used bit
× : Unused bit
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 25.7-12 8-bit Reception of Synchronous Clock Mode
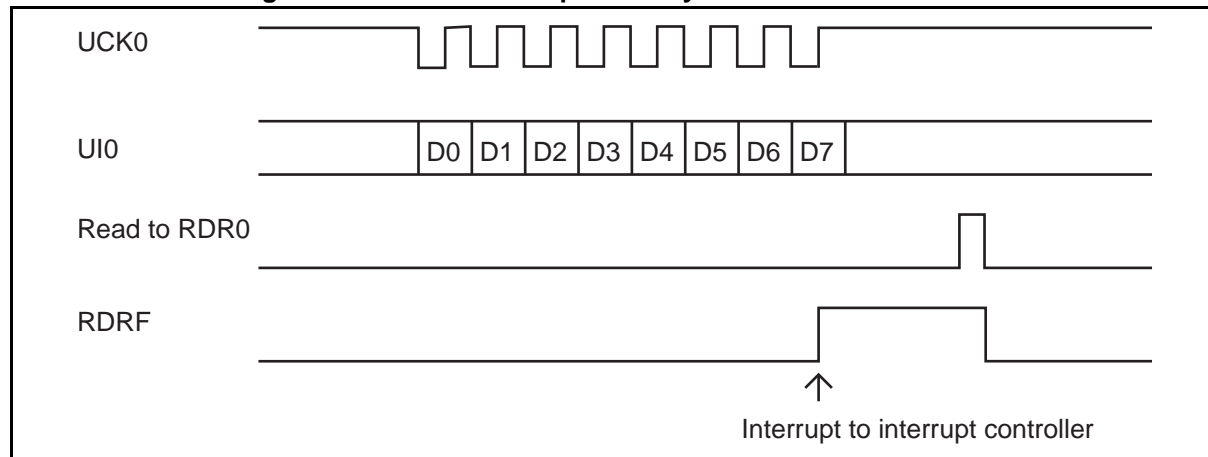
Operation when reception error occurs

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

Overrun error (OVE)

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

Figure 25.7-13 Overrun Error

MB95330H Series
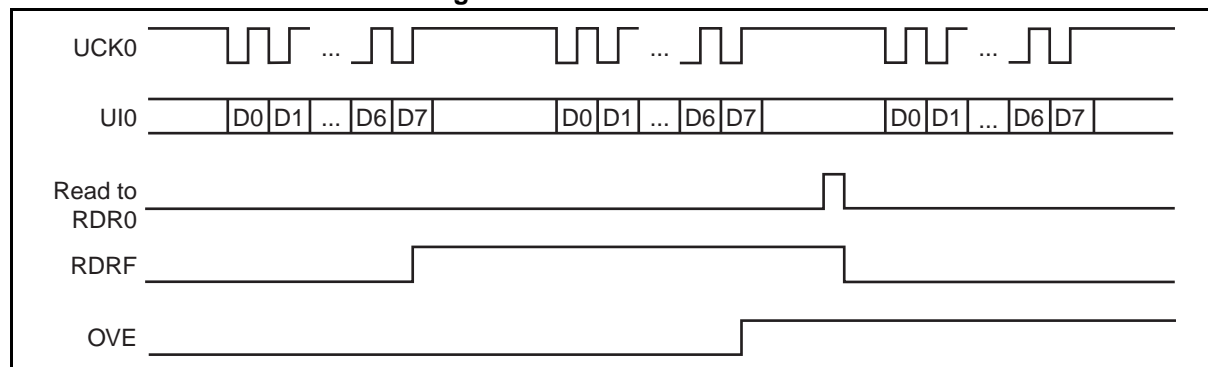
● Transmission in UART/SIO operation mode 1

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

Figure 25.7-14 Registers Used for Transmission in Operation Mode 1

SMC10 (UART/SIO serial mode control register 1)

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

SMC20 (UART/SIO serial mode control register 2)

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

SSR0 (UART/SIO serial status and data register)

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

TDR0 (UART/SIO serial output data register)

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

RDR0 (UART/SIO serial input data register)

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

◎ : Used bit
× : Unused bit
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 25.7-15 8-bit Transmission in Synchronous Clock Mode
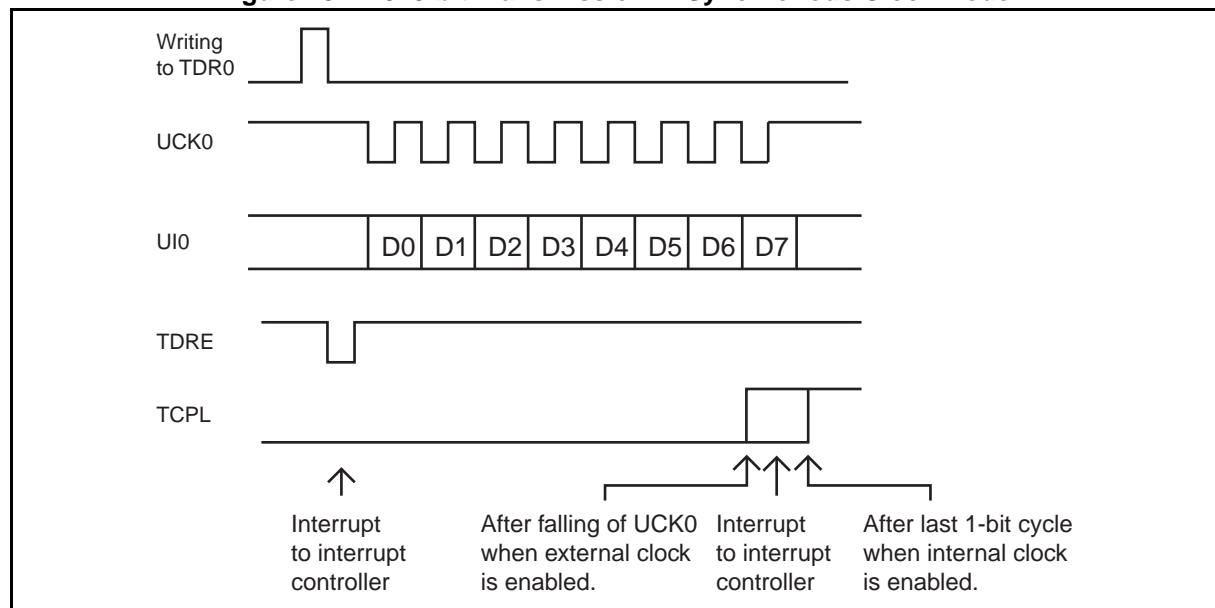
● Concurrent transmission and reception

<When external clock is enabled>

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

<When internal clock is enabled>

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

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

• See Section 25.4 "Pins of UART/SIO" for operation with serial clock output and operation with serial clock input.

MB95330H Series

25.8   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 UCK0, UI0, or UO0 pin

The following setting is used.

	UART
To set UCK0 pin as an input	DDR1:P14 = 0 SMC20:SCKE = 0
To set UCK0 pin as an output	SMC20:SCKE = 1
To use UI0 pin	DDR1:P16 = 0
To use UO0 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 the bit to "0".
To set the STOP bit to 2-bit length	Set the bit to "1".

MB95330H Series

● 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 Section 25.7.1 "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)	Transmit 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 transmit 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 26
UART/SIO DEDICATED BAUD RATE GENERATOR

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

26.1 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 select register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Block Diagram of UART/SIO Dedicated Baud Rate Generator

Figure 26.1-1 Block Diagram of UART/SIO Dedicated Baud Rate Generator
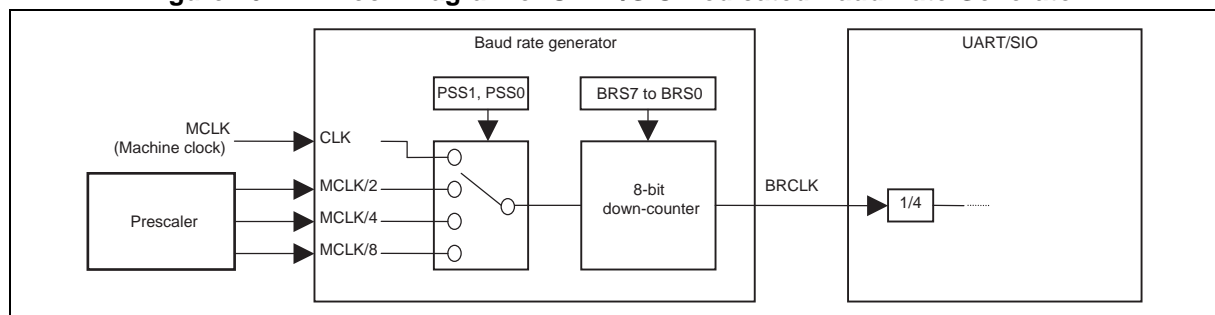
■ Input Clock

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.

26.2 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 MB95330H Series has one channel of UART/SIO dedicated baud rate generator.

Table 26.2-1 shows the registers of the UART/SIO dedicated baud rate generator.

Table 26.2-1 Registers of Dedicated Baud Rate Generator

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

26.3 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 select register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

■ Registers of UART/SIO Dedicated Baud Rate Generator

Figure 26.3-1 Registers of UART/SIO Dedicated Baud Rate Generator

UART/SIO dedicated baud rate generator prescaler select 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

MB95330H Series

26.3.1 UART/SIO Dedicated Baud Rate Generator Prescaler Select Register (PSSR0)

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

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

Figure 26.3-2 UART/SIO Dedicated Baud Rate Generator Prescaler Select Register (PSSR0)
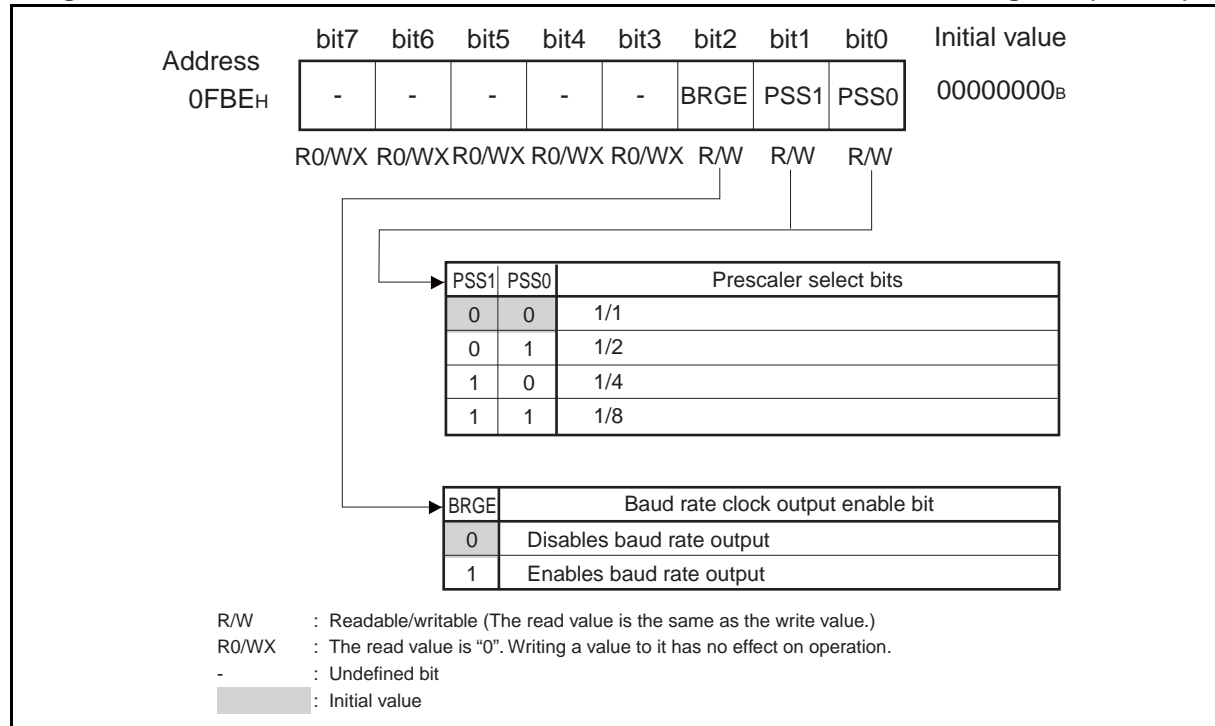
Table 26.3-1 Functions of Bits in UART/SIO Dedicated Baud Rate Generator Prescaler Select Register (PSSR0)

	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	BRGE: Baud rate clock output enable bit	This bit enables the output of the baud rate clock "BRCLK". **Writing "1":** loads BRS[7:0] to the 8-bit down-counter and outputs "BRCLK", which is supplied to the UART/SIO. **Writing "0":** stops the output of "BRCLK".
bit1, bit0	PSS1, PSS0: Prescaler select bits	<table><tr><td>PSS1</td><td>PSS0</td><td>Prescaler select</td></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>

26.3.2 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 26.3-3 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.

26.4 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 clock 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 26.4-1 Baud Rate Calculation Formula when UART/SIO Dedicated Baud Rate Generator Is Used

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

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 26.4-1 Sample 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 × 4)	Baud rate (10 MHz / Total division ratio)	Baud rate (16 MHz / Total division ratio)	Baud rate (16.25 MHz / 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 26.4-2 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 27

I^2C

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

27.1 Overview of I²C

The I²C interface supports the I²C 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.

■ I²C Functions

The I²C 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 I²C interface can connect multiple devices provided the bus capacitance does not exceed an upper limit of 400 pF. The I²C 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 I²C interface includes a function to wake up the MCU from standby mode.

Figure 27.1-1 I²C Interface Configuration

MB95330H Series
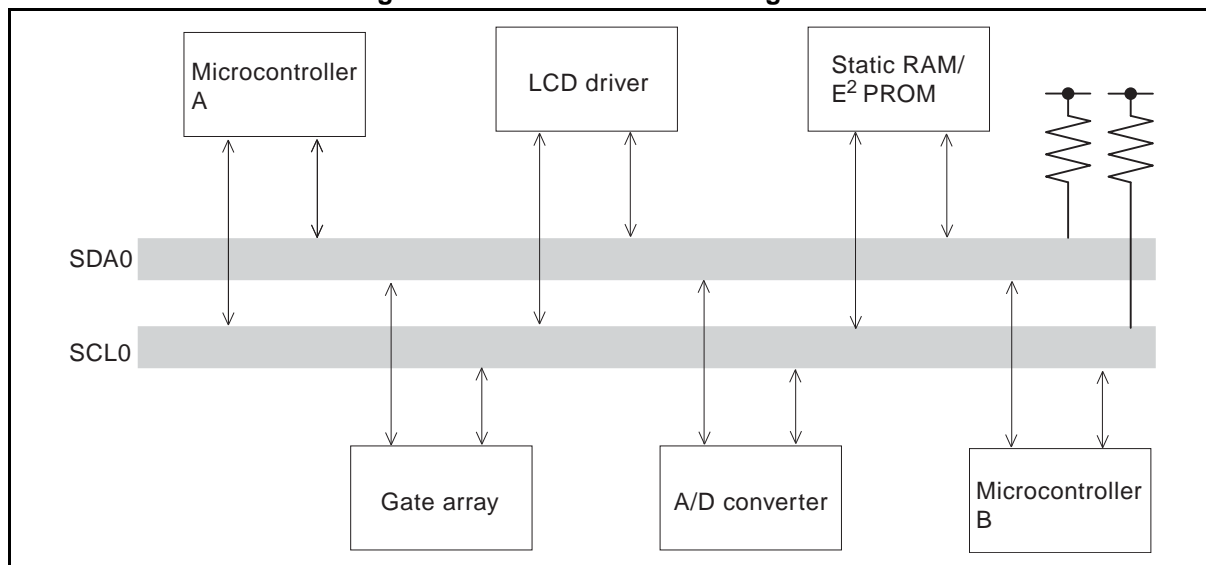
27.2   I²C Configuration

I²C 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 I²C

Figure 27.2-1 Block Diagram of I²C
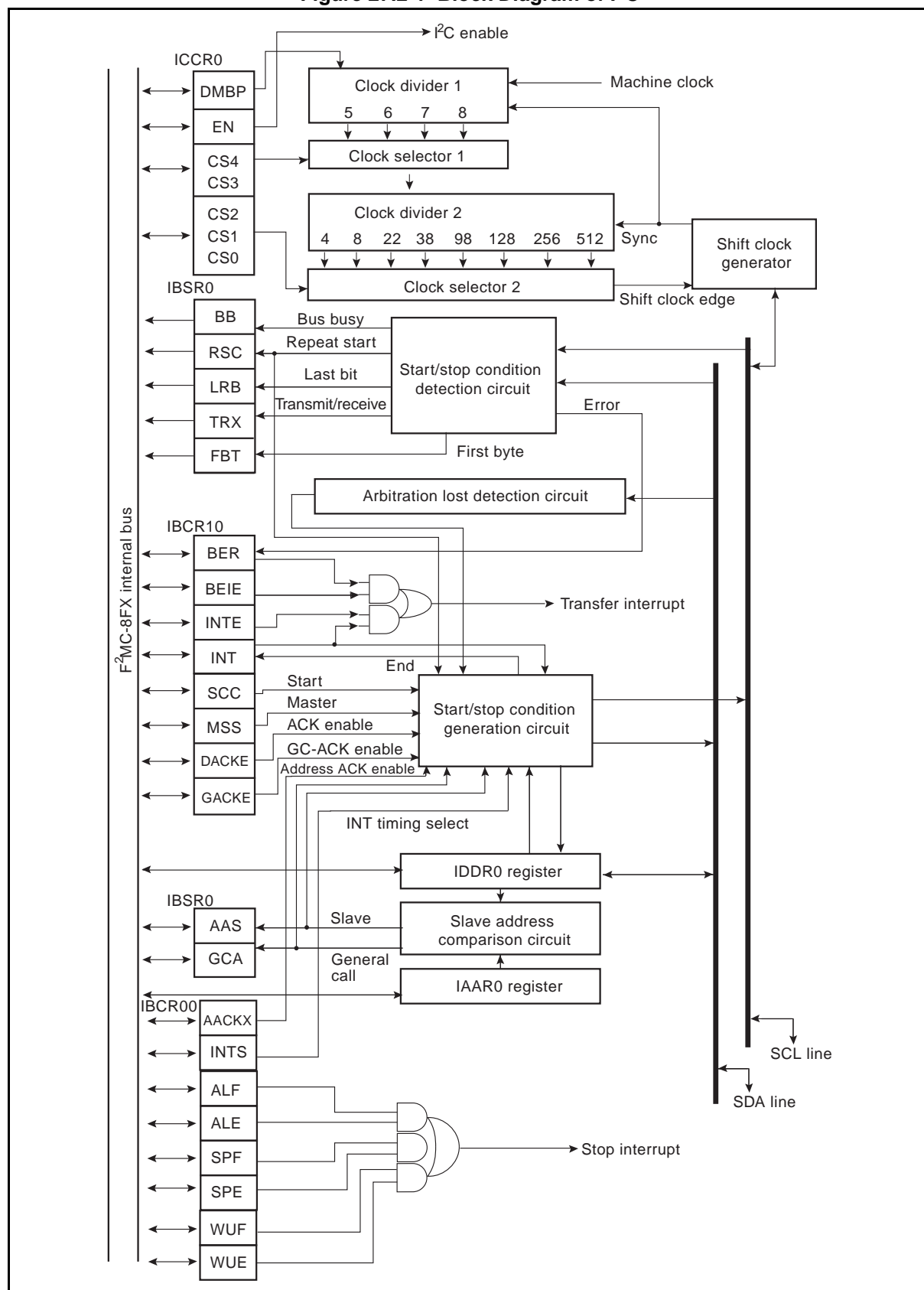
● Clock selector, clock divider, and shift clock generator

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).

