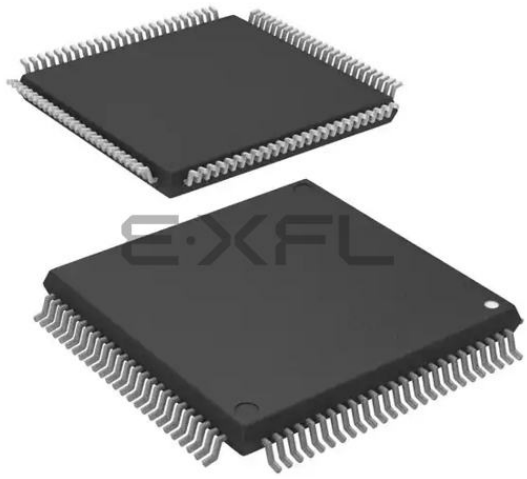
Renesas Electronics America Inc - R5F36CAENFB#30 Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	M16C/60
Core Size	16-Bit
Speed	32MHz
Connectivity	EBI/EMI, I²C, SIO, UART/USART, USB
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	85
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 26x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LFQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f36caenfb-30

9.3.2 Clock Mode Transition Procedure

Figure 9.1 shows Clock Mode Transition. Arrows indicate possible mode transitions.

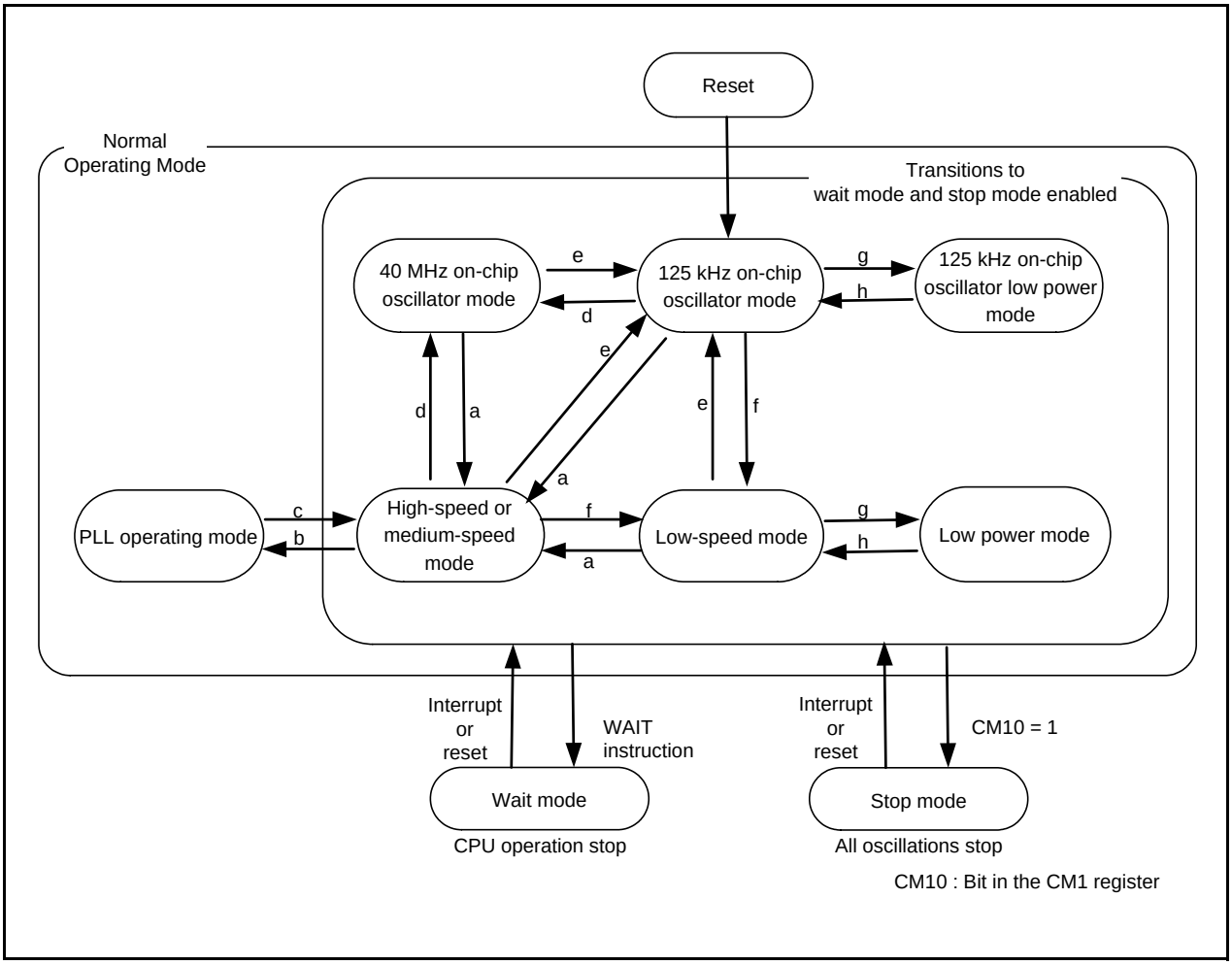


Figure 9.1 Clock Mode Transition

To start or stop clock oscillations, or to change modes in normal operating mode, follow the instructions below.

- Enter a different mode after the clock for that mode stabilizes completely.
- When stopping a clock, do it after mode transition is completed. Do not stop the clock at the same time as mode transition.
- When entering a new mode from PLL operating mode, high-speed or medium-speed mode, 40 MHz on-chip oscillator mode, or 125 kHz on-chip oscillator mode, or entering one of these modes from another mode, select divide by 8 or divide by 16.
- When the clock division ratio is switched in PLL operating mode, high-speed or medium-speed mode, or 40 MHz on-chip oscillator mode, the ratio changes in the order shown in Figure 9.2.
- To change the mode, follow procedures a to h listed below. For details on register and bit access, refer to 9.2 “Registers”. Letters a to h correspond to those in Figure 9.1 “Clock Mode Transition” and Figure 9.2 “Clock Divide Transition”.
- For details on oscillator start and stop, refer to 8.3.1 “Main Clock” to 8.3.6 “Sub Clock (fC)”.