MB95330H Series

27.3   I^{2}C Channel

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

■ I^{2}C Channel

The MB95330H Series has one channel of I^{2}C.

Table 27.3-1 and Table 27.3-2 show the pins and registers of I^{2}C respectively.

Table 27.3-1 Pins of I^{2}C

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

Table 27.3-2 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

27.4 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).

● 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).

MB95330H Series

■ Block Diagram of Pins of I^{2}C Bus Interface

Figure 27.4-1 Block Diagram of Pins SCL and SDA (P61/INT09/SCL/TI1 and P60/INT08/SDA/DTTI) of I^{2}C Bus Interface
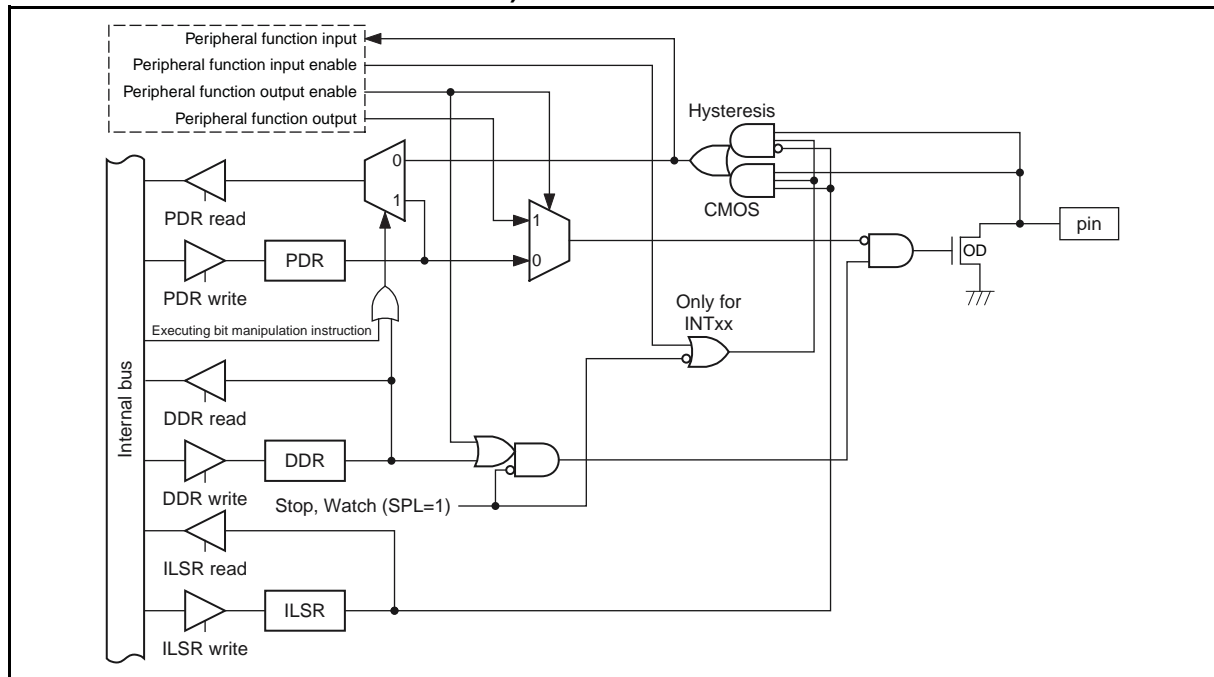
27.5 Registers of I^{2}C

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

■ Registers of I^{2}C

Figure 27.5-1 Registers of I^{2}C

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

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0060$_H$	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
0061$_H$	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
0062$_H$	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
0063$_H$	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
0064$_H$	-	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
0065$_H$	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

MB95330H Series

27.5.1 I^{2}C Bus Control Registers (IBCR00, IBCR10)

The I^{2}C 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.

■ I^{2}C Bus Control Register 0 (IBCR00)

Figure 27.5-2 I^{2}C Bus Control Register 0 (IBCR00)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0060H	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	

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 27.5-1 Functions of Bits in I²C 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 I²C 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 I²C 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 the read-modify-write (RMW) type of 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 the read-modify-write (RMW) type of instruction.

Table 27.5-1 Functions of Bits in I²C 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 the read-modify-write (RMW) type of 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 I²C operation. Notes: • Write "1" to this bit immediately before the MCU enters the stop or watch mode. To ensure that I²C 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 I²C transmission (upon detection of the falling edge of SDA). • During a MCU standby mode, the status flags, state machine, and I²C bus outputs for the I²C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I²C 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 I²C operation is disabled (ICCR0:EN = 0) or when a bus error occurs (IBCR10:BER = 1).

■ I²C Bus Control Register 1 (IBCR10)

Figure 27.5-3 I²C Bus Control Register 1 (IBCR10)

Table 27.5-2 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 27.5-2 Functions of Bits in I²C 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 (ICCR0:EN = 0) or when a bus error occurs (IBCR10:BER = 1).

MB95330H Series

27.5.2   I²C Bus Status Register (IBSR0)

The IBSR0 register indicates the status of the I²C interface.

■ I²C Bus Status Register (IBSR0)

Figure 27.5-4 I²C Bus Status Register (IBSR0)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0062H	BB	RSC	-	LRB	TRX	AAS	GCA	FBT	00000000B
	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 (00H) not received in slave mode.
1	General call address (00H) 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 : The read value is "0". Writing a value to it has no effect on operation.

\-   : Undefined bit

▒   : Initial value

Table 27.5-3 Functions of Bits in I²C 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.

MB95330H Series

27.5.3 I²C Data Register (IDDR0)

The IDDR0 register is used to set the data or address to send and to hold the data or address received.

■ I²C Data Register (IDDR0)

Figure 27.5-5 I²C Data Register (IDDR0)

I²C data register (IDDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0063H	D7	D6	D5	D4	D3	D2	D1	D0	00000000B
	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.

27.5.4   I^{2}C Address Register (IAAR0)

The IAAR0 register is used to set the slave address.

■ I^{2}C Address Register (IAAR0)

Figure 27.5-6 I^{2}C Address Register (IAAR0)

I^{2}C address register (IAAR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064$_H$	-	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.

27.5.5 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 27.5-7 I^{2}C Clock Control Register (ICCR0)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 0065$_H$	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	

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	The settings of CS4 and CS3 (divider m) are effective.
1	The settings of CS4 and CS3 (divider m) are not effective.

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 27.5-4 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: $$Fsck = \frac{\phi}{(m \times n + 2)}$$ 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.

MB95330H Series

27.6   I²C Interrupts

The I²C 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 I²C interface in stop/watch mode.

■ Transfer Interrupt

Table 27.6-1 shows the transfer interrupt control bits and I²C interrupt sources.

Table 27.6-1 Transfer Interrupt Control Bits and I²C 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 I²C 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 27.6-2 shows the stop interrupt control bits and I^{2}C interrupt sources (trigger events).

Table 27.6-2 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.

MB95330H Series

■ Register and Vector Table Addresses Related to I²C Interrupts

Table 27.6-3 Register and Vector Table Addresses Related to I²C Interrupts

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
I²C*	IRQ16	ILR4	L16	FFDA$_H$	FFDB$_H$

*: The I²C 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.

27.7   Operations of I^{2}C and Setting Procedure Example

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:

● Initial settings

1) Set the port for input (DDR6).
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 (IBCR10:BEIE = 1).

● Interrupt processing

1) Arbitrary processing
2) Clear the bus error interrupt request flag (IBCR10:BER = 0).

MB95330H Series

27.7.1   I²C Interface

The I²C interface is an eight-bit serial interface synchronized with the shift clock. It conforms to the I²C bus specification defined by Philips.

■ I²C System

The I²C 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 I²C 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.

■ I²C Protocol

Figure 27.7-1 shows the format required for data transfer.

Figure 27.7-1 Data Transfer Example
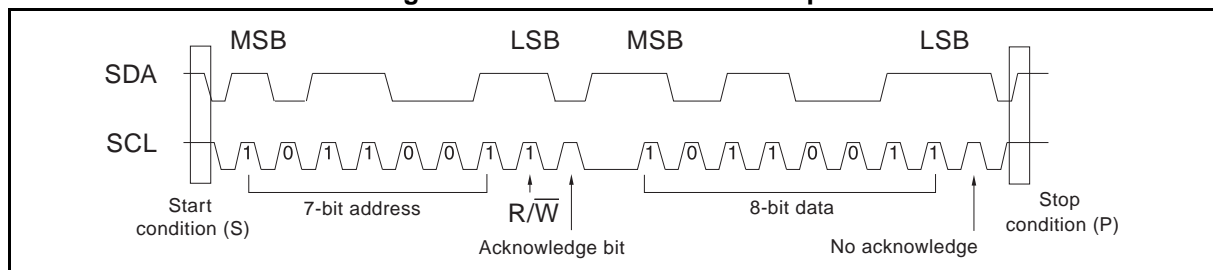
The slave address is transmitted after a start condition (S) is generated. This address is seven bits followed by the data direction bit (R/$\overline{W}$) in the eighth bit position. Data is transmitted after the address. The data is eight bits followed by an acknowledgment.

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 27.7-1, 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 I²C 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/$\overline{\text{W}}$ 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 27.7-1). In this case, the R/$\overline{\text{W}}$ 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/$\overline{\text{W}}$ bit sent by the master.

Each byte to be output on the SDA line is fixed at eight bits. As shown in Figure 27.7-1, 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 27.7-2 General Call Operation
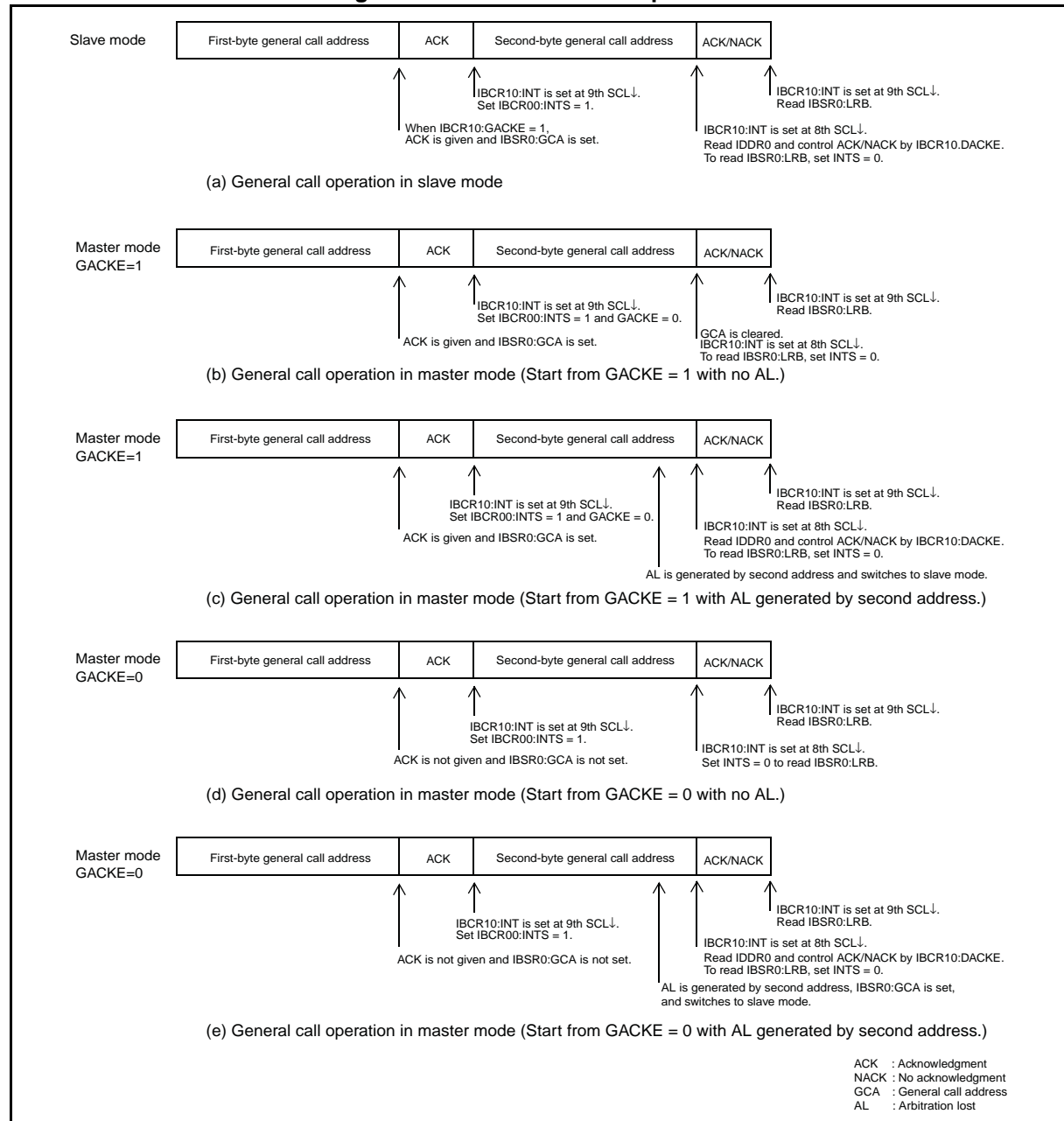
If this module sends a general call address at the same time as another device, you can determine whether the module successfully seized control of the bus by checking whether arbitration lost was detected when the second address byte was transferred. If arbitration lost was detected, the module goes to slave mode and continues to receive data from the master.

MB95330H Series

■ 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 27.7-3 or Figure 27.7-4, 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 27.7-3 Timing Diagram with No Interrupt Generated with IBCR00:ALF = 1

SCL or SDA pin at "L" level

SCL pin   "L"

SDA pin   "L"

                                                                                  1

I^{2}C 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
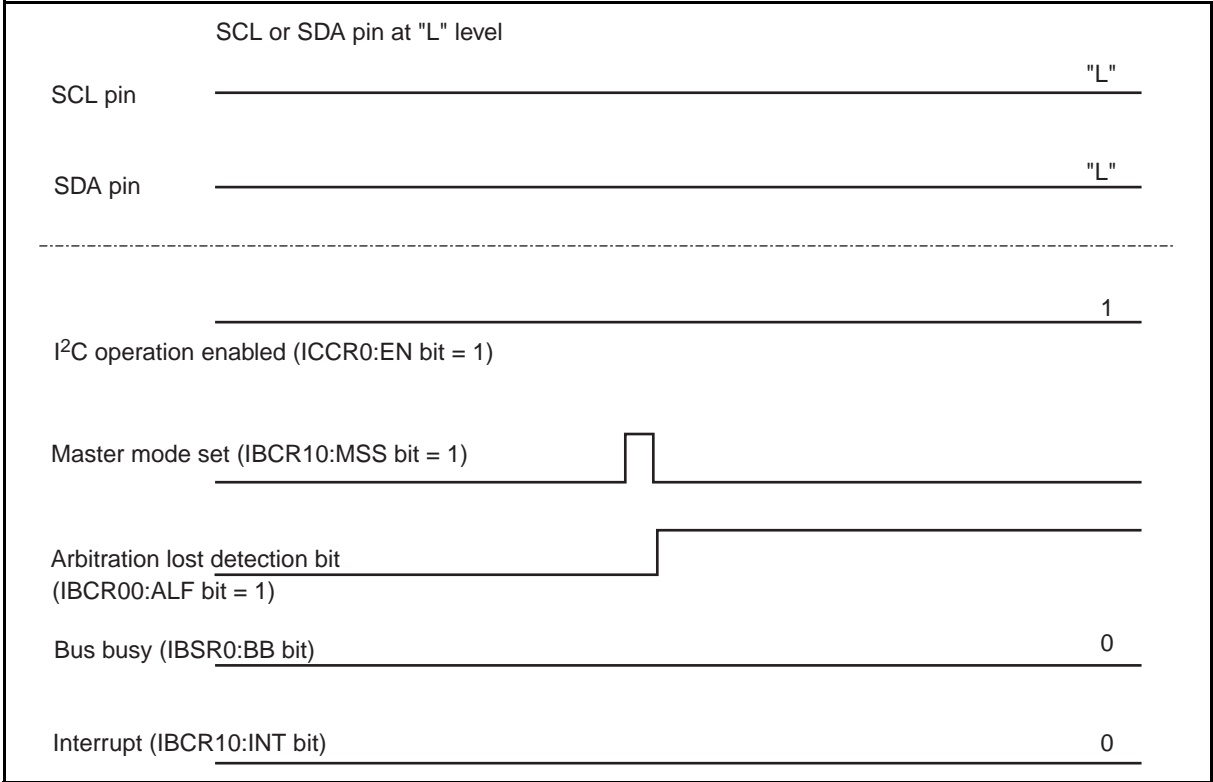
- Conditions (2) in which no interrupt is generated due to arbitration lost

If the program enables I²C operation (by setting the ICCR0:EN bit to "1") and triggers a start condition (by setting the IBCR10:MSS bit to "1") when the I²C bus is in use by another master.

This is because, as shown in Figure 27.7-4, this I²C module cannot detect the start condition (IBSR0:BB bit= 0) if another master starts communications on the I²C bus when the operation of this I²C module has been disabled (ICCR0:EN bit = 0).

Figure 27.7-4 Timing Diagram with No Interrupt Generated with IBCR00:ALF = 1
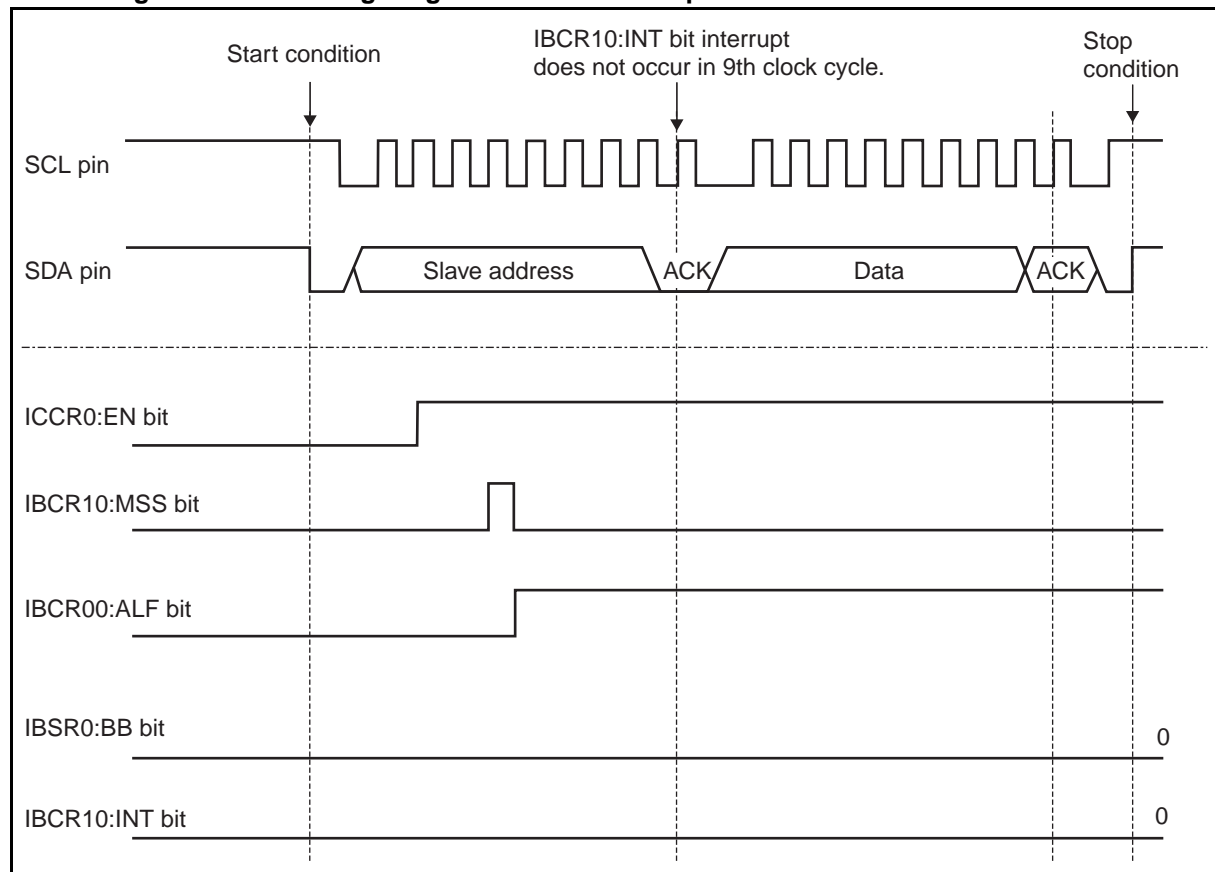
If this situation can occur, use the following procedure to set up the module from the software.

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 27.7-5 Sample Flow Chart 1
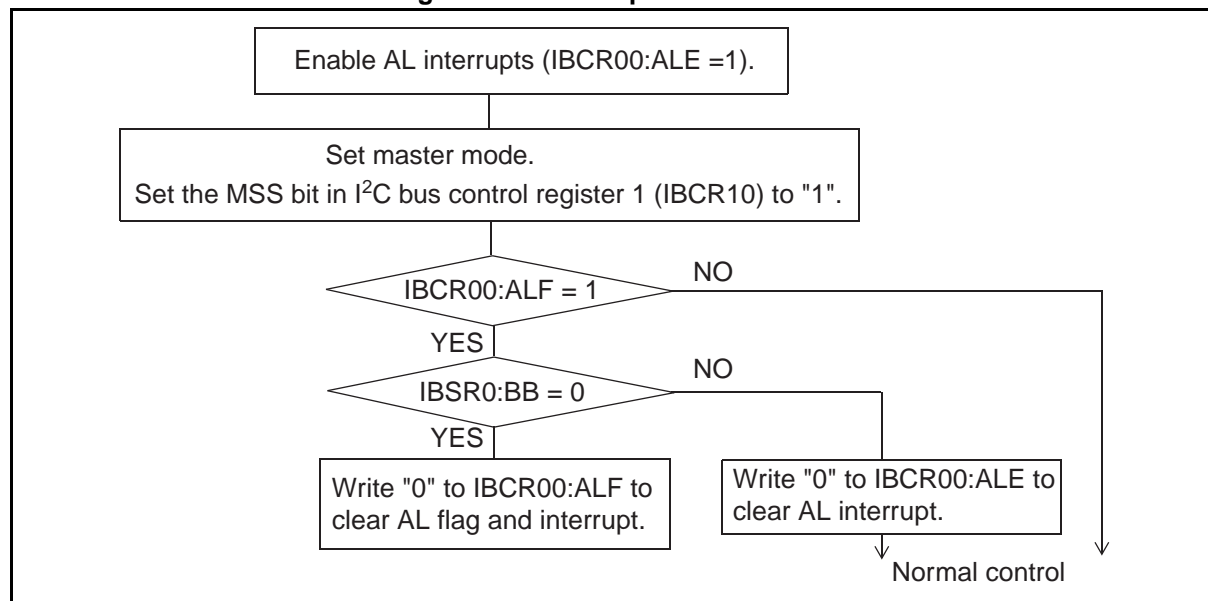
● Example of generating an interrupt (IBCR10:INT bit = 1) with "IBCR00:ALF bit = 1" detected

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 27.7-6 Timing Diagram with Interrupt Generated with "IBCR00:ALF Bit = 1" Detected

MB95330H Series
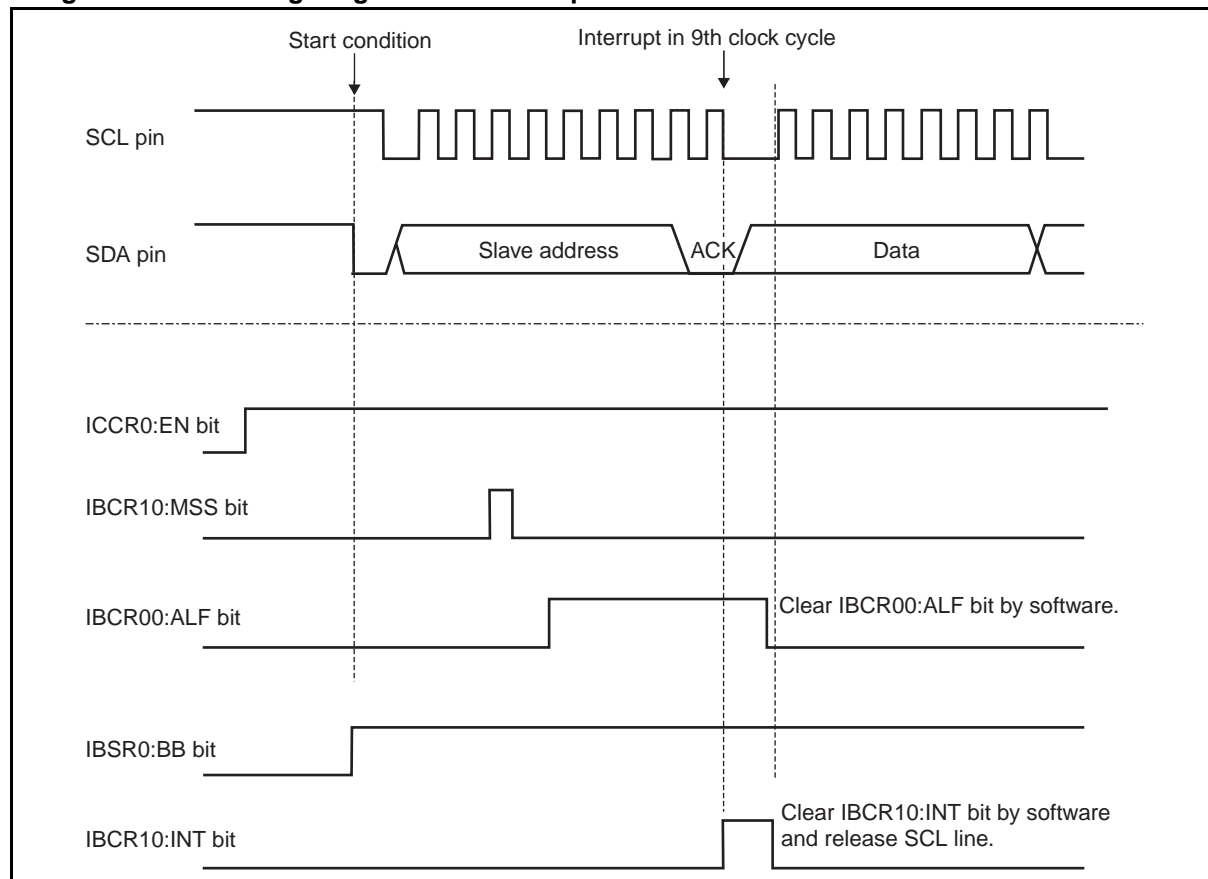
27.7.2 Function to Wake up the MCU from Standby Mode

The wakeup function enables the I^{2}C macro to be accessed while the MCU is in stop or watch mode.

■ Function to Wake Up the MCU from Standby Mode

The I^{2}C 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 I^{2}C 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 I^{2}C operation can restart as soon as possible.
- The wakeup function only applies to the MCU stop and watch modes.

Figure 27.7-7 Comparison of Normal I^{2}C 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 I^{2}C 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 I^{2}C transmission (falling edge detection of SDA).

The following sample flow chart illustrates the wakeup function.

Figure 27.7-8 Sample Flow Chart 2

MB95330H Series
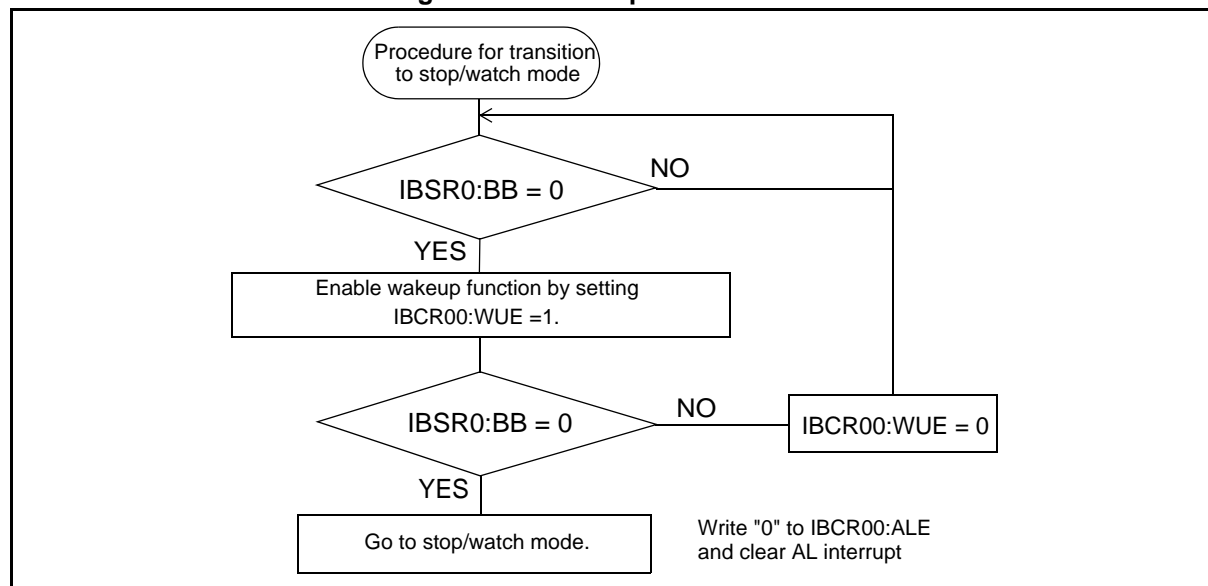
27.8 Notes on Using I^{2}C

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 27.5-4.
- "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 I²C 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 I²C transmission (upon detection of the falling edge of SDA).

- During a MCU standby mode, the status flags, state machine, and I²C bus outputs for the I²C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I²C 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 I²C 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 I²C wakeup function or the wakeup function for some other resource (such as an external interrupt).

MB95330H Series

27.9   Sample Settings for I²C

This section provides sample settings for the I²C interface.

■ Sample Settings

● Enabling/disabling I²C operation

Use the I²C operation enable bit (ICCR0:EN).

Operation	I²C operation enable bit (EN)
To disable I²C operation	Set the bit to "0".
To enable I²C operation	Set the bit to "1".

● Selecting the I²C 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 I²C 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 I²C 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 I²C 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 I²C communication

Use the start condition generation bit (IBCR10:SCC).

Operation	Start condition generation bit (SCC)
To restart communication	Set the bit to "1".

MB95330H Series

● 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 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

The dual operation Flash memory is located at $B000_H$ to $BFFF_H$ and at $F000_H$ to $FFFF_H$, or at $B000_H$ to $BFFF_H$ and at $E000_H$ to $FFFF_H$, or at $B000_H$ to $FFFF_H$ on the CPU memory map. The Flash memory interface circuit enables read access and write access from the CPU to the Flash memory.
dual operation Flash consists of an upper bank (16/8/4 Kbyte × 1)* and a lower bank (2 Kbyte × 2). Unlike conventional Flash products, writing/erasing data to/from one bank and reading data from another bank can be executed simultaneously.
* 16 Kbyte × 1(MB95F334H/F334K)
 8 Kbyte × 1(MB95F333H/F333K)
 4 Kbyte × 1(MB95F332H/F332K)

■ 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 type of 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	Write/sector erase
Write/sector erase	Read
Chip erase	

While data is being written to or erased from one bank, writing data to or sector-erasing data from another bank cannot be executed.

■ Features of Dual Operation Flash Memory

- Sector configuration: 8 Kbyte × 8 bits (2 Kbyte × 2 + 4 Kbyte) /
  12 Kbyte × 8 bits (2 Kbyte × 2 + 8 Kbyte) / 20 Kbyte × 8 bits (2 Kbyte × 2 + 16 Kbyte)
- 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

MB95330H Series

- 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 protect RAM data against power interruption.

28.2 Sector/Bank Configuration of Dual Operation Flash Memory

This section shows the sector/bank configuration of the dual operation 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 dual operation Flash memory is SA0 and SA1 and the upper bank SA2.

Figure 28.2-1 Sector/Bank Configuration of Dual Operation Flash Memory

MB95330H Series

28.3   Registers for Dual Operation Flash Memory

This section shows the registers for the dual operation Flash memory.

■ Registers for Dual Operation Flash Memory

Figure 28.3-1 Registers for Dual Operation 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 (RMW) type of 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".

MB95330H 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 2 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 2 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 $F800_H$-$FFFF_H$. **Writing "1":** maps the 2 Kbyte address area of SA2 to $B800_H$-$BFFF_H$, and SA1 to $F800_H$-$FFFF_H$.

MB95330H Series

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 an SAxE bit in SWRE0 enables writing data into the sector corresponding to that bit. Writing "0" to an SAxE 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 re-write the data, execute a reset operation.

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)

Bit name		Function
bit7 to bit3	Reserved bits	Always set these bits to "0".
bit2 to bit0	SA2E to SA0E: Writing function setup bits	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 below

Table of writing function setup bits and their corresponding Flash memory sectors

Bit Name	Corresponding Sector in Flash Memory
SA2E	SA2
SA1E	SA1
SA0E	SA0

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 into 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 into a sector cannot be enabled even though the SAxE bit corresponding to that sector is set to "1".

**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)**
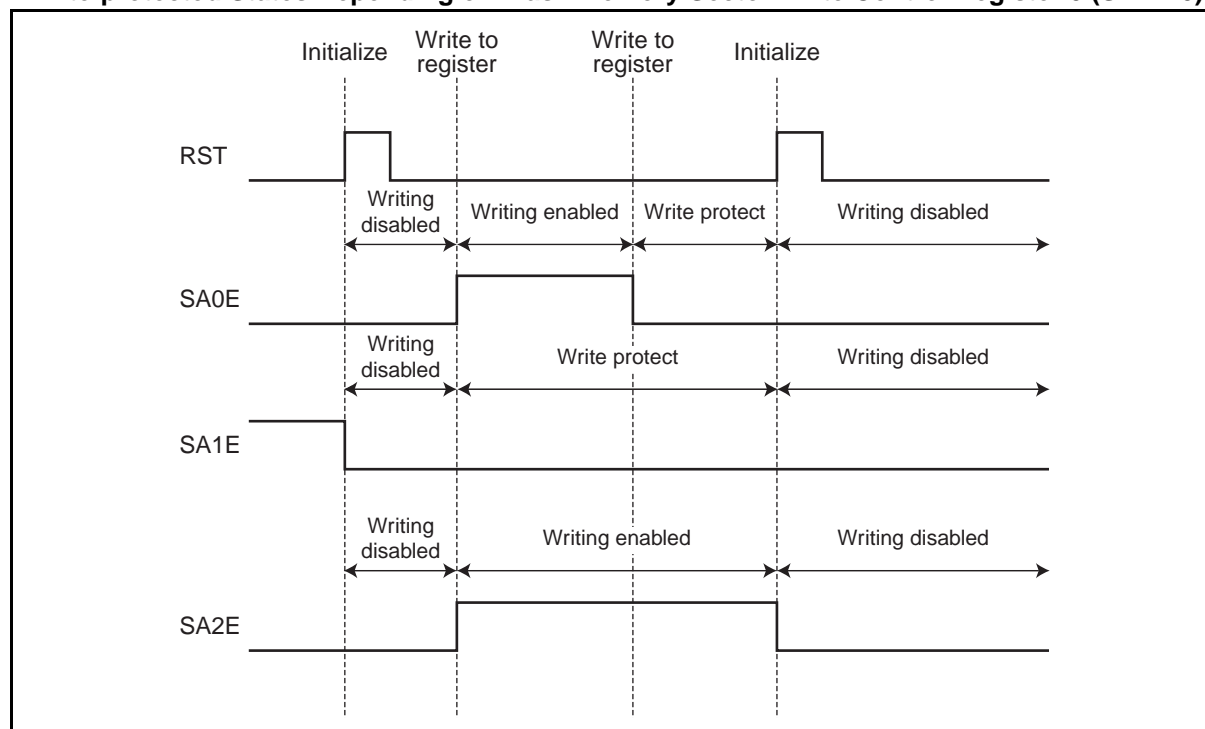
Writing disabled:

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".

MB95330H 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	0000XXXXв
	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 suspend 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 sector erase suspend 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 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.