12. Memory Space Expansion Function

12.1 Introduction

The following describes the memory space expansion function. In memory expansion or microprocessor mode, the memory space expansion function allows the access space to be expanded. Table 12.1 lists Memory Space Expansion Function Specifications. In this chapter, the external area accessed by the $\overline{CSi}$ (i = 0 to 3) signal is referred to as the $\overline{CSi}$ area.

Table 12.1 Memory Space Expansion Function Specifications

Item	Specification
1-MB mode	• Memory space 1 MB (no expansion) • Specify the external area ($\overline{CSi}$ area) accessed by the $\overline{CSi}$ signal.

i = 0 to 3

12.2 Registers

Table 12.2 lists registers related to the memory expansion function. Refer to 10. “Processor Mode” for the PM1 register.

Table 12.2 Registers

Address	Register	Symbol	Reset Value
0005h	Processor Mode Register 1	PM1	0000 1000b

14.2 Registers

Table 14.3 Registers (1/2)

Address	Register	Symbol	Reset Value
001Eh	Processor Mode Register 2	PM2	XX00 0X01b
0042h	$\overline{\text{INT7}}$ Interrupt Control Register	INT7IC	XX00 X000b
0043h	$\overline{\text{INT6}}$ Interrupt Control Register	INT6IC	XX00 X000b
0044h	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	XX00 X000b
0045h	Timer B5 Interrupt Control Register	TB5IC	XXXX X000b
0046h	Timer B4 Interrupt Control Register, UART1 Bus Collision Detection Interrupt Control Register	TB4IC, U1BCNIC	XXXX X000b
0047h	Timer B3 Interrupt Control Register, UART0 Bus Collision Detection Interrupt Control Register	TB3IC, U0BCNIC	XXXX X000b
0048h	$\overline{\text{INT5}}$ Interrupt Control Register	INT5IC	XX00 X000b
0049h	$\overline{\text{INT4}}$ Interrupt Control Register	INT4IC	XX00 X000b
004Ah	UART2 Bus Collision Detection Interrupt Control Register	BCNIC	XXXX X000b
004Bh	DMA0 Interrupt Control Register	DM0IC	XXXX X000b
004Ch	DMA1 Interrupt Control Register	DM1IC	XXXX X000b
004Dh	Key Input Interrupt Control Register, A/D Conversion (A/D1) Interrupt Control Register	KUPIC, ADEIC	XXXX X000b
004Eh	A/D Conversion (A/D0) Interrupt Control Register	ADIC	XXXX X000b
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	XXXX X000b
0050h	UART2 Receive Interrupt Control Register	S2RIC	XXXX X000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXX X000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXX X000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXX X000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXX X000b
0055h	Timer A0 Interrupt Control Register	TA0IC	XXXX X000b
0056h	Timer A1 Interrupt Control Register	TA1IC	XXXX X000b
0057h	Timer A2 Interrupt Control Register	TA2IC	XXXX X000b
0058h	Timer A3 Interrupt Control Register	TA3IC	XXXX X000b
0059h	Timer A4 Interrupt Control Register	TA4IC	XXXX X000b
005Ah	Timer B0 Interrupt Control Register	TB0IC	XXXX X000b
005Bh	Timer B1 Interrupt Control Register	TB1IC	XXXX X000b
005Ch	Timer B2 Interrupt Control Register	TB2IC	XXXX X000b
005Dh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	XX00 X000b
005Eh	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	XX00 X000b
005Fh	$\overline{\text{INT2}}$ Interrupt Control Register	INT2IC	XX00 X000b
0069h	DMA2 Interrupt Control Register	DM2IC	XXXX X000b
006Ah	DMA3 Interrupt Control Register	DM3IC	XXXX X000b
006Bh	UART5 Bus Collision Detection Interrupt Control Register	U5BCNIC	XXXX X000b
006Ch	UART5 Transmit Interrupt Control Register	S5TIC	XXXX X000b
006Dh	UART5 Receive Interrupt Control Register	S5RIC	XXXX X000b
006Eh	UART4 Bus Collision Detection Interrupt Control Register, Real-Time Clock Periodic Interrupt Control Register	U4BCNIC, RTCTIC	XXXX X000b

15.2.5 Watchdog Timer Control Register (WDC)

Watchdog Timer Control Register

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	Reset Value
	0	X						WDC	037Fh	00XX XXXXb

Bit Symbol	Bit Name	Function	RW
WDC0	Higher-order bits (b14 to b10) of the watchdog timer can be read		RO
WDC1			RO
WDC2			RO
WDC3			RO
WDC4			RO
— (b5)	No register bit. If necessary, set to 0. The read value is undefined.		—
— (b6)	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0 : Divide-by-16 1 : Divide-by-128	RW

WDC4-WDC0 (b4-b0)

When reading the watchdog timer value while the CSPRO bit in the CSPR register is 1 (count source protection mode enabled), read bits WDC4 to WDC0 more than three times to determine the values.

17. Timer A

17.1 Introduction

Timers A consists of timers A0 to A4. Each timer operates independently of the others. Table 17.1 lists Timer A Specifications, Table 17.2 lists Differences in Timer A Mode, Figure 17.1 shows Timer A and B Count Sources, Figure 17.2 shows Timer A Configuration, Figure 17.3 shows Timer A Block Diagram, and Table 17.3 lists I/O Ports.

Table 17.1 Timer A Specifications

Item	Specification
Configuration	16-bit timer × 5
Operating modes	• Timer mode The timer counts an internal count source. • Event counter mode The timer counts pulses from an external device, or overflows and underflows of other timers. • One-shot timer mode The timer outputs a single pulse before it reaches the count 0000h. • Pulse width modulation mode (PWM mode) The timer outputs pulses of given width and cycle successively. • Programmable output mode The timer outputs a given pulse width of a high/low level signal (timers A1, A2, and A4).
Interrupt sources	Overflow/underflow × 5

Table 17.2 Differences in Timer A Mode

Item	Timer				
	A0	A1	A2	A3	A4
Event counter mode (two-phase pulse signal processing)	No	No	Yes	Yes	Yes
Programmable output mode	No	Yes	Yes	No	Yes

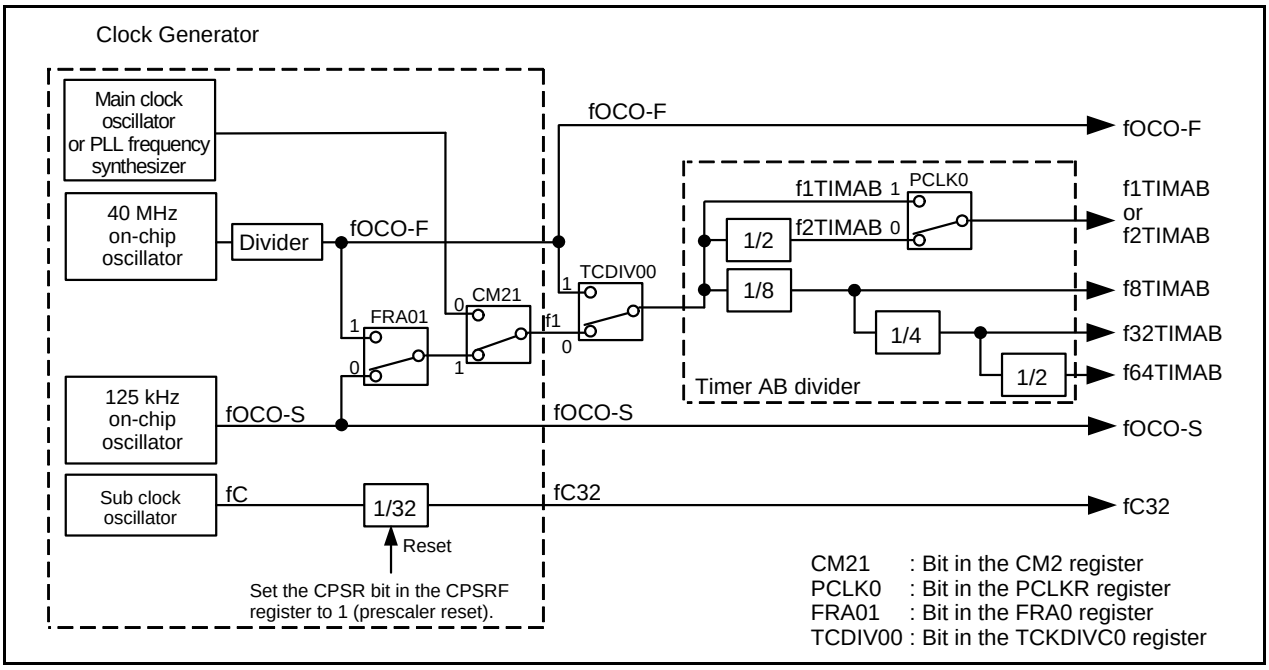


Figure 17.1 Timer A and B Count Sources

Table 17.9 Registers and Settings in Event Counter Mode (When Not Using Two-Phase Pulse Signal Processing) (1)

Register	Bit	Function and Setting
PCLKR	PCLK0	- (setting unnecessary)
CPSRF	CPSR	Write 1 to reset the clock prescaler.
TCKDIVC0	TCDIV00	- (setting unnecessary)
PWMFS	PWMFSi	Set to 0.
TACS0 to TACS2	7 to 0	- (setting unnecessary)
TAPOFS	POFSi	Select the output polarity when the MR0 bit in the TAiMR register is 1 (pulse output).
TAOW	TAiOW	Set to 0.
TAi1	15 to 0	- (setting unnecessary)
TABSR	TAiS	Set to 1 when starting counting. Set to 0 when stopping counting.
ONSF	TAiOS	Set to 0.
	TAZIE	Set to 0.
	TA0TGH to TA0TGL	Select a count source.
TRGSR	TAiTGH to TAiTGL	Select a count source.
UDF	TAiUD	Select a count operation.
	TAiP	Set to 0.
TAi	15 to 0	Set the counter value.
TAiMR	7 to 0	Refer to the TAiMR register below.

i = 0 to 4

Note:

1. This table does not describe a procedure.

18. Timer B

18.1 Introduction

Timer B consists of timers B0 to B5. Each timer operates independently of the others. Table 18.1 lists Timer B Specifications, Figure 18.1 shows Timer A and B Count Sources, Figure 18.2 shows the Timer B Configuration, Figure 18.3 shows the Timer B Block Diagram, and Table 18.2 lists the I/O Ports.

Table 18.1 Timer B Specifications

Item	Specification
Configuration	16-bit timer × 6
Operating modes	• Timer mode The timer counts an internal count source. • Event counter mode The timer counts pulses from an external device, or overflows and underflows of other timers. • Pulse period/pulse width measurement modes The timer measures pulse periods or pulse widths of an external signal.
Interrupt source	Overflow/underflow/active edge of measurement pulse × 6