■ **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 Writing
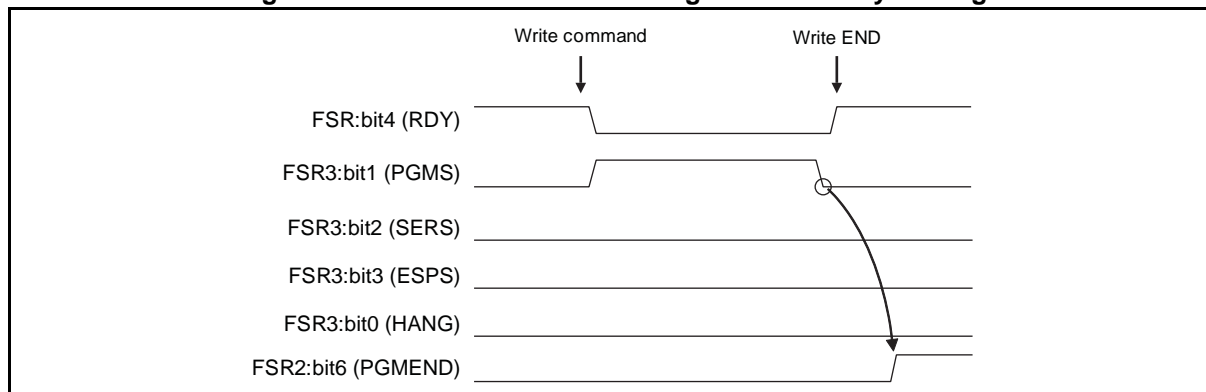
Figure 28.3-9 FSR2:PGMTO when Flash Memory Writing Failed
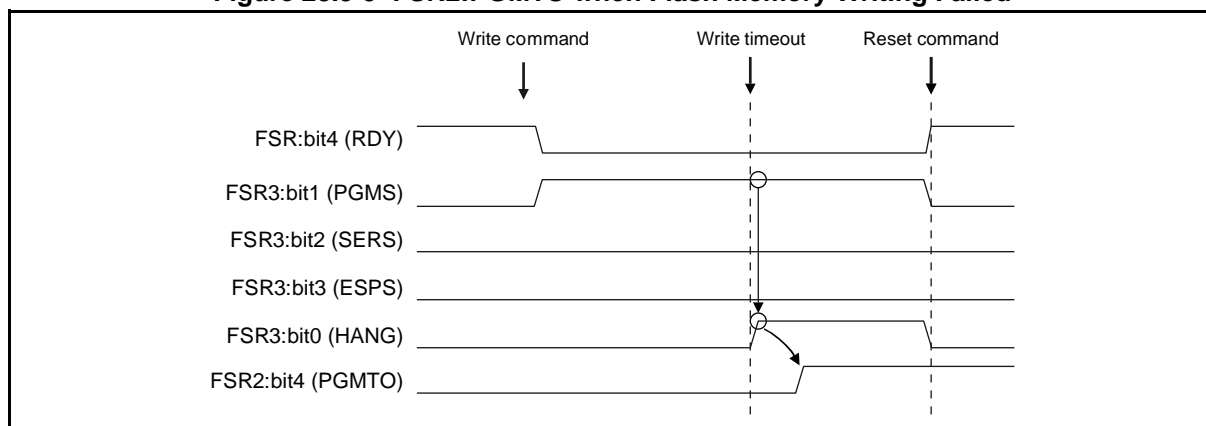
Figure 28.3-10 FSR2:ERSEND during Flash Memory Sector Erase
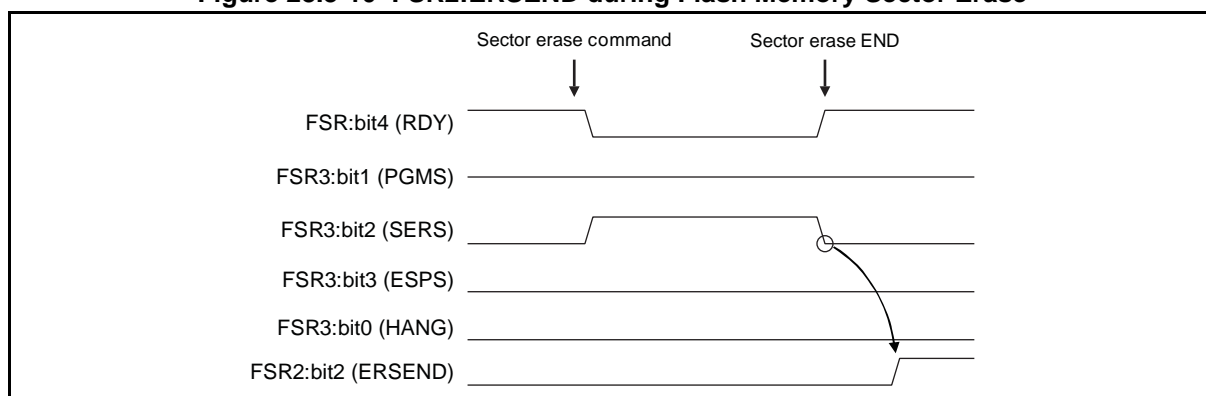
Figure 28.3-11 FSR2:ERSTO when Flash Memory Sector Erase Failed
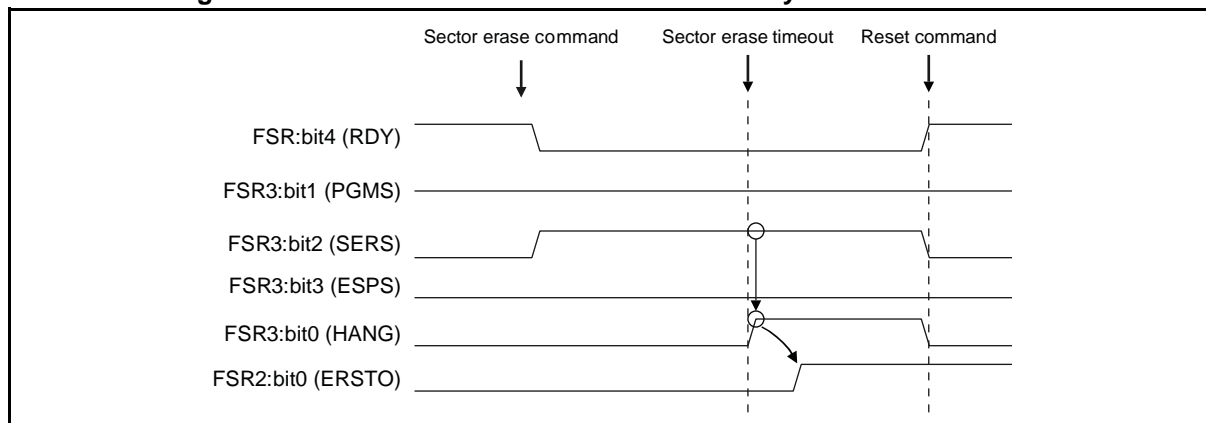
Figure 28.3-12 FSR2:PGMEND and FSR2:ERSEND when Flash Memory Writing Is in Progress with Flash Memory Sector Erase Suspended
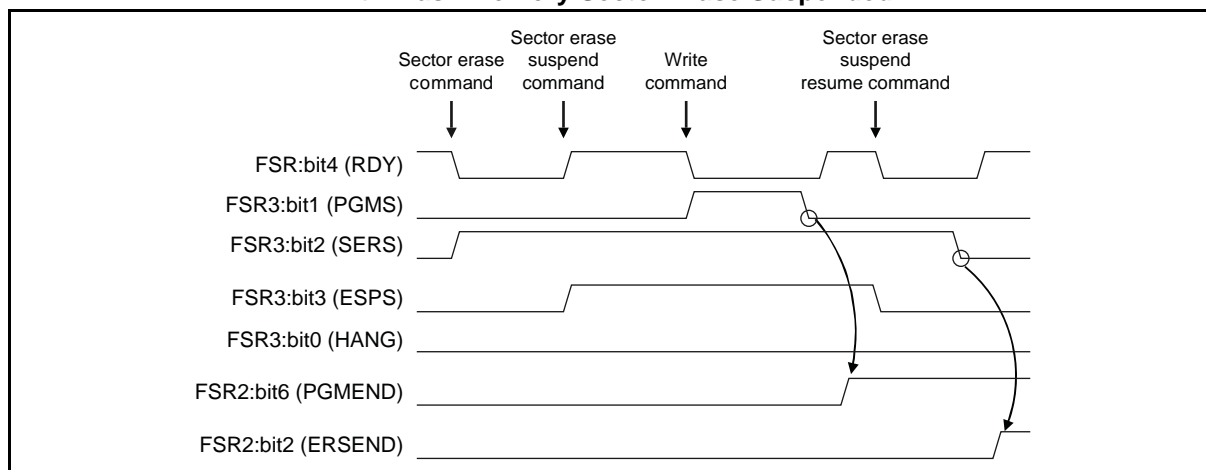
Figure 28.3-13 FSR2:PGMTO and FSR2:ERSEND when Flash Memory Writing Failed with Flash Memory Sector Erase Suspended
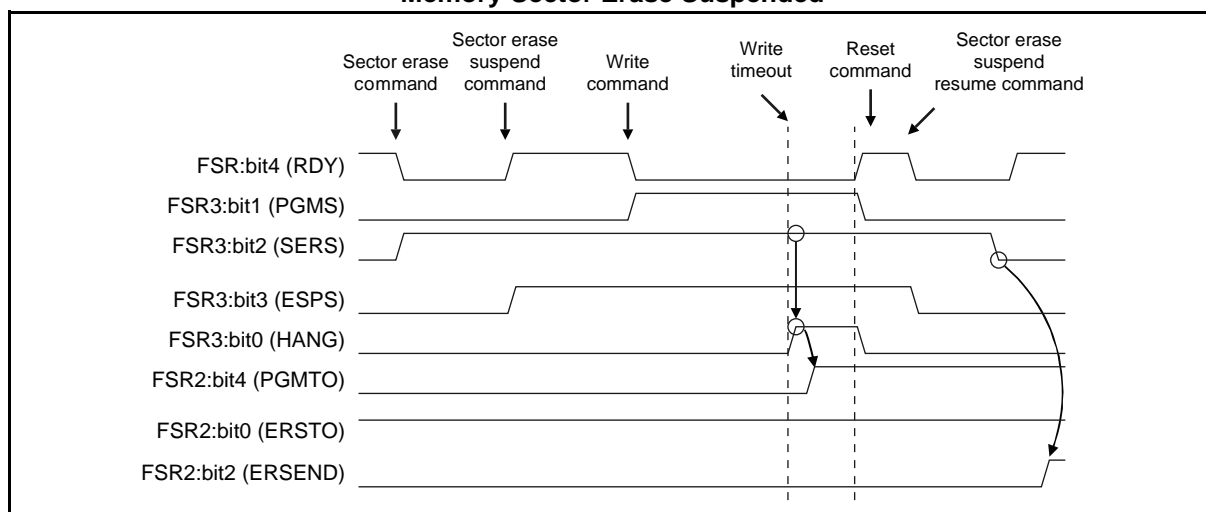
Figure 28.3-14 FSR2:ERSEND during Flash Memory Sector Erase
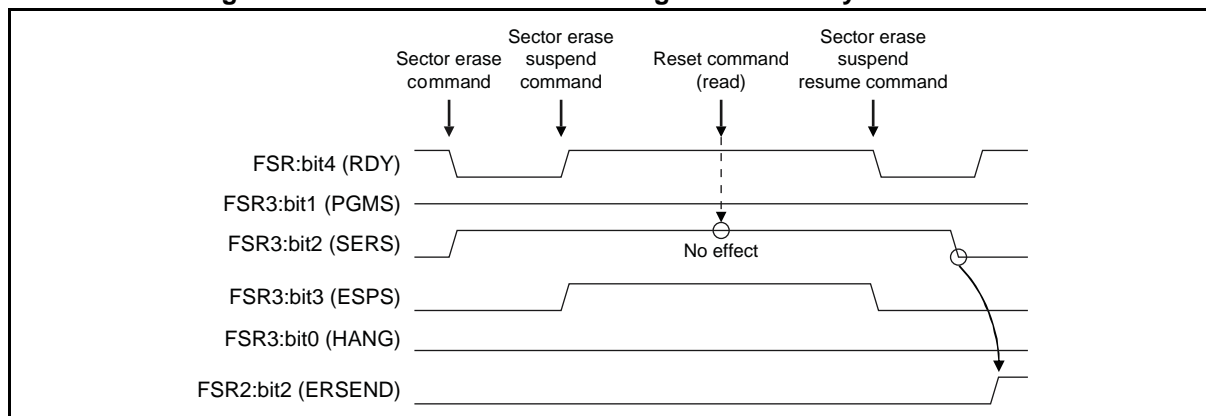
Figure 28.3-15 FSR2:PGMEND and FSR2:ERSTO when Flash Memory Sector Erase Failed after Sector Erase Has Resumed
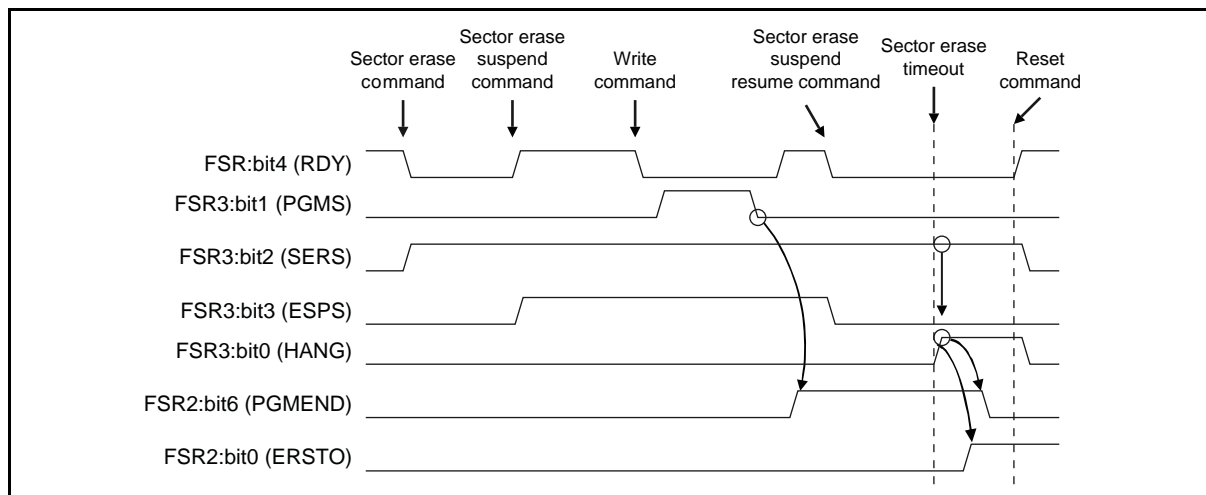
■ Flash Memory Sector Write Control Register 0 (SWRE0) Setup Flow Chart

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

MB95330H Series
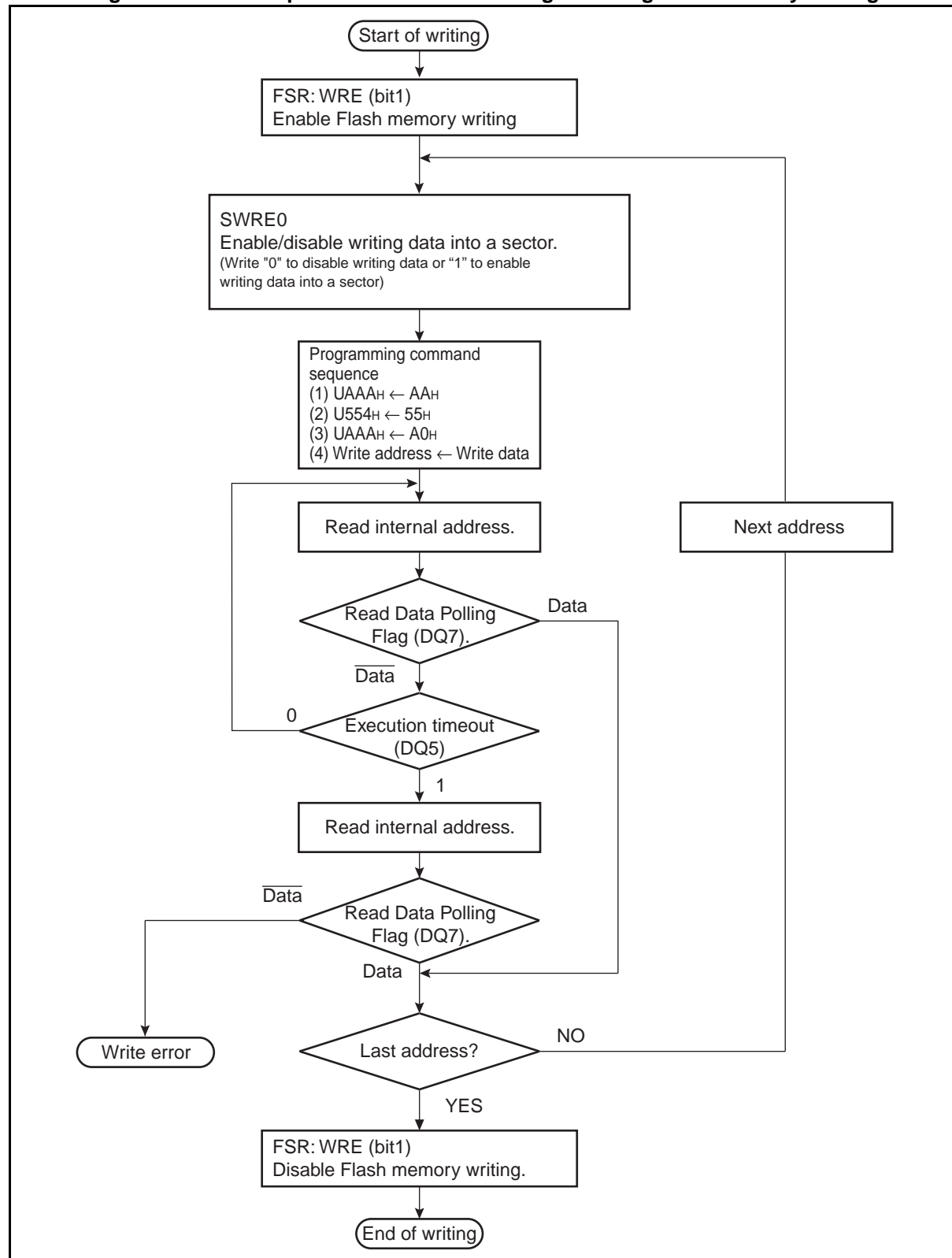
■ Note on Setting (FSR:WRE)

To write data to the Flash memory, set FSR:WRE to "1" to enable Flash memory writing and then set the SWRE0 register. 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.

Notes:

- Addresses in the table above are values on the CPU memory map. All addresses and data are in hexadecimal notation. However, "X" is an arbitrary value.
- "U" in an address in the table above is not arbitrary, but represents the upper four bits (bit15 to bit12) 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).

MB95330H Series

■ 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 (bit15 to bit12) 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, it is necessary to initialize the command sequencer in the Flash memory by executing a reset.

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.

MB95330H Series

● 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)	$\overline{DQ7} \rightarrow$ DATA: 7	Toggle → DATA: 6	$0 \rightarrow$ DATA: 5	$0 \rightarrow$ DATA: 3
	Chip/sector erasing → Erasing completed	$0 \rightarrow 1$	Toggle → Stop	$0 \rightarrow 1$	1
	Sector erasing wait → Erasing started	0	Toggle	0	$0 \rightarrow 1$
	Erasing → Sector erasing suspended (Sector being erased)	$0 \rightarrow 1$	Toggle → 1	0	$1 \rightarrow 0$
	Sector erasing suspended → Erasing resumed (Sector being erased)	$1 \rightarrow 0$	$1 \rightarrow$ Toggle	0	$0 \rightarrow 1$
	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	$0 \to 1$	0	$0 \to 1$	$1 \to 0$	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 Section 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.

MB95330H Series

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 timeout 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 indicates that writing has failed if the automatic algorithm is still running for the data polling or toggle bit function.

If an attempt is made to write "1" to a Flash memory address holding "0", for example, the Flash memory is locked, preventing the automatic algorithm from being terminated and valid data from being output from the data polling flag (DQ7). As the toggle bit flag (DQ6) does not stop toggling, the automatic algorithm does not terminate, the execution time of the automatic algorithm exceeds a specified time limit and the execution timeout flag (DQ5) outputs "1". The state in which the execution timeout flag (DQ5) outputs "1" means that the Flash memory has not been used correctly; but not 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	$0 \rightarrow$ DATA: 3	1	$0 \rightarrow 1$	$1 \rightarrow 0$	$0 \rightarrow 1$	DATA: 3

Table 28.5-10 State 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.

MB95330H Series

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.

■ 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.

MB95330H Series

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

```
                        ( Start of writing )
                               │
                               ▼
              ┌────────────────────────────────────┐
              │ FSR: WRE (bit1)                    │
              │ Enable Flash memory writing.       │
              └────────────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────────────┐
              │ SWRE0                               │
              │ Enable/disable writing data to a sector. │
              │ (Write "0" to disable writing data or "1" to enable │
              │ writing data io a sector.)          │
              └────────────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────────────┐
              │ Programming command                 │
              │ sequence                            │
              │ (1) UAAAH ← AAH                     │
              │ (2) U554H ← 55H                     │
              │ (3) UAAAH ← A0H                     │
              │ (4) Write address ← Write data      │
              └────────────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────────────┐
              │ Read internal address.             │         ┌──────────────────┐
              └────────────────────────────────────┘         │  Next address    │
                               │                             └──────────────────┘
                               ▼
                         Data polling ──── Data ────┐
                           (DQ7)                     │
                               │ Data               │
                               ▼                     │
                    0 ── Execution timeout           │
                           (DQ5)                     │
                               │ 1                   │
                               ▼                     │
              ┌────────────────────────────────────┐│
              │ Read internal address.             ││
              └────────────────────────────────────┘│
                               │                     │
            Data               ▼                     │
          ┌──────── Data polling                     │
          │          (DQ7)                           │
          │            │ Data ◄────────────────────── ┘
          ▼            ▼
   ( Write error )   Last address? ──── NO ──────┐
                         │ YES                    │
                         ▼
              ┌────────────────────────────────────┐
              │ FSR: WRE (bit1)                    │
              │ Disable Flash memory writing.      │
              └────────────────────────────────────┘
                               │
                               ▼
                        ( End of writing )
```

MB95330H Series

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.