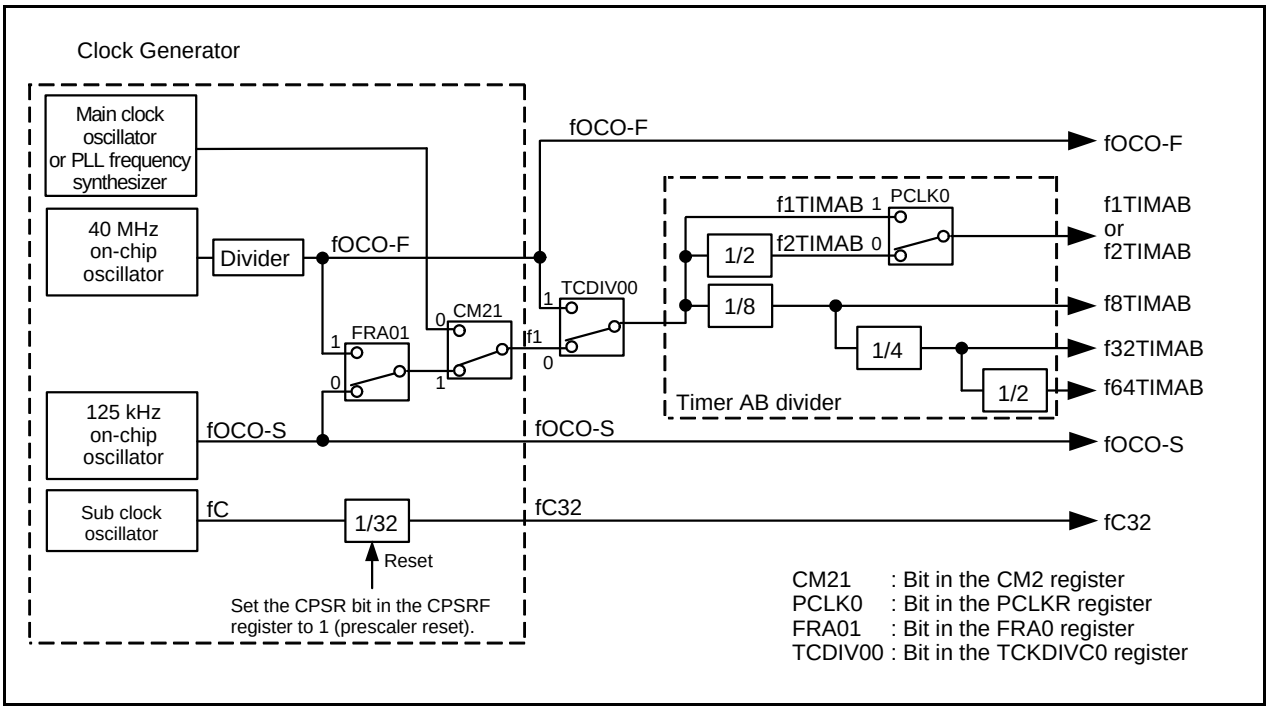


Figure 18.1 Timer A and B Count Sources

18.3.1.3 Count Source

Internal clocks are counted in timer mode, pulse period measurement mode, and pulse width measurement mode. Refer to Figure 18.1 “Timer A and B Count Sources” for details. Table 18.4 lists Timer B Count Sources.

f1 is any of the clocks listed below. Refer to 8. “Clock Generator” for details.

- Main clock divided by 1 (no division)
- PLL clock divided by 1 (no division)
- fOCO-S divided by 1 (no division)
- fOCO-F divided by 1 (no division)

Table 18.4 Timer B Count Sources

Count Source	Bit Setting Value				Remarks
	PCLK0	TCS3 TCS7	TCS2 to TCS0 TCS6 to TCS4	TCK1 to TCK0	
f1TIMAB	1	0	-	00b	f1 or fOCO-F (1)
		1	000b	-	
f2TIMAB	0	0	-	00b	f1 divided by 2 or fOCO-F divided by 2 (1)
		1	000b	-	
f8TIMAB	-	0	-	01b	f1 divided by 8 or fOCO-F divided by 8 (1)
		1	001b	-	
f32TIMAB	-	0	-	10b	f1 divided by 32 or fOCO-F divided by 32 (1)
		1	010b	-	
f64TIMAB	-	1	011b	-	f1 divided by 64 or fOCO-F divided by 64 (1)
fOCO-F	-	1	100b	-	fOCO-F
fOCO-S	-	1	101b	-	fOCO-S
fC32	-	0	-	11b	fC32
		1	110b	-	

PCLK0: Bit in the PCLKR register

TCS7 to TCS0: Bits in registers TBCS0 to TBCS3

TCK1 to TCK0: Bits in the TBiMR register (i = 0 to 5)

Note:

1. Select f1 or fOCO-F by setting the TCDIV00 bit in the TCKDIVC0 register.

20.2.7 Base Timer Control Register 1 (G1BCR1)

Base Timer Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
0				0			0

Symbol	Address	Reset Value
G1BCR1	02E3h	00h

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Set to 0.	RW
RST1	Base timer reset source select bit 1	0: The base timer is not reset when the base timer and G1PO0 register values match. 1: The base timer is reset when the base timer and G1PO0 register values match.	RW
RST2	Base timer reset source select bit 2	0: The base timer is not reset when low is input to the $\overline{INT1}$ pin. 1: The base timer is reset when low is input to the $\overline{INT1}$ pin.	RW
— (b3)	Reserved	Set to 0.	RW
BTS	Base timer start bit	0: Base timer reset 1: Base timer starts counting	RW
UD0	Increment/decrement control bit	b6 b5 0 0: Increment 0 1: Increment/decrement 1 0: Two-phase pulse signal processing 1 1: Do not set.	RW
UD1			
— (b7)	Reserved	Set to 0.	RW

RST1 (Base timer reset source select bit 1) (b1)

To rewrite the RST1 bit, rewrite it while the BTS bit is 0 (base timer reset) and then change the BTS bit to 1 (base timer starts counting).

When the base timer value matches the G1PO0 register value while the RST1 bit is 1, the base timer is reset after two fBT1 cycles. Refer to 20.3.1.4 “Base Timer Reset While the Base Timer is Counting” for details. When the RST1 bit is 1, set the RST4 bit in the G1BCR0 register to 0 (the base timer is not reset when the base timer and G1BTRR register values match).

RST2 (Base timer reset source select bit 2) (b2)

To rewrite the RST2 bit, rewrite it while the BTS bit is 0 and then rewrite the BTS bit to 1.

BTS (Base timer start bit) (b4)

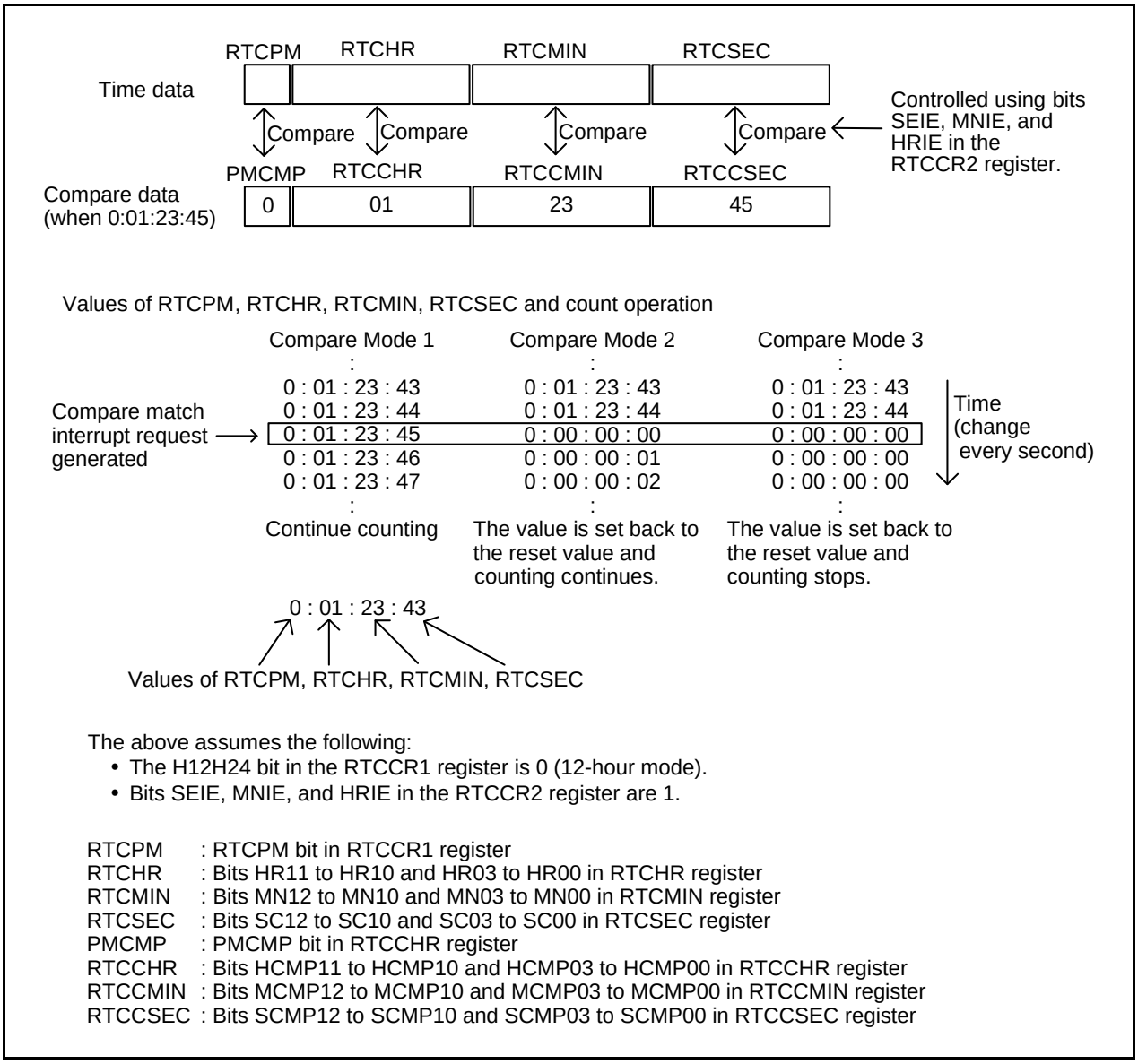
The value written to this bit is reflected to the internal circuit when the clock is synchronized with the base timer count source (fBT1).

UD1 and UD0 (Increment/decrement control bit) (b6-b5)

To rewrite bits UD1 and UD0, rewrite them while the BTS bit is 0 and then rewrite the BTS bit to 1.

When single-waveform output mode or SR waveform output mode is selected, set bits UD1 and UD0 to 00b (increment). When inverted waveform output mode is selected, set these bits to 00b (increment) or 01b (increment/decrement).

Figure 21.6 shows Difference between Compare Modes, Figure 21.7 shows Count Start/Stop Operating Example, Figure 21.8 shows Compare Mode 1 Operating Example, Figure 21.9 shows Compare Mode 2 Operating Example, and Figure 21.10 shows Compare Mode 3 Operating Example.



The above assumes the following:

- The H12H24 bit in the RTCCR1 register is 0 (12-hour mode).
- Bits SEIE, MNIE, and HRIE in the RTCCR2 register are 1.

RTCPM : RTCPM bit in RTCCR1 register
RTCHR : Bits HR11 to HR10 and HR03 to HR00 in RTCHR register
RTCMIN : Bits MN12 to MN10 and MN03 to MN00 in RTCMIN register
RTCSEC : Bits SC12 to SC10 and SC03 to SC00 in RTCSEC register
PMCMP : PMCMP bit in RTCCHR register
RTCCHR : Bits HCMP11 to HCMP10 and HCMP03 to HCMP00 in RTCCHR register
RTCCMIN : Bits MCMP12 to MCMP10 and MCMP03 to MCMP00 in RTCCMIN register
RTCCSEC : Bits SCMP12 to SCMP10 and SCMP03 to SCMP00 in RTCCSEC register

Figure 21.6 Difference between Compare Modes

23.3.1.2 Bit Rate and Duty Cycle

Bit rate is determined by a combination of fVIIC, the FASTMODE bit in the S20 register, and bits CCR4 to CCR0 in the S20 register.

Table 23.11 lists the Bit Rate of Internal SCL Output and Duty Cycle. When the change in the internal SCL output high level is a negative value, although the low period increases the amount that the high periods decreases, the bit rate does not increase. The values described is the following table are the values of the internal SCL output before being effected by the SCL output of an external device.

Table 23.11 Bit Rate of Internal SCL Output and Duty Cycle

Item	Standard Clock Mode (FASTMODE = 0)	Fast-mode (FASTMODE = 1) (CCR value = other than 5)	Fast-mode (FASTMODE = 1) (CCR value = 5)
Bit rate (bps)	$\frac{fVIIC}{8 \times CCR\ value}$	$\frac{fVIIC}{4 \times CCR\ value}$	$\frac{fVIIC}{2 \times CCRvalue} = \frac{fVIIC}{10}$
Duty cycle	50% Fluctuation of high level: -4 to +2 fVIIC cycles	50% Fluctuation of high level: -2 to +2 fVIIC cycles	35 to 45%

CCR value: Value set to bits CCR4 to CCR0

When the CCR value (setting value of bits CCR4 to CCR0) is 5 (00101b) in fast-mode, the bit rate is assumed to reach 400 kbps, the maximum bit rate in fast-mode.

The bit rate and duty cycle are as follows.

- Bit rate:

$$\frac{fVIIC}{2 \times CCR\ value} = \frac{fVIIC}{10}$$

When fVIIC is 4 MHz, the bit rate is 400 kbps.

- Duty cycle is 35 to 45%

Even if the bit rate is 400 kbps, the 1.3 μs minimum low period of the SCLMM clock (I2C-bus standard) is allocated. Table 23.12 lists the Bit Setting for Bits CCR4 to CCR0 and Bit Rate (fVIIC = 4 MHz).