MB95330H Series

Figure 28.6-2 Sample Procedure for Erasing Data from Sectors in Flash Memory
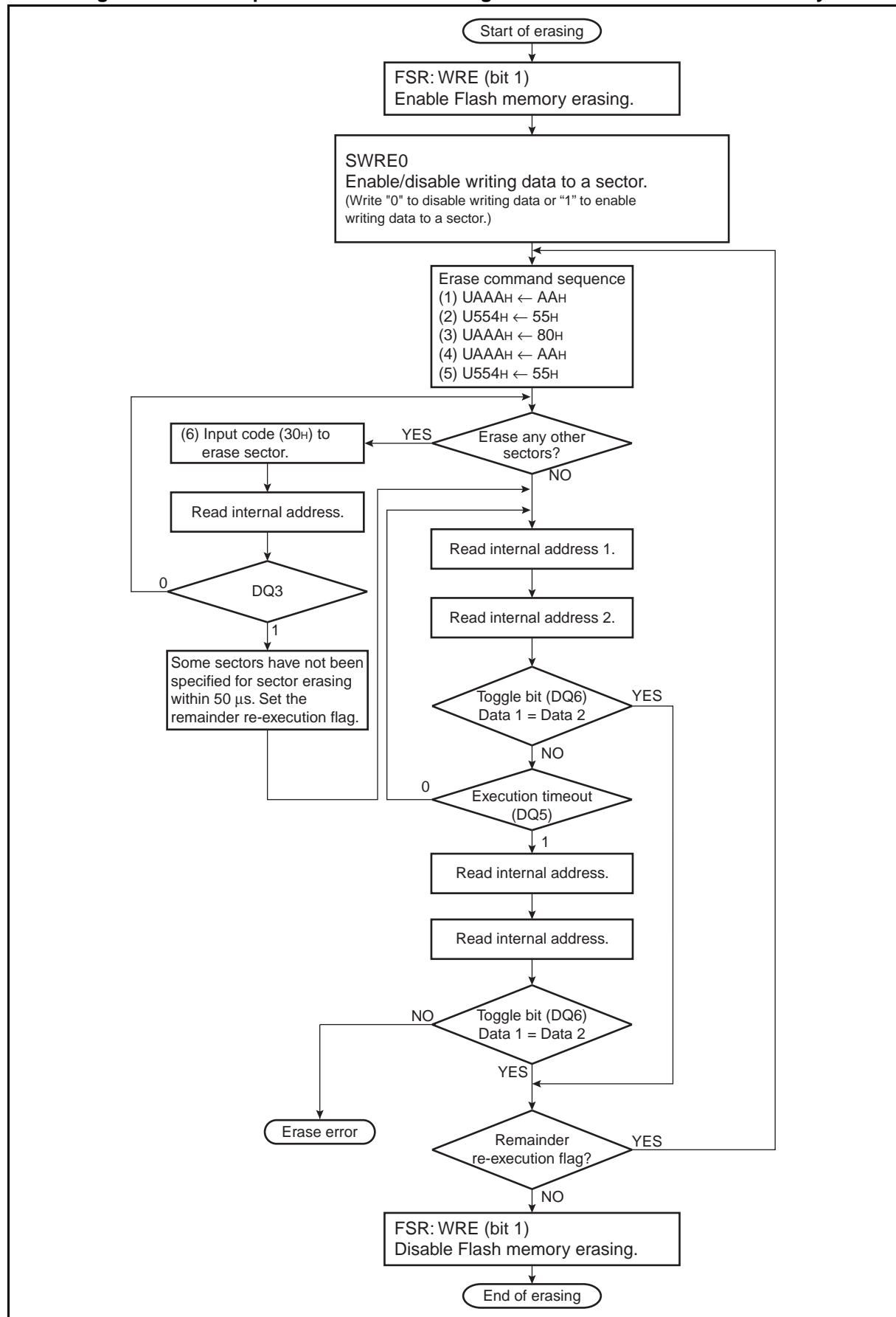
28.6.5 Suspending Sector Erasing from Flash Memory

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 during 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 the 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 the 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.

Note:

Keep an interval of 20 ms between issuing a sector erase command or a sector erase resume command and issuing a sector erase suspend command.

MB95330H Series

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 Memory

Pay attention in particular to the following points when using the dual operation Flash:
- **Interrupt generated when the upper bank is updated**
- **Procedure for 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/written 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 an 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 for Setting the Sector Swap Enable Bit (FSR:SSEN)
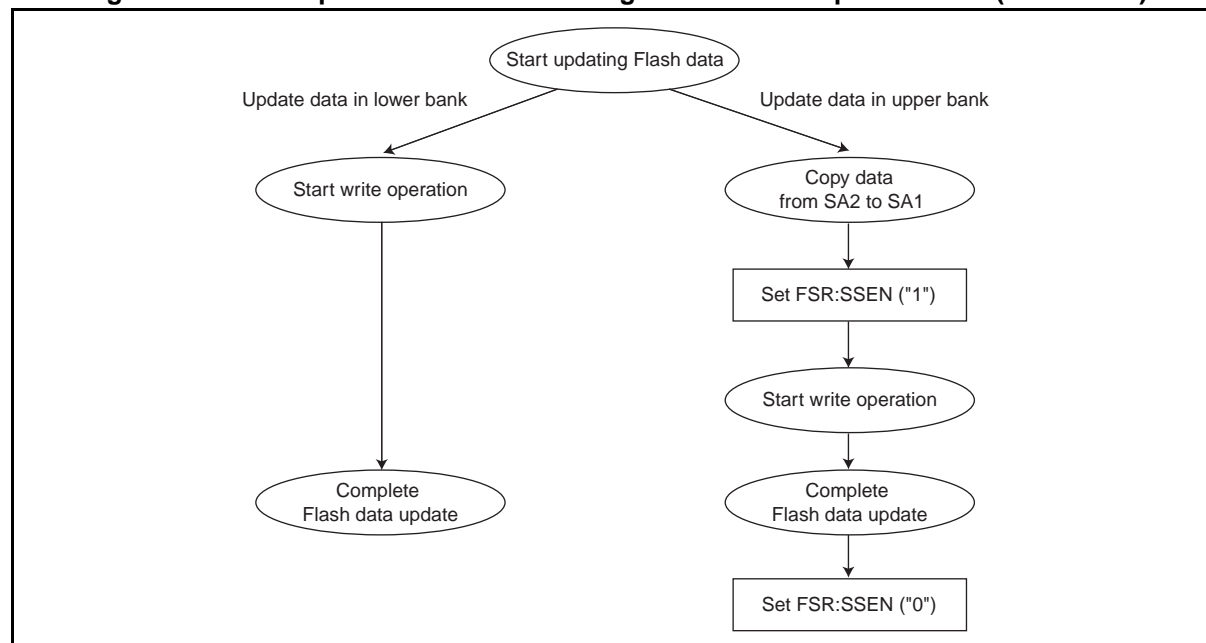
■ **Operation during Writing/Erasing**

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, making 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 making state transition.

■ **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$_H$

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

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.

MB95330H Series

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 provides an example of serial programming connection.

29.1 Basic Configuration of Serial Programming Connection

29.2 Example of Serial Programming Connection

29.1 Basic Configuration of Serial Programming Connection

The MB95330H Series supports Flash memory serial onboard programming. This section describes the configuration.

■ Basic Configuration of Serial Programming Connection

The BGM adaptor MB2146-08-E, manufactured by Fujitsu Semiconductor Limited, 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
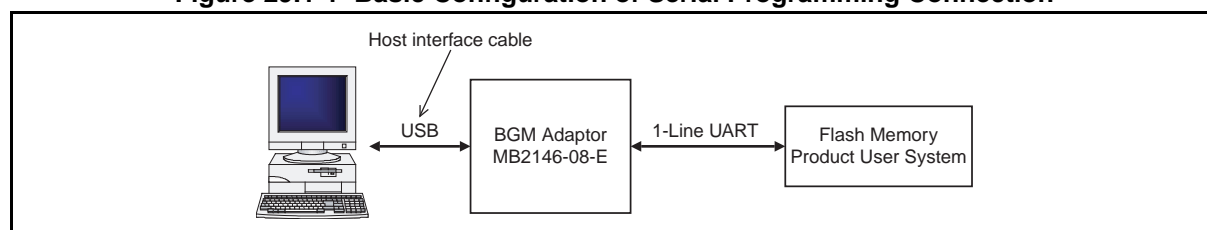
Table 29.1-1 Pins Used for Fujitsu Semiconductor Standard Serial Onboard Programming

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 BGM adapter MB2146-08-E.
C	Capacitor connection	Connect it to a bypass capacitor and then to the ground.
$\overline{RST}$	Reset	The $\overline{RST}$ 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 29.2-2.)

● 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.

■ Entry of MCU into 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

MB95330H Series
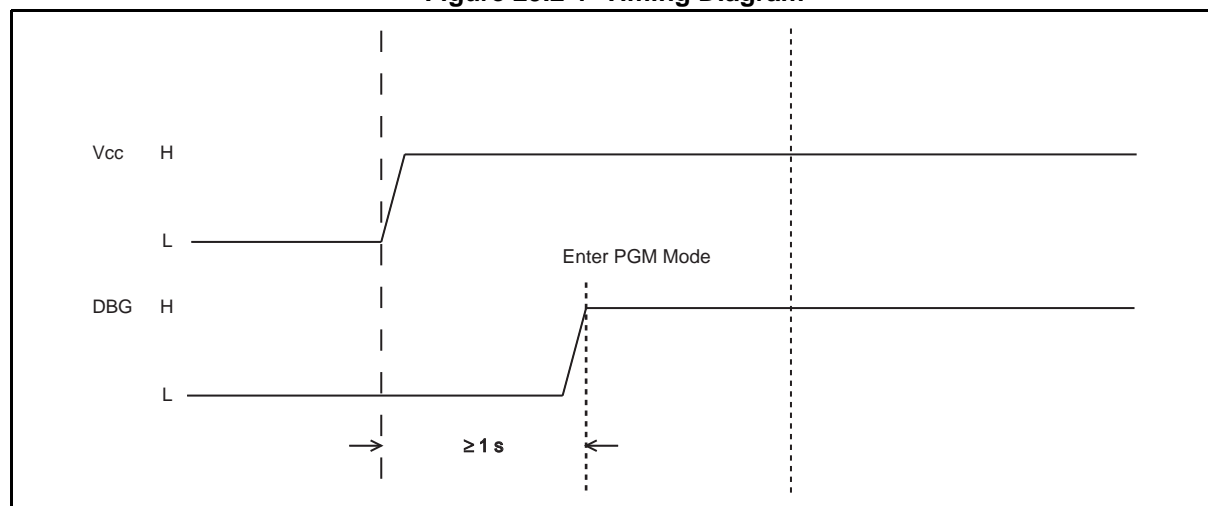
■ Example of Serial Programming Connection

Figure 29.2-2 shows an example of connection for serial writing.

The power is supplied from the programmer through the V_{CC} pin to the adaptor.

Figure 29.2-2 Example of Serial Programming Connection
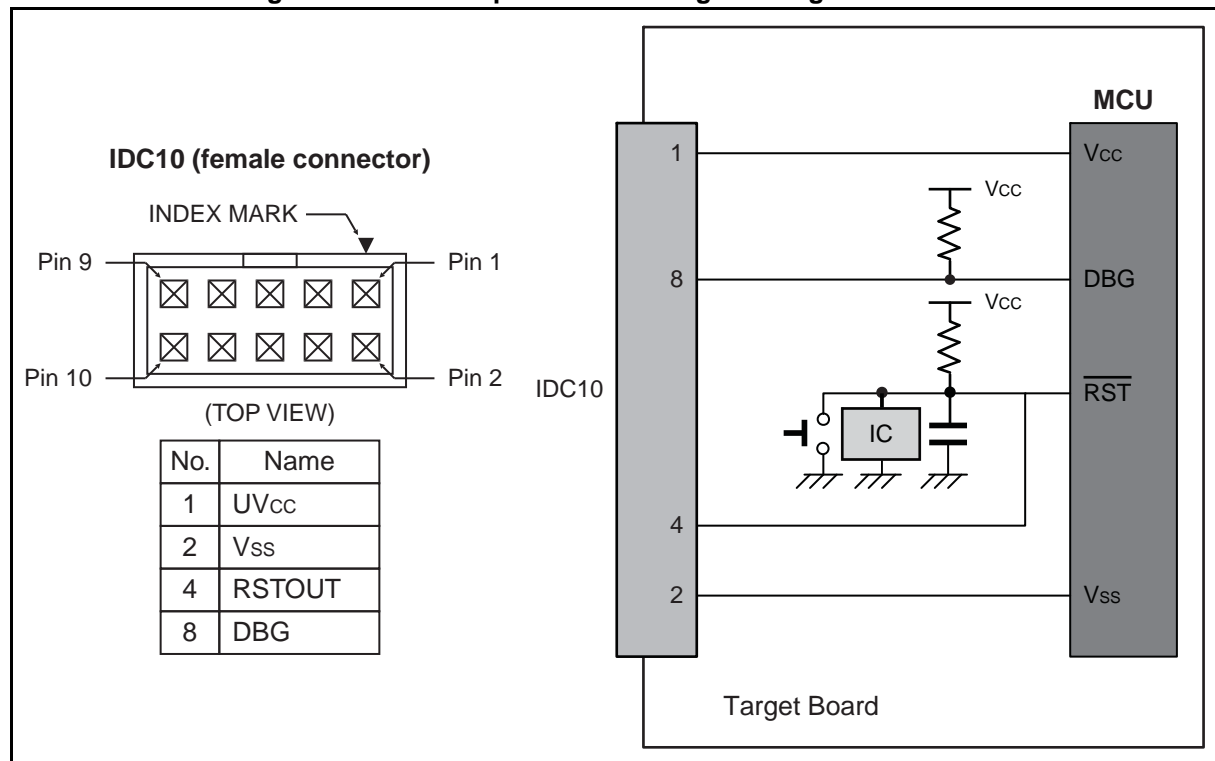
CHAPTER 30
NON-VOLATILE REGISTER (NVR) FUNCTION

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 I/O area. In the MB95330H 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 I/O 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 basic configuration of serial programming connection for the Flash memory products.

Figure 30.1-1 Retrieval of NVR during Reset

NVR Interface (I/O Area)	Address	Value
	0FE4$_H$	X1110101$_B$
	0FE5$_H$	XX101010$_B$
	0FEB$_H$	10100101$_B$
	0FEC$_H$	10010110$_B$
NVR (Flash Area)	FFBC$_H$	X1110101$_B$
	FFBD$_H$	XX101010$_B$
	FFBE$_H$	10100101$_B$
	FFBF$_H$	10010110$_B$

Memory Map
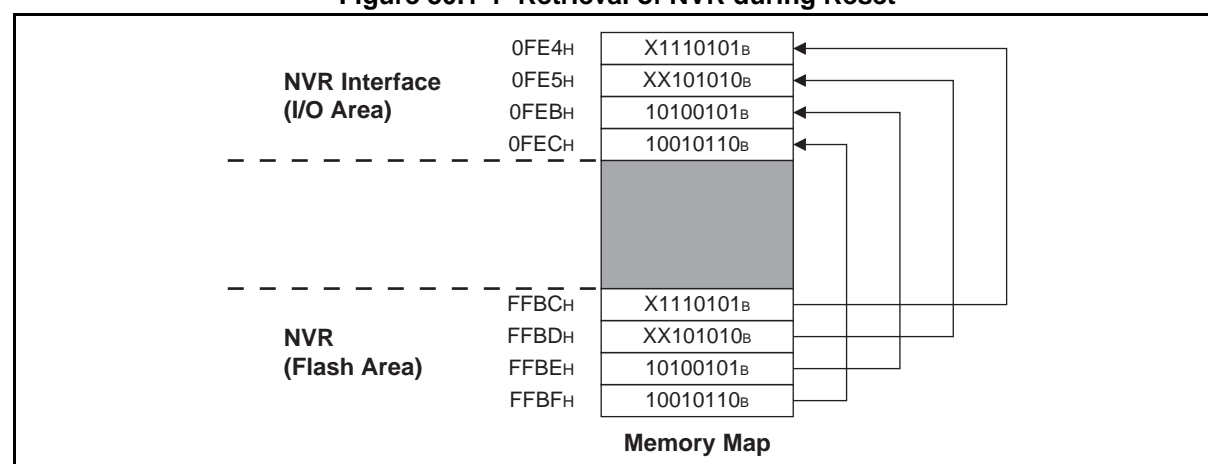
MB95330H Series

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
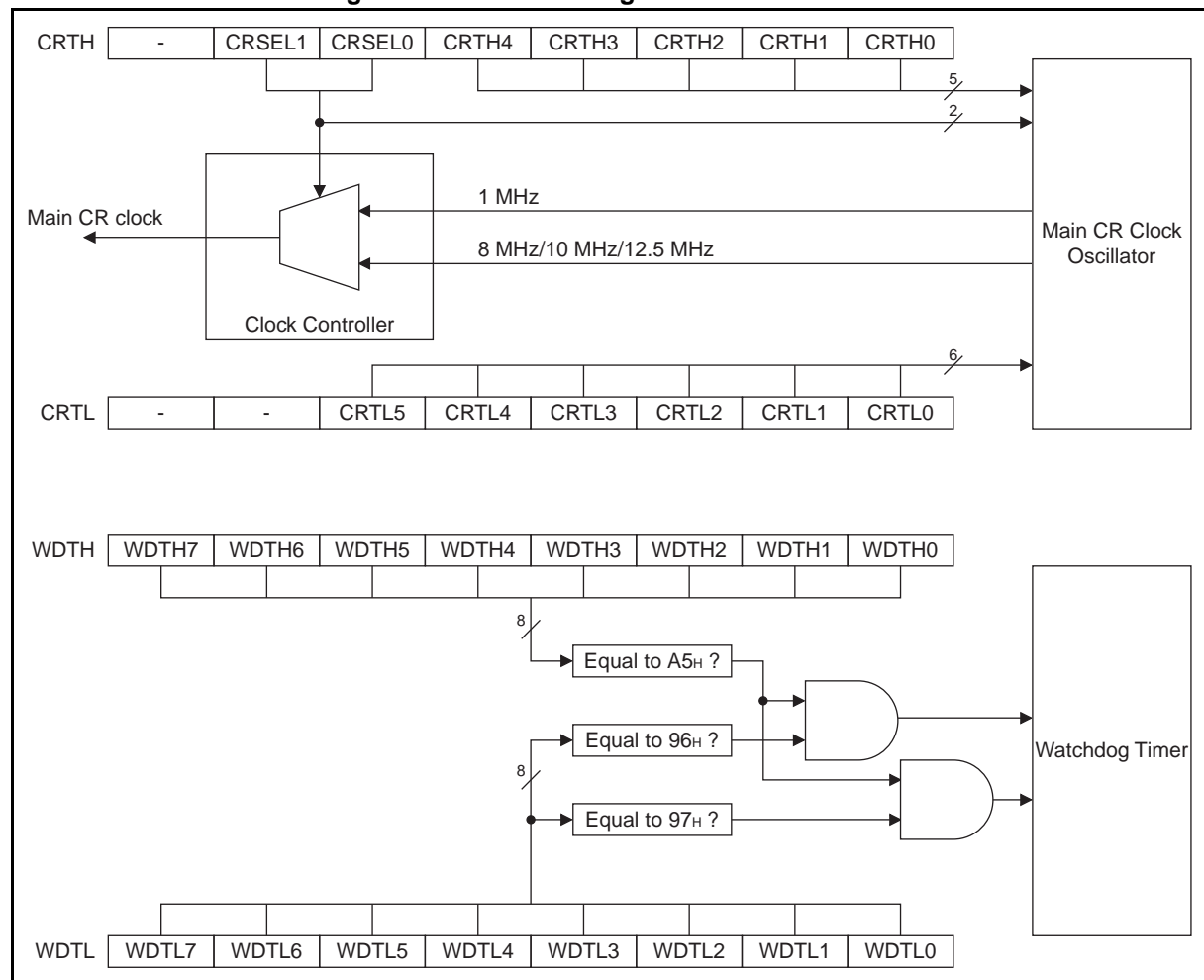
30.3   Registers of NVR Interface

This section lists the registers of the NVR interface.