Table 23.12 Bit Setting for Bits CCR4 to CCR0 and Bit Rate (fVIIC = 4 MHz)

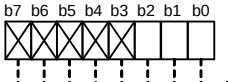
Bits CCR4 to CCR0 in the S20 Register					Bit Rate (kbps)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard Clock Mode	Fast-mode
0	0	0	0	0	Do not set (1)	Do not set (1)
0	0	0	0	1	Do not set (1)	Do not set (1)
0	0	0	1	0	Do not set (1)	Do not set (1)
0	0	0	1	1	Do not set (2)	333
0	0	1	0	0	Do not set (2)	250
0	0	1	0	1	100	400
0	0	1	1	0	83.3	166
:	:	:	:	:	:	:
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes:

1. Do not set bits CCR4 to CCR0 to 0 to 2 regardless of the fVIIC frequency.
2. Do not exceed the maximum bit rates of 100 kbps in standard clock mode and 400 kbps in fast-mode.

24.2.25 USB FIFO Clear Register j (USBFCLRj) (j = 1, 2)

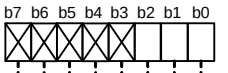
USB FIFO Clear Register 1 (USBFCLR1)



Symbol	Address	Reset Value
USBFCLR1	D199h	XXh

Bit Symbol	Bit Name	Function	RW
EP1CLR	Endpoint 1 FIFO buffer clear bit	When this bit is set to 1, both layers of the endpoint 1 receive FIFO buffer are cleared.	WO
EP2CLR	Endpoint 2 FIFO buffer clear bit	When this bit is set to 1, both layers of the endpoint 2 transmit FIFO buffer are cleared.	WO
EP3CLR	Endpoint 3 FIFO buffer clear bit	When this bit is set to 1, the endpoint 3 transmit FIFO buffer is cleared.	WO
— (b7-b3)	No register bits. Should be written with 0 and read as undefined.		—

USB FIFO Clear Register 2 (USBFCLR2)



Symbol	Address	Reset Value
USBFCLR2	D19Ah	XXh

Bit Symbol	Bit Name	Function	RW
EP4CLR	Endpoint 4 FIFO buffer clear bit	When this bit is set to 1, both layers of the endpoint 4 receive FIFO buffer are cleared.	WO
EP5CLR	Endpoint 5 FIFO buffer clear bit	When this bit is set to 1, both layers of the endpoint 5 transmit FIFO buffer are cleared.	WO
EP6CLR	Endpoint 6 FIFO buffer clear bit	When this bit is set to 1, the endpoint 6 transmit FIFO buffer is cleared.	WO
— (b7-b3)	No register bits. Should be written with 0 and read as undefined values.		—

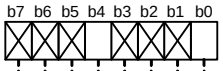
Access the USBFCLRj register in 8-bit units. Use the MOV instruction to access this register.

EPiCLR (Endpoint i FIFO buffer clear bit) (b2-b0) (i = 1 to 6)

Do not write 1 to the EPiCLR bit when endpoint i transmitting or receiving.

24.2.26 USB Endpoint Stall Register 0 (USBEPSTL0)

USB Endpoint Stall Register 0 (USBEPSTL0)



Symbol	Address	Reset Value
USBEPSTL0	D1A0h	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
EP0STLS	Endpoint 0 stall setting bit	0: Default 1: Stall	RW
— (b3-b1)	No register bits. Should be written with 0 and read as undefined values.		—
EP0STLC	Endpoint 0 stall clear bit	When this bit is set to 1, the EP0STLS bit becomes 0 (default).	WO
— (b7-b5)	No register bits. Should be written with 0 and read as undefined values.		—

Access the USBEPSTL0 register in 8-bit units.
Use the MOV instruction when writing to this register. Any bits that are not set to 1 should be set to 0.

EP0STLS (Endpoint 0 stall setting bit) (b0)

To set the USB bus line to the stall state by a program, set the EP0STLS bit to 1 (stall). Writing 0 has no effect.
Conditions to become 0:

- Write 1 to the EP0STLC bit.
- Receive the 8-byte setup command to decode by a program (refer to Table 24.13 “Processing Commands When Received”).
- The EP0ASCE bit in the USBCTLR register is 1 (stall auto-clear enabled), and STALL is returned in response to a transaction from the host.

Do not set bits EP0STLS and EP0STLC to 1 simultaneously (by one instruction).
When the SETUPTS bit in the USBIFR1 register is set to 1 (receive completion of setup command), the bus line is not stalled even if the EP0STLS bit is set to 1.

EP0STLC (Endpoint 0 stall clear bit) (b4)

Do not set bits EP0STLS and EP0STLC to 1 simultaneously (by one instruction).

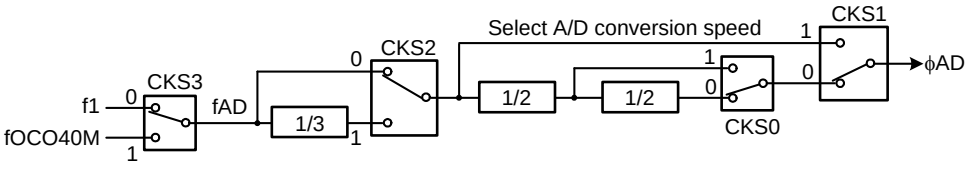
25.3 Operations

25.3.1 A/D Conversion Cycle

A/D conversion cycle is based on fAD and ϕAD. Figure 25.4 shows fAD and ϕAD. fAD and ϕAD are set for A/D0 and A/D1 separately.

When the CKS3 bit in the AD0CON2 register is 1 (fOCO40M is fAD), do not set the CKS2 bit in the AD0CON2 register to 0 and the CKS1 bit in the AD0CON1 register to 1 (fAD = ϕAD). Set registers associated with A/D converter after setting the CKS3 bit.