■ Registers of NVR Interface

Figure 30.3-1 Registers of NVR Interface

	Address	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	

	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	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WDTH	0FEBH	WDTH7	WDTH6	WDTH5	WDTH4	WDTH3	WDTH2	WDTH1	WDTH0	XXXXXXXXB
		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	0FECH	WDTL7	WDTL6	WDTL5	WDTL4	WDTL3	WDTL2	WDTL1	WDTL0	XXXXXXXXB
		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

MB95330H Series

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)

Bit name		Function
bit7	Undefined bit	The read value is always "0". Writing a value to it has no effect on operation.
bit6, bit5	CRSEL1, CRSEL0: Main CR clock frequency select bits	These two bits are loaded from the Flash address $FFBC_H$ (bit6 and 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[1:0]. | CRSEL[1:0]	Main CR clock frequency | |---|---| | 00_B | 1 MHz | | 01_B | 12.5 MHz | | 10_B | 10 MHz | | 11_B | 8 MHz | See Section 30.5 "Notes on Using NVR" for notes on changing the main CR frequency selection.
bit4 to bit0	CRTH4 to CRTH0: Main CR coarse trimming bits	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. | CRTH [4:0]	Main CR clock frequency | |---|---| | 00000_B | Highest | | : | : | | 11111_B | Lowest | See Section 30.4 "Notes on Main CR Clock Trimming" and Section 30.5 "Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.

MB95330H Series

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)
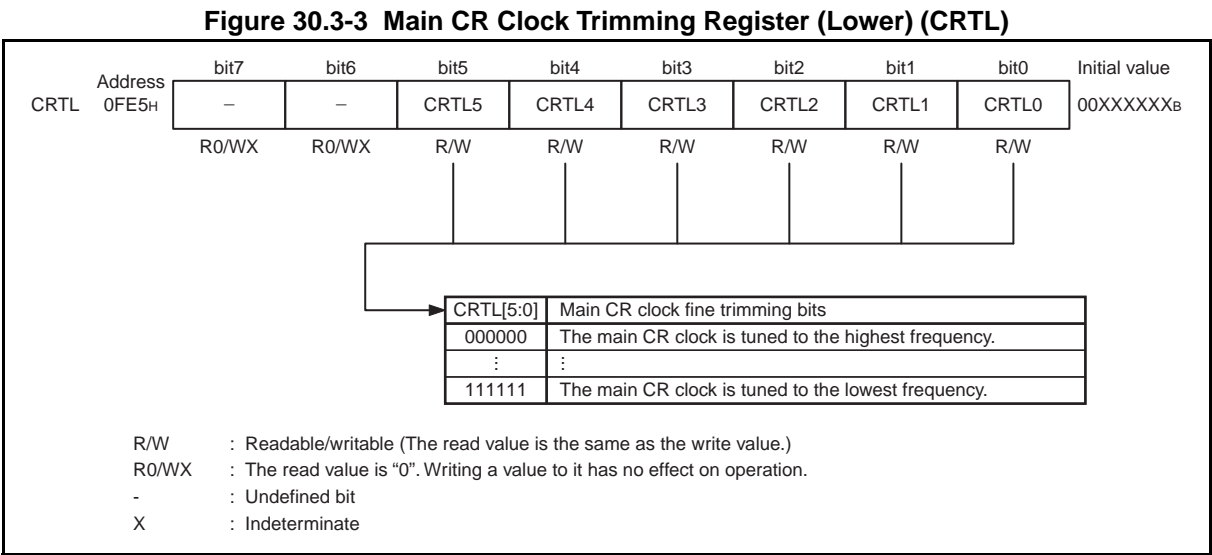
Table 30.3-2 Functions of Bits in CR Trimming Register (Lower) (CRTL)

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	CRTL5 to CRTL0: 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. ・<table><tr><td>CRTL [5:0]</td><td>Main CR clock frequency</td></tr><tr><td>000000$_B$</td><td>Highest</td></tr><tr><td>:</td><td>:</td></tr><tr><td>111111$_B$</td><td>Lowest</td></tr></table> See Section 30.4 "Notes on Main CR Clock Trimming" and Section 30.5 "Notes on Using NVR" for details of main CR clock trimming and notes on changing the main CR clock values respectively.

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)

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).
Others	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
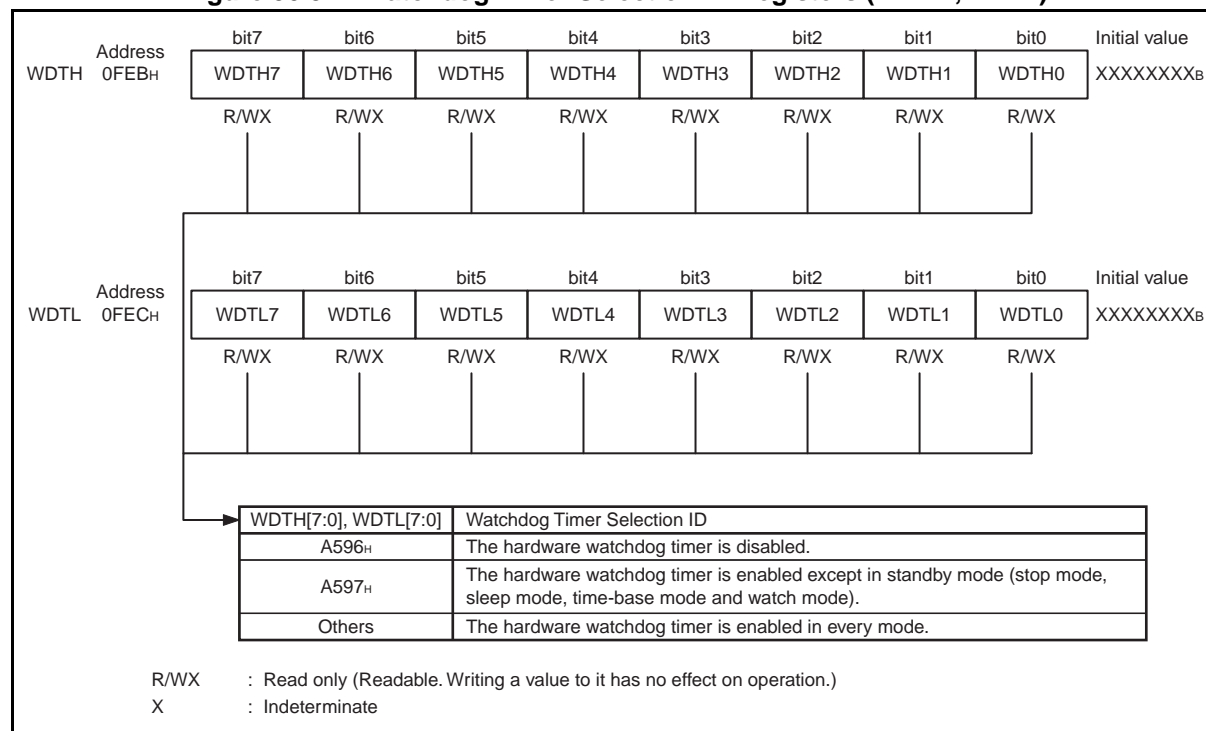
Table 30.3-3 Functions of Bits in Watchdog Timer ID Register (Upper) (WDTH)

Bit name		Function
bit7 to bit0	WDTH7 to WDTH0: Watchdog timer selection ID (upper)	These eight 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 Section 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	WDTL7 to WDTL0: Watchdog timer selection ID (lower)	These eight 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 Section 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 is disabled; the software watchdog timer is enabled.
A597$_H$	The hardware watchdog timer is enabled; the software watchdog timer is disabled. The hardware watchdog timer can be stopped in all standby modes (stop mode, sleep mode, time-base timer mode and watch mode).
Other than the above	The hardware watchdog timer is enabled; the software watchdog timer is disabled. The hardware watchdog timer keeps operating in all standby modes (stop mode, sleep mode, time-base timer mode 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 I/O 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

MB95330H Series

The relationship between coarse trimming step size and CR frequency is illustrated in the diagram below.

Figure 30.4-1 Coarse Trimming Step Size

Note: The size of fine trimming ranges from -6.5% (CRTL[5:0] = 111111$_B$) to +6.5% (CRTL[5:0] = 000000$_B$).
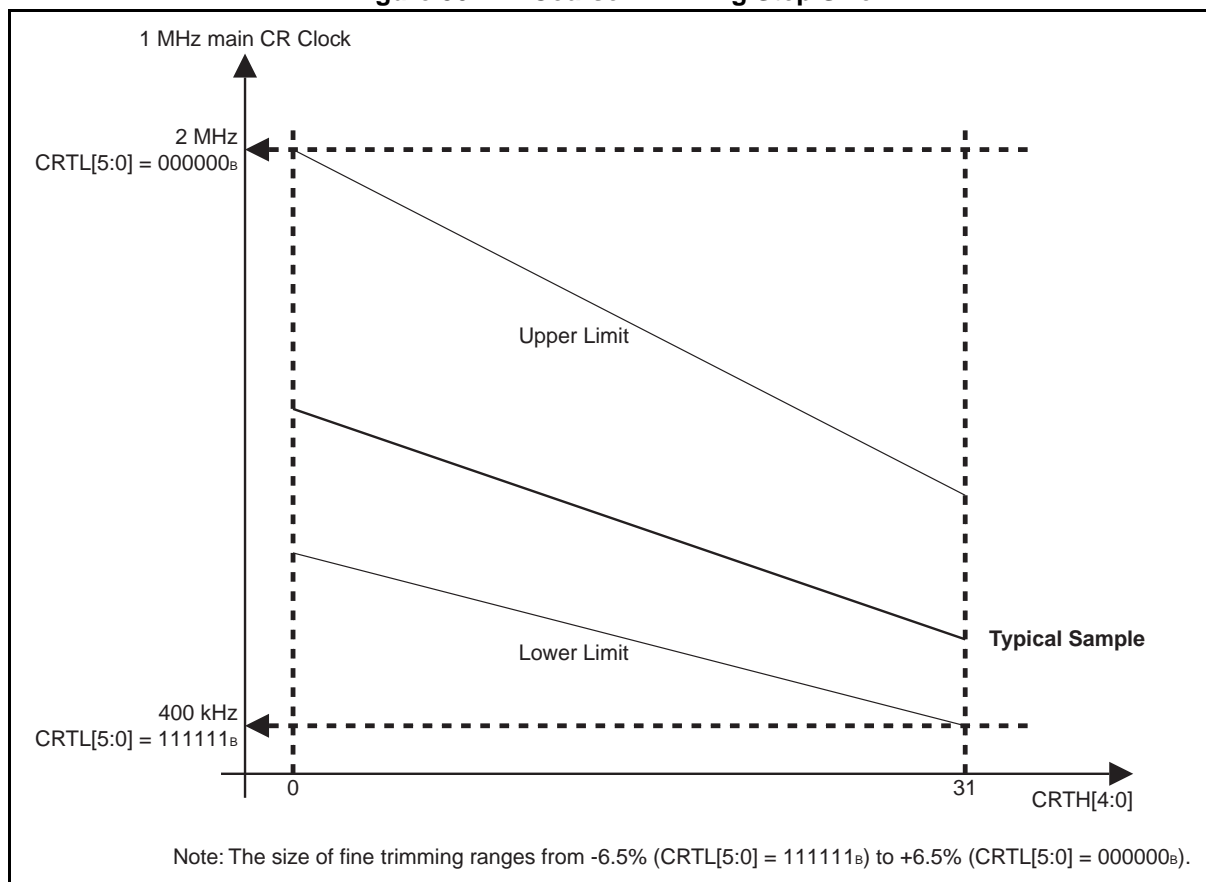
30.5 Notes on Using NVR

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:CRSEL1, CRSEL0. However, unstable oscillation occurs for a certain period of time after the modification of clock frequency has been initiated. To prevent such oscillation, it is strongly recommended that the following actions should be taken. Firstly, switch the CPU clock source from the main CR clock to another clock (main clock / subclock / sub-CR clock), then modify the main CR parameters, and switch back to the main CR clock.

2. Please note that the NVR interface does not program a modified value to the NVR Flash area. If the CRTH and CRTL registers are modified, the modified value is programmed to the NVR Flash area by the Flash writer.

■ 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 Semiconductor 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 System Configuration Register (SYSC)

The controller consists of the SYSC register, which is an 8-bit register used to configure the clock and reset system, and select 8/16-bit PPG output ports.

■ Functions of SYSC

- 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/SNI1 pin and that for the PG2/X1A/SNI2 pin

- Selection of the port/oscillation function for the PF0/X0 pin and that for the PF1/X1 pin

- Selection of the external clock input function for the HCLK1 pin and the HCLK2 pin

- Selection of the EC0 input pin as the external count clock input pin for the 8/16-bit composite timer

- Selection of the 8/16-bit PPG output ports from P10-P11, P13-P16 and P62-P67

MB95330H Series

31.2 System Configuration Register (SYSC)

This section provides details of the SYSC register.

■ System Configuration Register (SYSC)

Figure 31.2-1 System Configuration Register (SYSC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE8H	PGSEL	PFSEL	EXCK1	EXCK0	EC0SL	PPGSEL	RSTOE	RSTEN	11000011B
	R/W	R/W	R/W	R/W	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 $\overline{RST}$/PF2 pin.
1	Enables the reset input function of the $\overline{RST}$/PF2 pin.

RSTOE	Reset output enable/disable bit
0	Disables the reset output function of the $\overline{RST}$/PF2 pin.
1	Enables the reset output function of the $\overline{RST}$/PF2 pin.

PPGSEL	8/16-bit PPG output port select bit
0	Selects P10-P11 and P13-P16 as the 8/16-bit PPG output ports.
1	Selects P62-P67 as the 8/16-bit PPG output ports.

EC0SL	EC0 clock select bit
0	Selects the P12/EC0 pin as the external count clock input pin.
1	Selects the P04/EC0 pin as the external count clock input pin.

EXCK1	EXCK0	HCLK source select bits
0	0	Disables HCLK1/HCLK2 input. (The main oscillation input at the X0/X1 pin is selected as the main oscillation clock input.)
0	1	Selects P04 as the main oscillation clock input pin.
1	0	Selects P05 as the main oscillation clock input pin.
1	1	Setting prohibited

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	PG1/PG0 function select bit
0	Selects the PG1/PG0 pins as subclock oscillator pins.
1	Selects the PG1/PG0 pins as general-purpose I/O ports.

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

[shaded] : Initial value

Table 31.2-1 Functions of Bits in SYSC Register (1 / 2)

Bit name		Function
bit7	PGSEL: PG1/PG0 function select bit	This bit is used to select the function of the PG1/PG0 pins. If this bit is set to "0", the PG1/PG0 pins are selected as subclock oscillator pins, and the subclock oscillation is enabled or disabled by the subclock oscillation enable bit (SYCC2:SOSCE). If this bit is set to "1", the PG1/PG0 pins are selected as general-purpose I/O ports.
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	EXCK[1:0]: HCLK source select bits	This bit is used to select the external clock input pin to be used as the main oscillation clock. The main oscillator clock is selected from the main oscillation input at the X0/X1 pin, HCLK1 input or HCLK2 input as shown below. { EXCK table below }
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. 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	PPGSL: 8/16-bit PPG output port select bit	This bit is used to select the 8/16-bit PPG output ports. If this bit is set to "0", the P10-P11 pins and the P13-P16 pins are selected as the 8/16-bit PPG output ports. If this bit is set to "1", the P62-P67 pins are selected as the 8/16-bit PPG output ports. { PPGSL table below }
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 reset input enable/disable bit (bit0, SYSC:RSTEN) of 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.

EXCK[1:0] table (bit5, bit4):

EXCK1	EXCK0	HCLK input pin selection
0	0	HCLK1/HCLK2 input is disabled. (The main oscillation input at the X0/X1 pin is selected as the main oscillation clock input.)
0	1	P04 is selected as the main oscillation clock input pin for HCLK1.
1	0	P05 is selected as the main oscillation clock input pin for HCLK2.
1	1	Setting prohibited

PPGSL table (bit2):

PPG ch.	PPGSEL = 0	PPGSEL = 1
ch. 0	P13-P14	P62-P63
ch. 1	P10-P11	P64-P65
ch. 2	P15-P16	P66-P67

Table 31.2-1 Functions of Bits in SYSC Register (2 / 2)

Bit name		Function
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 MB95F332H/F333H/F334H 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.

Note:

To keep the reset input/output function after the reset, RSTEN (SYSC:bit 0) and RSTOE (SYSC:bit 1) are initialized to "1" after the power is switched on. They will not be initialized by any other type of reset.

If the reset input/output functions have to be used in the system, it is strongly recommended that SYSC:RSTEN be initialized to "1" in the initialize program routine after a reset for stable operation. With the reset input/output functions having been enabled, all types of reset, including the watchdog reset, can be used.

31.3 Notes on Using Controller

This section provides notes on using the controller.

■ Notes on Using Controller

● Setting input pin for EC0 and HCLK

Though P04 can be selected as the input pin for EC0 and HCLK, to avoid any unexpected result, <u>do not</u> set P04 as the input pin for both EC0 and HCLK at the same time.

● Setting PPGSEL to "0" when using the MPG function

While the MPG function is in use, P62-P67 are being used as MPG output ports. In this situation, if it is necessary to use the PPG function, set the PPGSEL bit to "0" to switch the PPG output ports to P10-P11 and P13-P16.

APPENDIX

This section shows the I/O map, interrupt list, memory map, pin states and mask options.

APPENDIX A   I/O Map

This section shows the I/O map used in the MB95330H 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	XXXXXXXX$_B$
000A$_H$	TBTC	Time-base timer control register	R/W	00000000$_B$
000B$_H$	WPCR	Watch prescaler control register	R/W	00000000$_B$
000C$_H$	WDTC	Watchdog timer control register	R/W	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$_H$ 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$_H$	PUL1	Port 1 pull-up register	R/W	00000000$_B$
002E$_H$ to 0034$_H$	—	(Disabled)	—	—
0035$_H$	PULG	Port G pull-up register	R/W	00000000$_B$
0036$_H$	T01CR1	8/16-bit composite timer 01 status control register 1 ch. 0	R/W	00000000$_B$
0037$_H$	T00CR1	8/16-bit composite timer 00 status control register 1 ch. 0	R/W	00000000$_B$
0038$_H$	T11CR1	8/16-bit composite timer 11 status control register 1 ch. 1	R/W	00000000$_B$
0039$_H$	T10CR1	8/16-bit composite timer 10 status control register 1 ch. 1	R/W	00000000$_B$
003A$_H$	PC01	8/16-bit PPG timer 01 control register	R/W	00000000$_B$
003B$_H$	PC00	8/16-bit PPG timer 00 control register	R/W	00000000$_B$

MB95330H Series

Table A-1 I/O MAP (2 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
003C$_H$	PC11	8/16-bit PPG timer 11 control register	R/W	00000000$_B$
003D$_H$	PC10	8/16-bit PPG timer 10 control register	R/W	00000000$_B$
003E$_H$	PC21	8/16-bit PPG timer 21 control register	R/W	00000000$_B$
003F$_H$	PC20	8/16-bit PPG timer 20 control register	R/W	00000000$_B$
0040$_H$	TMCSRH1	16-bit reload timer control status register upper ch. 1	R/W	00000000$_B$
0041$_H$	TMCSRL1	16-bit reload timer control status register lower ch. 1	R/W	00000000$_B$
0042$_H$, 0043$_H$	—	(Disabled)	—	—
0044$_H$	PCNTH1	16-bit PPG status control register upper ch. 1	R/W	00000000$_B$
0045$_H$	PCNTL1	16-bit PPG status control register lower ch. 1	R/W	00000000$_B$
0046$_H$, 0047$_H$	—	(Disabled)	—	—
0048$_H$	EIC00	External interrupt circuit control register ch. 0/ch. 1	R/W	00000000$_B$
0049$_H$	EIC10	External interrupt circuit control register ch. 2/ch. 3	R/W	00000000$_B$
004A$_H$	EIC20	External interrupt circuit control register ch. 4/ch. 5	R/W	00000000$_B$
004B$_H$	EIC30	External interrupt circuit control register ch. 6/ch. 7	R/W	00000000$_B$
004C$_H$	EIC01	External interrupt circuit control register ch. 8/ch. 9	R/W	00000000$_B$
004D$_H$ to 004F$_H$	—	(Disabled)	—	—
0050$_H$	SCR	LIN-UART serial control register	R/W	00000000$_B$
0051$_H$	SMR	LIN-UART serial mode register	R/W	00000000$_B$
0052$_H$	SSR	LIN-UART serial status register	R/W	00001000$_B$
0053$_H$	RDR/TDR	LIN-UART receive/transmit data register	R/W	00000000$_B$
0054$_H$	ESCR	LIN-UART extended status control register	R/W	00000100$_B$
0055$_H$	ECCR	LIN-UART extended communication control register	R/W	000000XX$_B$
0056$_H$	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	00000001$_B$
0059$_H$	TDR0	UART/SIO serial output data register ch. 0	R/W	00000000$_B$
005A$_H$	RDR0	UART/SIO serial input data register ch. 0	R	00000000$_B$
005B$_H$ to 005F$_H$	—	(Disabled)	—	—
0060$_H$	IBCR00	I^{2}C bus control register 0	R/W	00000000$_B$
0061$_H$	IBCR10	I^{2}C bus control register 1	R/W	00000000$_B$
0062$_H$	IBSR0	I^{2}C bus status register	R/W	00000000$_B$
0063$_H$	IDDR0	I^{2}C data register	R/W	00000000$_B$
0064$_H$	IAAR0	I^{2}C address register	R/W	00000000$_B$
0065$_H$	ICCR0	I^{2}C clock control register	R/W	00000000$_B$
0066$_H$	OPCUR	16-bit MPG output control register (upper)	R/W	00000000$_B$
0067$_H$	OPCLR	16-bit MPG output control register (lower)	R/W	00000000$_B$
0068$_H$	IPCUR	16-bit MPG input control register (upper)	R/W	00000000$_B$
0069$_H$	IPCLR	16-bit MPG input control register (lower)	R/W	00000000$_B$
006A$_H$	NCCR	16-bit MPG noise cancellation control register	R/W	00000000$_B$

Table A-1 I/O MAP (3 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
006B$_H$	TCSR	16-bit MPG timer 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$_H$	—	(Disabled)	—	—
0F80$_H$	WRARH0	Wild register address setting register (upper) ch. 0	R/W	00000000$_B$
0F81$_H$	WRARL0	Wild register address setting register (lower) ch. 0	R/W	00000000$_B$
0F82$_H$	WRDR0	Wild register data setting register ch. 0	R/W	00000000$_B$
0F83$_H$	WRARH1	Wild register address setting register (upper) ch. 1	R/W	00000000$_B$
0F84$_H$	WRARL1	Wild register address setting register (lower) ch. 1	R/W	00000000$_B$
0F85$_H$	WRDR1	Wild register data setting register ch. 1	R/W	00000000$_B$
0F86$_H$	WRARH2	Wild register address setting register (upper) ch. 2	R/W	00000000$_B$
0F87$_H$	WRARL2	Wild register address setting register (lower) ch. 2	R/W	00000000$_B$
0F88$_H$	WRDR2	Wild register data setting register ch. 2	R/W	00000000$_B$
0F89$_H$ to 0F91$_H$	—	(Disabled)	—	—
0F92$_H$	T01CR0	8/16-bit composite timer 01 status control register 0 ch. 0	R/W	00000000$_B$
0F93$_H$	T00CR0	8/16-bit composite timer 00 status control register 0 ch. 0	R/W	00000000$_B$
0F94$_H$	T01DR	8/16-bit composite timer 01 data register ch. 0	R/W	00000000$_B$
0F95$_H$	T00DR	8/16-bit composite timer 00 data register ch. 0	R/W	00000000$_B$
0F96$_H$	TMCR0	8/16-bit composite timer 00/01 timer mode control register ch. 0	R/W	00000000$_B$
0F97$_H$	T11CR0	8/16-bit composite timer 11 status control register 0 ch. 1	R/W	00000000$_B$
0F98$_H$	T10CR0	8/16-bit composite timer 10 status control register 0 ch. 1	R/W	00000000$_B$
0F99$_H$	T11DR	8/16-bit composite timer 11 data register ch. 1	R/W	00000000$_B$
0F9A$_H$	T10DR	8/16-bit composite timer 10 data register ch. 1	R/W	00000000$_B$

MB95330H Series

Table A-1 I/O MAP (4 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0F9B$_H$	TMCR1	8/16-bit composite timer 10/11 timer mode control register ch. 1	R/W	00000000$_B$
0F9C$_H$	PPS01	8/16-bit PPG01 cycle setting buffer register ch. 0	R/W	11111111$_B$
0F9D$_H$	PPS00	8/16-bit PPG00 cycle setting buffer register ch. 0	R/W	11111111$_B$
0F9E$_H$	PDS01	8/16-bit PPG01 duty setting buffer register ch. 0	R/W	11111111$_B$
0F9F$_H$	PDS00	8/16-bit PPG00 duty setting buffer register ch. 0	R/W	11111111$_B$
0FA0$_H$	PPS11	8/16-bit PPG11 cycle setting buffer register ch. 1	R/W	11111111$_B$
0FA1$_H$	PPS10	8/16-bit PPG10 cycle setting buffer register ch. 1	R/W	11111111$_B$
0FA2$_H$	PDS11	8/16-bit PPG11 duty setting buffer register ch. 1	R/W	11111111$_B$
0FA3$_H$	PDS10	8/16-bit PPG10 duty setting buffer register ch. 1	R/W	11111111$_B$
0FA4$_H$	PPGS	8/16-bit PPG startup register	R/W	00000000$_B$
0FA5$_H$	REVC	8/16-bit PPG output reverse register	R/W	00000000$_B$
0FA6$_H$	PPS21	8/16-bit PPG21 cycle setting buffer register ch. 2	R/W	11111111$_B$
0FA7$_H$	PPS20	8/16-bit PPG20 cycle setting buffer register ch. 2	R/W	11111111$_B$
0FA8$_H$	TMRH1	16-bit timer register (upper) ch. 1	R/W	00000000$_B$
	TMRLRH1	16-bit reload register (upper) ch. 1		
0FA9$_H$	TMRL1	16-bit timer register (lower) ch. 1	R/W	00000000$_B$
	TMRLRL1	16-bit reload register (lower) ch. 1		
0FAA$_H$	PDS21	8/16-bit PPG21 duty setting buffer register ch. 2	R/W	11111111$_B$
0FAB$_H$	PDS20	8/16-bit PPG20 duty setting buffer register ch. 2	R/W	11111111$_B$
0FAC$_H$ to 0FAF$_H$	—	(Disabled)	—	—
0FB0$_H$	PDCRH1	16-bit PPG down-counter register (upper) ch. 1	R	00000000$_B$
0FB1$_H$	PDCRL1	16-bit PPG down-counter register (lower) ch. 1	R	00000000$_B$
0FB2$_H$	PCSRH1	16-bit PPG cycle setting buffer register (upper) ch. 1	R/W	11111111$_B$
0FB3$_H$	PCSRL1	16-bit PPG cycle setting buffer register (lower) ch. 1	R/W	11111111$_B$
0FB4$_H$	PDUTH1	16-bit PPG duty setting buffer register (upper) ch. 1	R/W	11111111$_B$
0FB5$_H$	PDUTL1	16-bit PPG duty setting buffer register (lower) ch. 1	R/W	11111111$_B$
0FB6$_H$ to 0FBB$_H$	—	(Disabled)	—	—
0FBC$_H$	BGR1	LIN-UART baud rate generator register 1	R/W	00000000$_B$
0FBD$_H$	BGR0	LIN-UART baud rate generator register 0	R/W	00000000$_B$
0FBE$_H$	PSSR0	UART/SIO baud rate generator prescaler select register ch. 0	R/W	00000000$_B$
0FBF$_H$	BRSR0	UART/SIO baud rate generator baud rate setting register ch. 0	R/W	00000000$_B$
0FC0$_H$ to 0FC2$_H$	—	(Disabled)	—	—
0FC3$_H$	AIDRL	A/D input disable register (lower)	R/W	00000000$_B$
0FC4$_H$	OPDBRH0	16-bit MPG output data buffer register (upper) ch. 0	R/W	00000000$_B$
0FC5$_H$	OPDBRL0	16-bit MPG output data buffer register (lower) ch. 0	R/W	00000000$_B$
0FC6$_H$	OPDBRH1	16-bit MPG output data buffer register (upper) ch. 1	R/W	00000000$_B$
0FC7$_H$	OPDBRL1	16-bit MPG output data buffer register (lower) ch. 1	R/W	00000000$_B$
0FC8$_H$	OPDBRH2	16-bit MPG output data buffer register (upper) ch. 2	R/W	00000000$_B$
0FC9$_H$	OPDBRL2	16-bit MPG output data buffer register (lower) ch. 2	R/W	00000000$_B$

Table A-1 I/O MAP (5 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0FCA$_H$	OPDBRH3	16-bit MPG output data buffer register (upper) ch. 3	R/W	00000000$_B$
0FCB$_H$	OPDBRL3	16-bit MPG output data buffer register (lower) ch. 3	R/W	00000000$_B$
0FCC$_H$	OPDBRH4	16-bit MPG output data buffer register (upper) ch. 4	R/W	00000000$_B$
0FCD$_H$	OPDBRL4	16-bit MPG output data buffer register (lower) ch. 4	R/W	00000000$_B$
0FCE$_H$	OPDBRH5	16-bit MPG output data buffer register (upper) ch. 5	R/W	00000000$_B$
0FCF$_H$	OPDBRL5	16-bit MPG output data buffer register (lower) ch. 5	R/W	00000000$_B$
0FD0$_H$	OPDBRH6	16-bit MPG output data buffer register (upper) ch. 6	R/W	00000000$_B$
0FD1$_H$	OPDBRL6	16-bit MPG output data buffer register (lower) ch. 6	R/W	00000000$_B$
0FD2$_H$	OPDBRH7	16-bit MPG output data buffer register (upper) ch. 7	R/W	00000000$_B$
0FD3$_H$	OPDBRL7	16-bit MPG output data buffer register (lower) ch. 7	R/W	00000000$_B$
0FD4$_H$	OPDBRH8	16-bit MPG output data buffer register (upper) ch. 8	R/W	00000000$_B$
0FD5$_H$	OPDBRL8	16-bit MPG output data buffer register (lower) ch. 8	R/W	00000000$_B$
0FD6$_H$	OPDBRH9	16-bit MPG output data buffer register (upper) ch. 9	R/W	00000000$_B$
0FD7$_H$	OPDBRL9	16-bit MPG output data buffer register (lower) ch. 9	R/W	00000000$_B$
0FD8$_H$	OPDBRHA	16-bit MPG output data buffer register (upper) ch. A	R/W	00000000$_B$
0FD9$_H$	OPDBRLA	16-bit MPG output data buffer register (lower) ch. A	R/W	00000000$_B$
0FDA$_H$	OPDBRHB	16-bit MPG output data buffer register (upper) ch. B	R/W	00000000$_B$
0FDB$_H$	OPDBRLB	16-bit MPG output data buffer register (lower) ch. B	R/W	00000000$_B$
0FDC$_H$	OPDUR	16-bit MPG output data register (upper)	R	0000XXXX$_B$
0FDD$_H$	OPDLR	16-bit MPG output data register (lower)	R	XXXXXXXX$_B$
0FDE$_H$	CPCUR	16-bit MPG compare clear register (upper)	R/W	XXXXXXXX$_B$
0FDF$_H$	CPCLR	16-bit MPG compare clear register (lower)	R/W	XXXXXXXX$_B$
0FE0$_H$, 0FE1$_H$	—	(Disabled)	—	—
0FE2$_H$	TMBUR	16-bit MPG timer buffer register (upper)	R	XXXXXXXX$_B$
0FE3$_H$	TMBLR	16-bit MPG timer buffer register (lower)	R	XXXXXXXX$_B$
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$_H$	SYSC	System configuration register	R/W	11000011$_B$
0FE9$_H$	CMCR	Clock monitoring control register	R/W	00000000$_B$
0FEA$_H$	CMDR	Clock monitoring data register	R	00000000$_B$
0FEB$_H$	WDTH	Watchdog timer selection ID register (upper)	R	XXXXXXXX$_B$
0FEC$_H$	WDTL	Watchdog timer selection ID register (lower)	R	XXXXXXXX$_B$
0FED$_H$	—	(Disabled)	—	—
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)	—	—

MB95330H Series

- **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.

APPENDIX B   Table of Interrupt Sources

This section shows the table of interrupt sources used in the MB95330H Series.

■ Table of Interrupt Sources

See CHAPTER 5 "CPU" for interrupt operation.

Table B-1 MB95330H 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, ch. 4	IRQ00	$FFFA_H$	$FFFB_H$	L00 [1:0]	High
External interrupt ch. 1, ch. 5	IRQ01	$FFF8_H$	$FFF9_H$	L01 [1:0]	
External interrupt ch. 2, ch. 6	IRQ02	$FFF6_H$	$FFF7_H$	L02 [1:0]	
External interrupt ch. 3, ch. 7	IRQ03	$FFF4_H$	$FFF5_H$	L03 [1:0]	
UART/SIO ch. 0, MPG (DTTI)	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]	
LIN-UART (reception)	IRQ07	$FFEC_H$	$FFED_H$	L07 [1:0]	
LIN-UART (transmission)	IRQ08	$FFEA_H$	$FFEB_H$	L08 [1:0]	
8/16-bit PPG ch. 1 (lower)	IRQ09	$FFE8_H$	$FFE9_H$	L09 [1:0]	
8/16-bit PPG ch. 1 (upper)	IRQ10	$FFE6_H$	$FFE7_H$	L10 [1:0]	
8/16-bit PPG ch. 2 (upper)	IRQ11	$FFE4_H$	$FFE5_H$	L11 [1:0]	
8/16-bit PPG ch. 0 (upper)	IRQ12	$FFE2_H$	$FFE3_H$	L12 [1:0]	
8/16-bit PPG ch. 0 (lower)	IRQ13	$FFE0_H$	$FFE1_H$	L13 [1:0]	
8/16-bit composite timer ch. 1 (upper)	IRQ14	$FFDE_H$	$FFDF_H$	L14 [1:0]	
8/16-bit PPG ch. 2 (lower)	IRQ15	$FFDC_H$	$FFDD_H$	L15 [1:0]	
16-bit reload timer ch. 1, MPG (write timing/compare clear), I^2C	IRQ16	$FFDA_H$	$FFDB_H$	L16 [1:0]	
16-bit PPG timer ch. 1, MPG (position detection/compare match)	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]	
External interrupt ch. 8, ch. 9	IRQ21	$FFD0_H$	$FFD1_H$	L21 [1:0]	
8/16-bit composite timer ch. 1 (lower)	IRQ22	$FFCE_H$	$FFCF_H$	L22 [1:0]	
Flash memory	IRQ23	$FFCC_H$	$FFCD_H$	L23 [1:0]	Low

MB95330H Series

APPENDIX C   Memory Maps

This section shows the memory maps of the MB95330H Series.

■ Memory Maps

Figure C-1 Memory Maps of Different Products
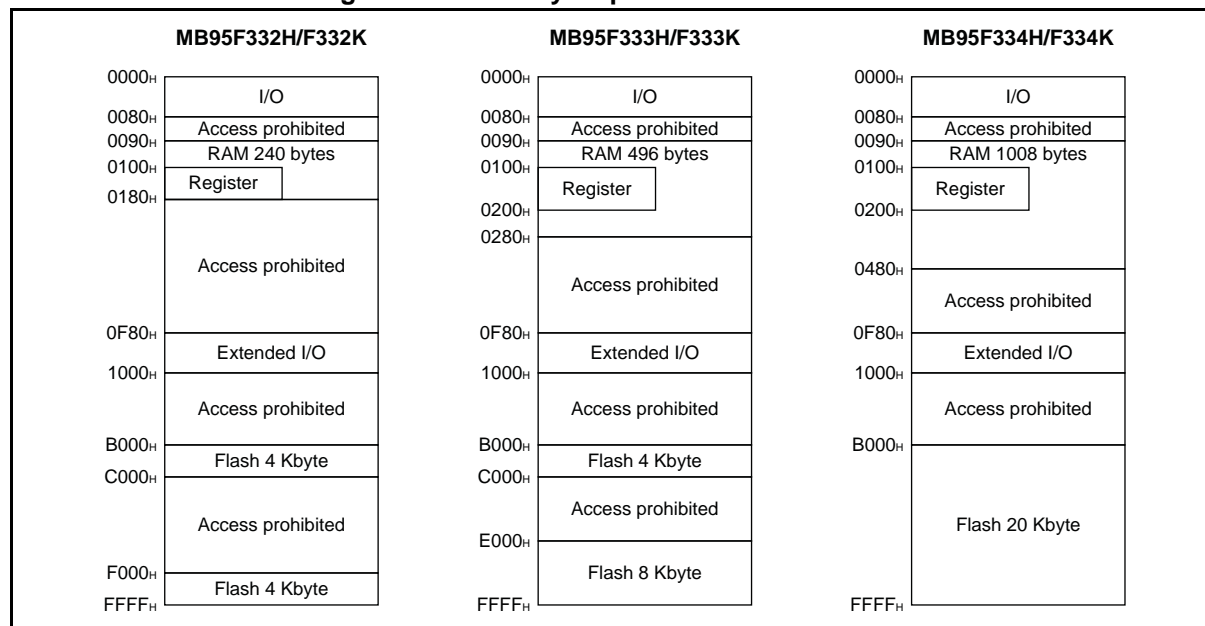
Parameter Part number	Flash memory	RAM
MB95F332H/F332K	8 Kbyte	240 bytes
MB95F333H/F333K	12 Kbyte	496 bytes
MB95F334H/F334K	20 Kbyte	1008 bytes

APPENDIX D　Pin States of MB95330H Series

Table D-1 below shows the pin states of the MB95330H Series in each mode.