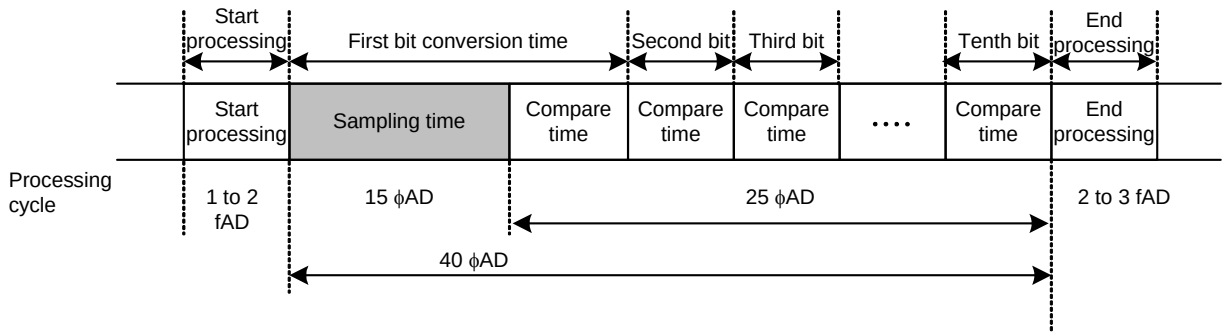
Also, when the CKS3 bit in the AD1CON2 register is 1 (fOCO40M is fAD), do not set the CKS2 bit in the AD1CON2 register to 0 and the CKS1 bit in the AD1CON1 register to 1 (fAD = ϕAD). Set registers associated with A/D converter after setting the CKS3 bit.



CKS0: Bit in registers AD0CON0 and AD1CON0
CKS1: Bit in registers AD0CON1 and AD1CON1
CKS2, CKS3: Bits in registers AD0CON2 and AD1CON2

Figure 25.4 fAD and ϕAD

Figure 25.5 shows A/D Conversion Timing.



The above figure applies under the following conditions:
- One-shot mode
- ϕAD = fAD

Figure 25.5 A/D Conversion Timing

28.8.6.2 Handling Procedure for Errors

When errors occur, follow the procedures below.
Do not execute the program, block erase, lock bit program, and block blank check commands when either FMR06 or FMR07 bit is 1 (completed in error). Execute each command after executing the clear status register command.

Command sequence error

(1) Execute the clear status register command and set bits FMR06 and FMR07 to 0 (completed as expected).
(2) Check if the command is written correctly and execute the correct command.

Erase error

(1) Execute the clear status register command and set the FMR07 bit to 0 (completed as expected).
(2) Execute the read lock bit status command. If the lock bit in the block where the error occurred is set to 0 (locked), set the FMR02 bit in the FMR register to 1 (lock bit disabled).
(3) Execute the block erase command again.
(4) Repeat (1) to (3) until an erase error is not generated.

If an error still occurs even after repeating three times, do not use that block.
When handling an erase error of the block blank check command and erasing is not necessary, execute (1) only.

Program error

[When a program operation is executed]

(1) Execute the clear status register command and set the FMR06 bit to 0 (completed as expected).
(2) Execute the read lock bit status command. If the lock bit in the block where the error occurred is set to 0, set the FMR02 bit in the FMR0 register to 1.
(3) Execute the program command again.

If the lock bit is set to 1 (unlocked), do not use the address in which error has occurred as it is. Execute the block erase command to erase the block, in which the error has occurred, before executing the program command to write to the same address again.
If an error still occurs, do not use that block.

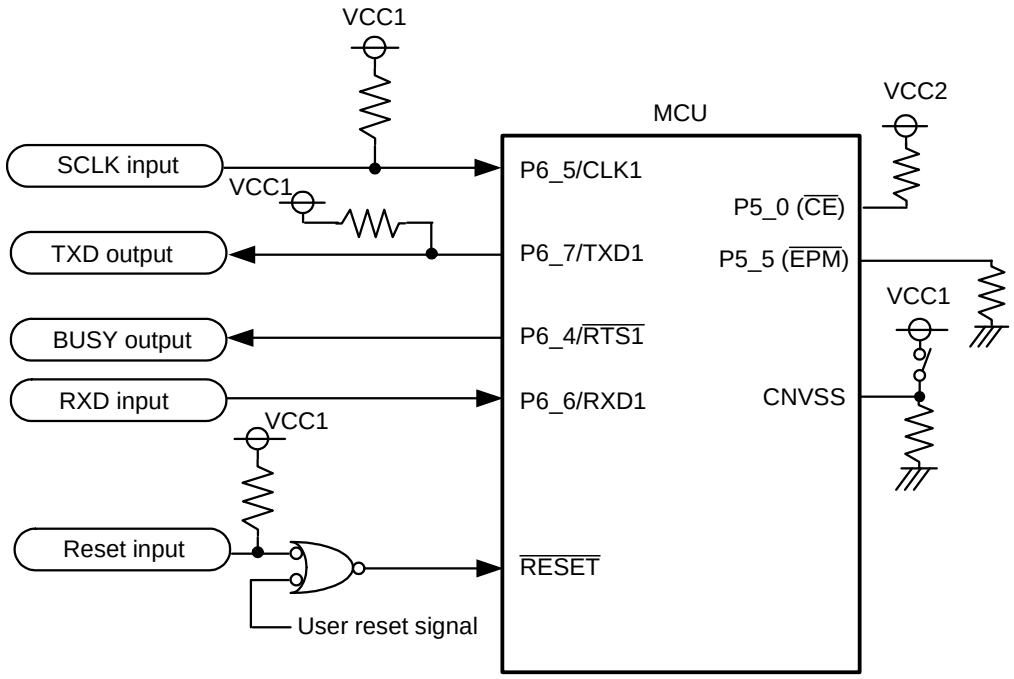
[When a lock bit program operation is executed]

(1) Execute the clear status register command and set the FMR06 bit to 0.
(2) Set the FMR02 bit in the FMR0 register to 1.
(3) Execute the block erase command to erase the block where the error occurred.
(4) Execute the lock bit program command again after writing the data as needed.

If an error still occurs, do not use that block.

Table 28.22 Setting of Standard Serial I/O Mode 1

Signal	Input Level
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS → VCC1
$\overline{\text{CE}}$	VCC2
SCLK	VCC1



Notes:
1. Control pins and external circuitry will vary depending on the programmer. For more information, refer to the programmer manual.
2. In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the CNVSS pin input with a switch.
3. If in standard serial I/O mode 1, there is a possibility that the user reset signal will go low during serial I/O mode, break the connection between the user reset signal and $\overline{\text{RESET}}$ pin by using, for example, a jumper switch.

Figure 28.17 Circuit Application in Standard Serial I/O Mode 1

$V_{CC1} = V_{CC2} = 5\ V$

Timing Requirements

($V_{CC1} = V_{CC2} = 5$ V, $V_{SS} = 0$ V, at T_{opr} = -20°C to 85°C/-40°C to 85°C unless otherwise specified)

29.2.2.3 Timer A Input

Table 29.24 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	100		ns
$t_{w(TAH)}$	TAiIN input high pulse width	40		ns
$t_{w(TAL)}$	TAiIN input low pulse width	40		ns

Table 29.25 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	400		ns
$t_{w(TAH)}$	TAiIN input high pulse width	200		ns
$t_{w(TAL)}$	TAiIN input low pulse width	200		ns

Table 29.26 Timer A Input (External Trigger Input in One-Shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high pulse width	100		ns
$t_{w(TAL)}$	TAiIN input low pulse width	100		ns

Table 29.27 Timer A Input (External Trigger Input in Pulse Width Modulation Mode and Programmable Output Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high pulse width	100		ns
$t_{w(TAL)}$	TAiIN input low pulse width	100		ns

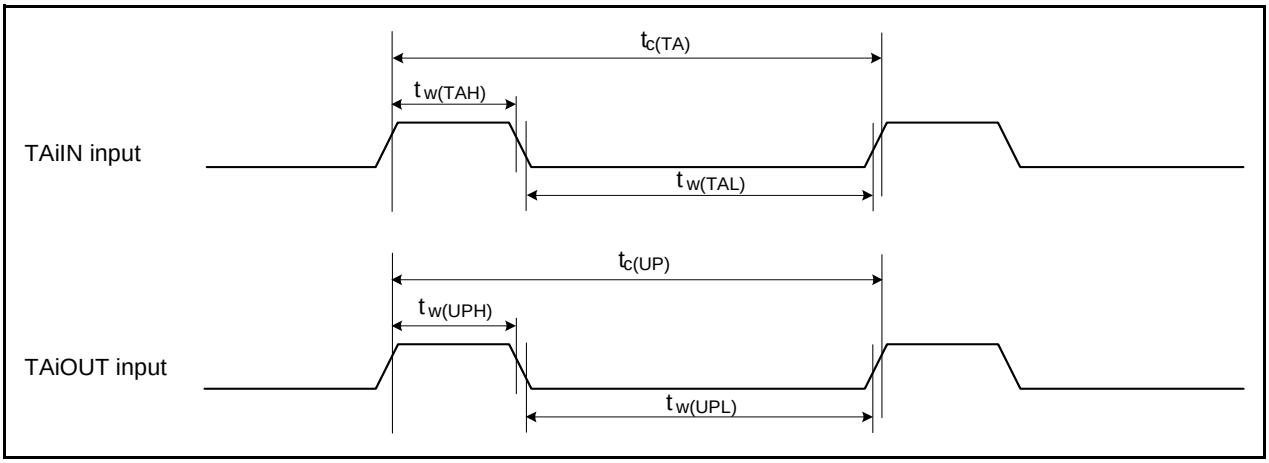


Figure 29.9 Timer A Input

Table 30.2 Registers with Write-Only Bits (2/2)

Address	Register	Symbol
D120h	USB Endpoint 0 IN Data Register	USBEPDR0I
D134h	USB Endpoint 2 Data Register	USBEPDR2
D138h	USB Endpoint 3 Data Register	USBEPDR3
D144h	USB Endpoint 5 Data Register	USBEPDR5
D148h	USB Endpoint 6 Data Register	USBEPDR6
D190h	USB Trigger Register 0	USBTRG0
D191h	USB Trigger Register 1	USBTRG1
D192h	USB Trigger Register 2	USBTRG2
D198h	USB FIFO Clear Register 0	USBFCLR0
D199h	USB FIFO Clear Register 1	USBFCLR1
D19Ah	USB FIFO Clear Register 2	USBFCLR2
D1A0h	USB Endpoint Stall Register 0	USBEPSTL0
D1A1h	USB Endpoint Stall Register 1	USBEPSTL1
D1A2h	USB Endpoint Stall Register 2	USBEPSTL2
D1C0h	USB Endpoint Information Register	USBEPIR

30.22.3.10 Software Command

Observe the notes below when using the following commands.

- Program
- Block erase
- Lock bit program
- Read lock bit status
- Block blank check

(a) The FMR00 bit in the FMR0 register indicates the status while executing these commands. Do not execute other commands while the FMR00 bit is 0 (busy).
(b) Use these commands in 40 MHz on-chip oscillator mode, high-speed mode, medium-speed mode, and PLL operating mode. Do not change clock modes while the FMR00 bit in the FMR0 register is 0 (busy).
(c) After executing the program, block erase, or lock bit program command, perform a full status check per command (Do not execute multiple commands or same command more than once before performing a full status check).
(d) Do not execute the program, block erase, lock bit program, or block blank check command when either or both bits FMR06 and FMR07 in the FMR0 register are 1 (error).
(e) Do not use these commands in slow read mode (when the FMR22 bit is 1) or low current consumption read mode (when both bits FMR22 and FMR23 are 1).

30.22.3.11 PM13 Bit

The PM13 bit in the PM1 register becomes 1 while the FMR01 bit in the FMR0 register is 1 (CPU rewrite mode enabled). The PM13 bit returns to the former value by setting the FMR01 bit to 0 (CPU rewrite mode disabled). When the PM13 bit is changed during CPU rewrite mode, the value of the PM13 bit after being changed is not reflected until the FMR01 bit is set to 0.

30.22.3.12 Area Where the Rewrite Control Program is Executed

Bits PM10 and PM13 in the PM1 register become 1 in CPU rewrite mode. Execute the rewrite program in internal RAM or an external area which can be used when both bits PM10 and PM13 are 1.

30.22.3.13 Program and Erase Cycles and Execution Time

Execution time of the program, block erase, and lock bit program commands becomes longer as the number of programming and erasing increases.

30.22.3.14 Suspending the Auto-Erase and Auto-Program Operations

When the program, block erase, and lock bit program commands are suspended, the blocks for those commands must be erased. Execute the program and lock bit program commands again after erasing.

Those commands are suspended by the following reset or interrupts:

- Hardware, power-on, voltage monitor 0, voltage monitor 1, voltage monitor 2, oscillator stop detect, watchdog timer, software resets.
- $\overline{\text{NMI}}$, watchdog timer, oscillator stop/restart detect, voltage monitor 1, and voltage monitor 2 interrupts.