■ Pin States in Each Mode

Table D-1 Pin States in Each Mode (1 / 3)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		In 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 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.)
PG1/X0A/ SNI1	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 - 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.)
PG2/X1A/ SNI2	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 - 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/$\overline{\text{RST}}$	Function I/O	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input[*3]
P60/INT08/ SDA/DTTI	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - 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 - Input interception[*2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - Input enabled[*1] (However, it does not function.)
P61/INT09/ SCL/TI1							
P62/TO10/ PPG00/OPT0	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.)
P63/TO11/ PPG01/OPT1							

MB95330H Series

Table D-1 Pin States in Each Mode (2 / 3)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		In reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P64/EC1/ PPG10/OPT2	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - 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 - Input interception[2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - Input enabled[1] (However, it does not function.)
P65/PPG11/ OPT3 P66/PPG1/ PPG20/OPT4	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.)
P67/TRG1/ PPG21/OPT5	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - 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 - Input interception[2] (However, an external interrupt can be input when the external interrupt is enabled.)	- Hi-Z - Input enabled[1] (However, it does not function.)
P10/PPG10 P11/PPG11	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[2]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[2]	- Retain - Input interception[2]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[2]	- Hi-Z - Input enabled[1] (However, it does not function.)
P12/DBG/ EC0	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.)
P13/PPG00	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception[2]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[2]	- Retain - Input interception[2]	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception[2]	- Hi-Z - Input enabled[1] (However, it does not function.)
P14/UCK0/ PPG01	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- 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[1] (However, it does not function.)

Table D-1 Pin States in Each Mode (3 / 3)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		In reset
			SPL=0	SPL=1	SPL=0	SPL=1	
P15/UO0/ PPG20	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception^{*2}	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception^{*2}	- Retain - Input interception^{*2}	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception^{*2}	- Hi-Z - Input enabled^{*1} (However, it does not function.)
P16/UI0/ PPG21	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- 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^{*1} (However, it does not function.)
P17/TO1/ SNI0	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	- Retain - Input interception^{*2}	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception^{*2}	- Retain - Input interception^{*2}	- Hi-Z (However, the setting of the pull-up control is effective.) - Input interception^{*2}	- Hi-Z - Input enabled^{*1} (However, it does not function.)
P00/INT00/ AN00 P01/INT01/ AN01 P02/INT02/ AN02/SCK P03/INT03/ AN03/SOT P04/INT04/ AN04/SIN/ HCLK1/EC0 P05/INT05/ AN05/ HCLK2/ TO00 P06/INT06/ AN06/TO01 P07/INT07/ AN07	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 interception^{*2}

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

MB95330H 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
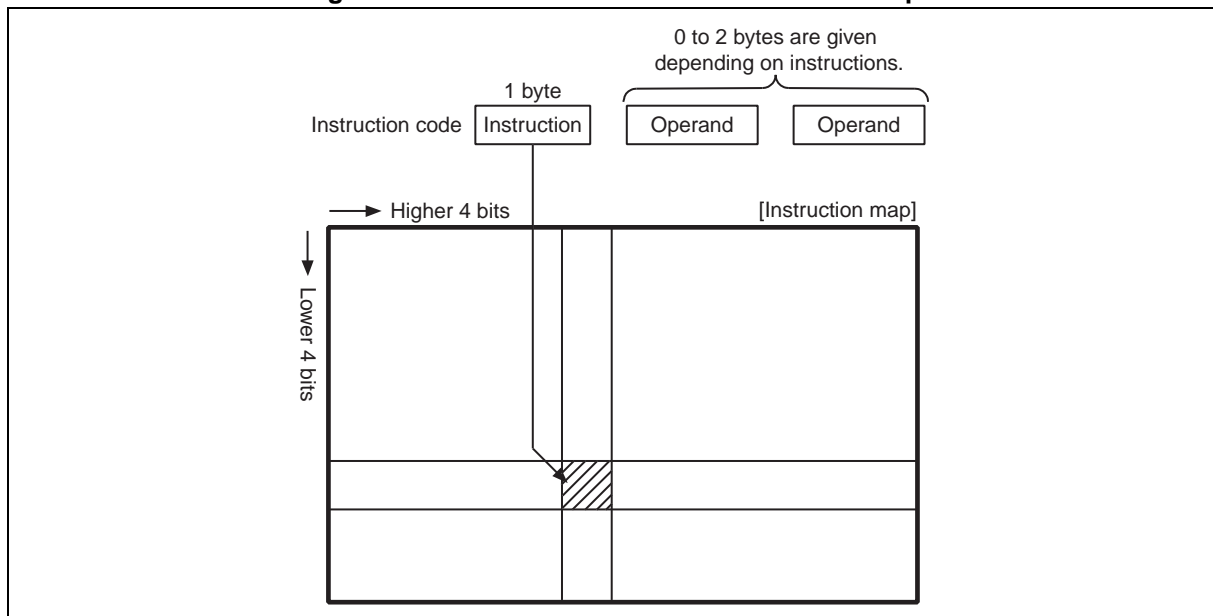
- The instruction is classified into following four types; forwarding system, operation system, branch system and others.
- There are various methods of addressing, and ten kinds of addressing can be selected by the selection and the operand specification of the instruction.
- This provides with the bit operation instruction, and can operate the read modification write.
- There is an instruction that directs special operation.

■ **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.)

MB95330H Series

■ 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
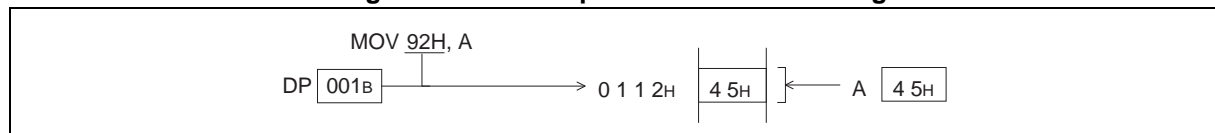
● Extended addressing

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

MB95330H Series
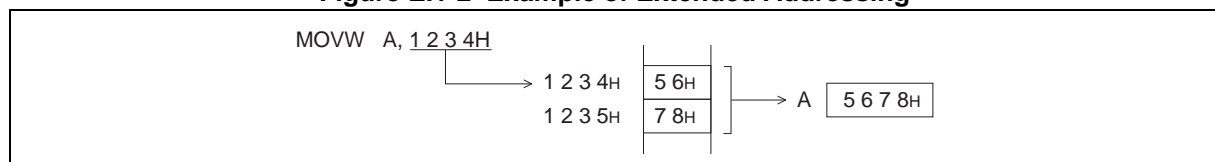
● Bit direct addressing

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
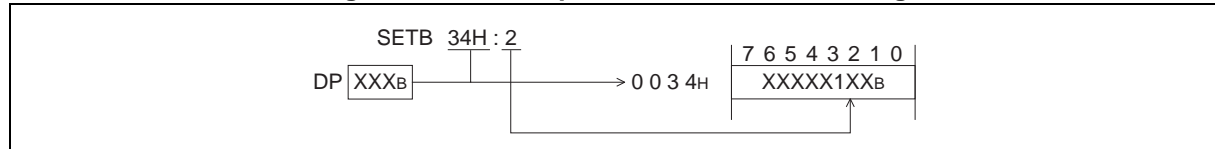
● Index addressing

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
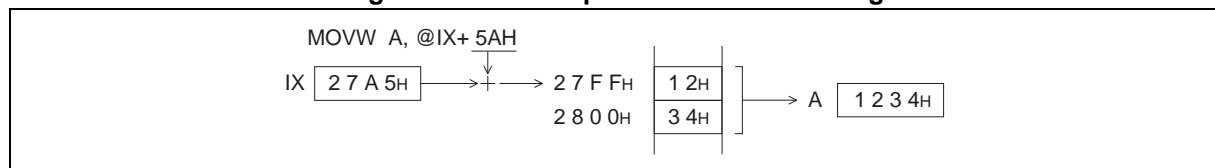
● Pointer addressing

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
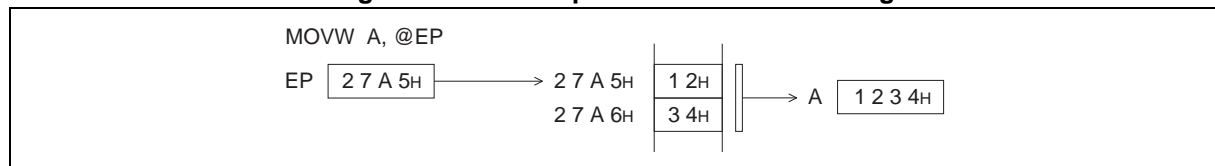
● General-purpose register addressing

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
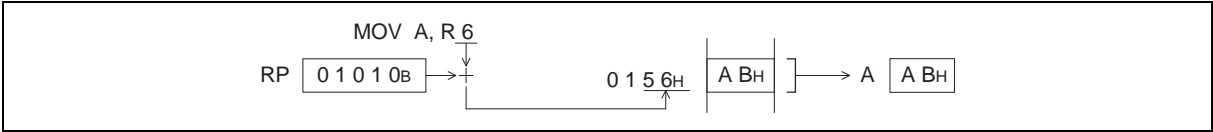
● Immediate addressing

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

MB95330H Series
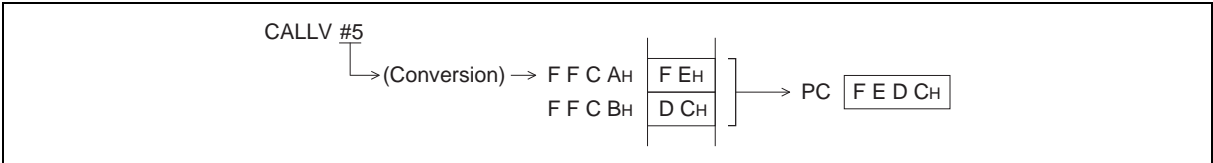
● Relative addressing

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

BNE FEH

Old PC $9 A B C_H$ → + — $9ABC_H + FFFE_H$ → New PC $9 A B A_H$
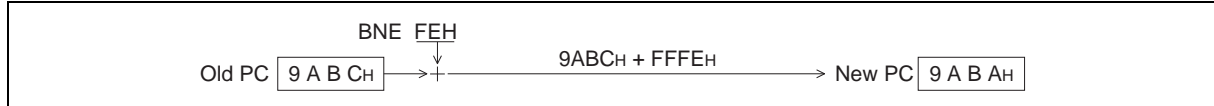
In this example, by jumping to the address where the operation code of BNE is stored, it results in an infinite loop.

● 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

NOP

Old PC $9 A B C_H$ ————————————→ New PC $9 A B D_H$
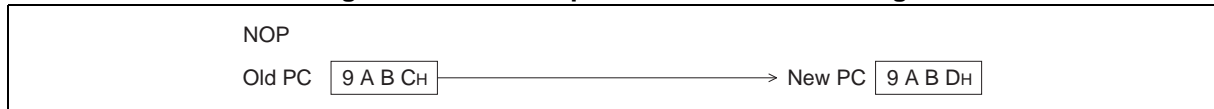
E.2   Special Instruction

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

(Before executing)		(After executing)
A	1 2 3 4H	A	1 2 3 4H
Old PC	X X X XH	New PC	1 2 3 4H
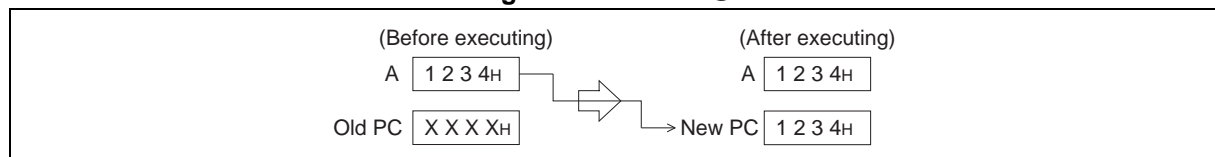
● MOVW A, PC

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
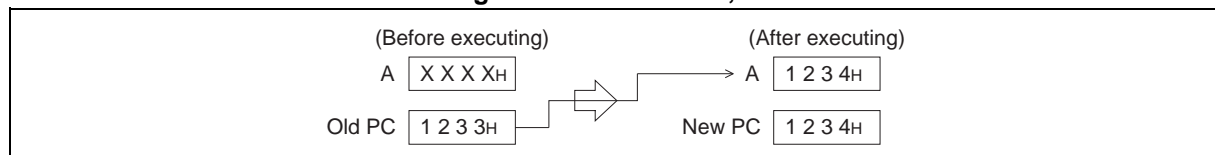
(Before executing)		(After executing)
A	X X X XH	A	1 2 3 4H
Old PC	1 2 3 3H	New PC	1 2 3 4H

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.

MB95330H Series

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
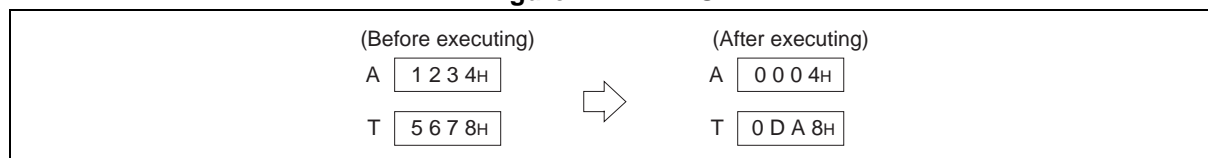
● XCHW A, PC

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
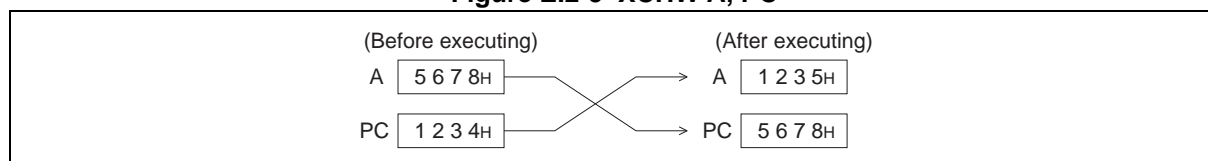
When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-5, the value "1235$_H$" stored in A corresponds to the address where 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
        .              │   │                   CMP   A, #EOL         data here
        .              │   │                   BNE   PTS1
                       │   │                   POPW  A
                       │   └────────────────   XCHW  A,  EP
                       └─────                   JMP   @A
```
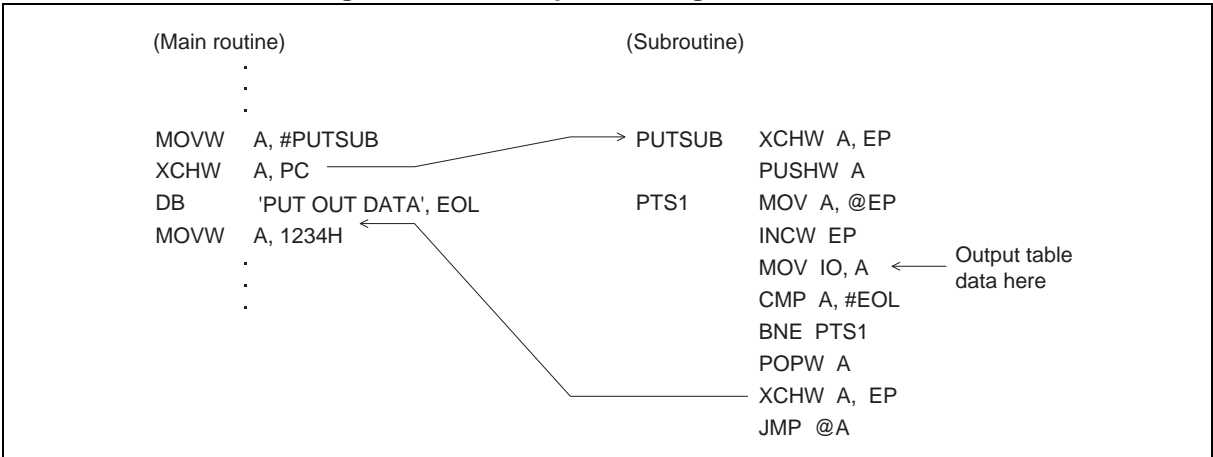
● CALLV #vct

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

```
        (Before executing)              (After executing)
PC    │ 5 6 7 8H │              ────→ PC   │ F E D CH │

SP    │ 1 2 3 4H │ ─(-2)─       ────→ SP   │ 1 2 3 2H │

1 2 3 2H  │ X XH │                   1 2 3 2H  │ 5 6H │
1 2 3 3H  │ X XH │                   1 2 3 3H  │ 7 9H │


F F C 6H  │ F EH │                   F F C 6H  │ F EH │
F F C 7H  │ D CH │                   F F C 7H  │ D CH │
```
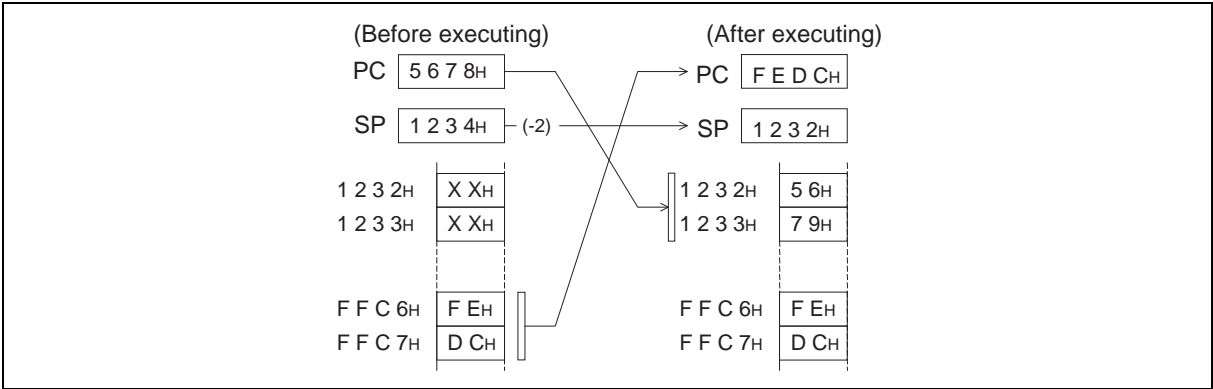
After the CALLV #vct instruction is executed, the contents of PC saved on the stack area are the address of the operation code of the next instruction, rather than the address of the operation code of CALLV #vct. Accordingly, Figure E.2-7 shows that the value saved in the stack (1232_H and 1233_H) is 5679_H, which is the address of the operation code of the instruction that follows "CALLV vct" (return address).

MB95330H Series

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.

MB95330H Series

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

MB95330H Series

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^{2}MC-8FX.

■ Instruction Map

Table E.5-1 Instruction Map of F^{2}MC-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	MOVW @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

MB95330H Series

APPENDIX F   Mask Options

Table F-1 shows the mask option list of the MB95330H Series.

■ Mask Option List

Table F-1 Mask Option List

No.	Part Number	MB95F332H MB95F333H MB95F334H	MB95F332K MB95F333K MB95F334K
	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

MB95330H Series

INDEX

The index starts from the next page.
It is arranged in alphabetical order.

Index

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series

MB95330H Series
Register Index

MB95330H Series

MB95330H Series

Pin Function Index

MB95330H Series

Interrupt Vector Index

I

FUJITSU SEMICONDUCTOR LIMITED