TOSHIBA

8 Bit Microcontroller

TLCS-870/C1 Series

TMP89CM42

TOSHIBA CORPORATION

Considerations for Using Both Mask ROM and Flash Products

- Flash Memory Control Registers

  Mask ROM products do not contain the flash memory control registers shown in the table below. Therefore, a program that accesses these registers operates differently between mask ROM and flash products. If you use a flash product to check the operation of a program written for a mask ROM product, be sure not to write instructions that access these registers in the program.

Register Name	Address	Mask ROM Product	Flash Product
		89CM42, 89CH42	89FM42, 89FH42
FLSCR1	0x0FD0	Not available	Available
FLSCR2 / FLSCRM	0x0FD1		
FLSSTB	0x0FD2		
SPCR	0x0FD3		

- Conversion Accuracy of the AD Converter

  The conversion accuracy of the AD converter differs between mask ROM and flash products, as shown below. When developing your application system, careful consideration must be given to these accuracy differences.

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = −40 to 85 °C)

Parameter	Condition	Min	Typ.	Max		Unit
				89CM42 89CH42	89FM42 89FH42	
Non-linearity error	V_{DD} = A_{VDD} / V_{AREF} = 5.0 V V_{SS} = 0.0 V	–	–	±4	±3	LSB
Zero-point error		–	–	±4	±3	
Full-scale error		–	–	±4	±3	
Total error		–	–	±4	±3	

(V_{SS} = 0.0 V, 2.7 V ≤ V_{DD} < 4.5 V, Topr = −40 to 85 °C)

Parameter	Condition	Min	Typ.	Max		Unit
				89CM42 89CH42	89FM42 89FH42	
Non-linearity error	V_{DD} = A_{VDD} / V_{AREF} = 2.7 V V_{SS} = 0.0 V	–	–	±4	±3	LSB
Zero-point error		–	–	±4	±3	
Full-scale error		–	–	±4	±3	
Total error		–	–	±4	±3	

(V_{SS} = 0.0 V, 2.2 V ≤ V_{DD} < 2.7 V, Topr = −40 to 85 °C)

Parameter	Condition	Min	Typ.	Max		Unit
				89CM42 89CH42	89FM42 89FH42	
Non-linearity error	V_{DD} = A_{VDD} / V_{AREF} = 2.2 V V_{SS} = 0.0 V	–	–	±5	±4	LSB
Zero-point error		–	–	±5	±4	
Full-scale error		–	–	±5	±4	
Total error		–	–	±5	±4	

Precaution for Using the Emulation Chip (Development Tool)

- Precaution for debugging the voltage detection circuit

  In debug using the RTE870/C1 In-Circuit Emulator (ICE mode) with the TMP89C900 mounted on it, no interrupt is generated when the supply voltage rises to the detection voltage. Since the #!Undefined!# may operate differently, take account of this difference when debugging programs.

  For detail, refer to the chapter of Voltage Detection Circuit.

Revision History

Date	Revision	
2008/2/16	1	First Release
2008/9/4	2	Contents Revised
2009/7/16	3	Contents Revised

Table of Contents

Considerations for Using Both Mask ROM and Flash Products

TMP89CM42

2. CPU Core

6. Power-on Reset Circuit

7. Voltage Detection Circuit

8. I/O Ports

9. Special Function Registers

10. Low Power Consumption Function for Peripherals

11. Divider Output ($\overline{\text{DVO}}$)

12. Time Base Timer (TBT)

13. 16-bit Timer Counter (TCA)

14. 8-bit Timer Counter (TC0)

15. Real Time Clock (RTC)

16. Asynchronous Serial Interface (UART)

17. Synchronous Serial Interface (SIO)

18. Serial Bus Interface (SBI)

19. Key-on Wakeup (KWU)

20. 10-bit AD Converter (ADC)

21. Input/Output Circuit

22. Electrical Characteristics

23. Package Dimensions

CMOS 8-Bit Microcontroller

TMP89CM42

The TMP89CM42 is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 32768 bytes of Mask ROM.

Product No.	ROM (Mask ROM)	RAM	Package	Flash MCU	Emulation Chip
TMP89CM42UG	32768 bytes	2048 bytes	LQFP44-P-1010-0.80B	TMP89FM42UG	* TMP89C900XBG

Note : * ; Under development

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C1 series
   - Instruction execution time :

     100 ns (at 10 MHz)

     122 μs (at 32.768 kHz)
   - 133 types & 732 basic instructions
2. 25 interrupt sources (External : 6 Internal : 19 , Except reset)
3. Input / Output ports (40 pins)

   Note : Two of above pins can not be used for the I/O port, because they should be connected with the high frequency OSC input.
   - Large current output: 8 pins (Typ. 20mA)
4. Watchdog timer
   - Interrupt or reset can be selected by the program.
5. Power-on reset circuit
6. Voltage detection circuit
7. Divider output function
8. Time base timer
9. 16-bit timer counter (TCA) : 2 ch
   - Timer, External trigger, Event Counter, Window, Pulse width measurement, PPG OUTPUT modes
10. 8-bit timer counter (TC0) : 4 ch
    - Timer, Event Counter, PWM, PPG OUTPUT modes
    - Usable as a 16-bit timer, 12-bit PWM output and 16-bit PPG output by the cascade connection of two channels.
11. Real time clock
12. UART : 1ch
13. UART/SIO : 1ch Note : One SIO channel can be used at the same time.
14. I^{2}C/SIO : 1ch
15. Key-on wake-up : 8 ch
16. 10-bit successive approximation type AD converter
    - Analog input : 8ch
17. Clock operation mode control circuit : 2 circuit

    Single clock mode / Dual clock mode
18. Low power consumption operation (8 mode)

- STOP mode:

  Oscillation stops. (Battery/Capacitor back-up.)
- SLOW1 mode:

  Low power consumption operation using low-frequency clock.(High-frequency clock stop.)
- SLOW2 mode:

  Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)
- IDLE0 mode:

  CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Released when the reference time set to TBT has elapsed.
- IDLE1 mode:

  The CPU stops, and peripherals operate using high frequency clock. Release by interruputs(CPU restarts).
- IDLE2 mode:

  CPU stops and peripherals operate using high and low frequency clock. Release by interruputs. (CPU restarts).
- SLEEP0 mode:

  CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock. Released when the reference time set to TBT has elapsed.
- SLEEP1 mode:

  CPU stops, and peripherals operate using low frequency clock. Release by interruput.(CPU restarts).

19. Wide operation voltage:

    4.3 V to 5.5 V at 10MHz /32.768 kHz
    2.7 V to 5.5 V at 4.2 MHz /32.768 kHz
    2.2 V to 5.5 V at 2MHz /32.768 kHz

1.2 Pin Assignment

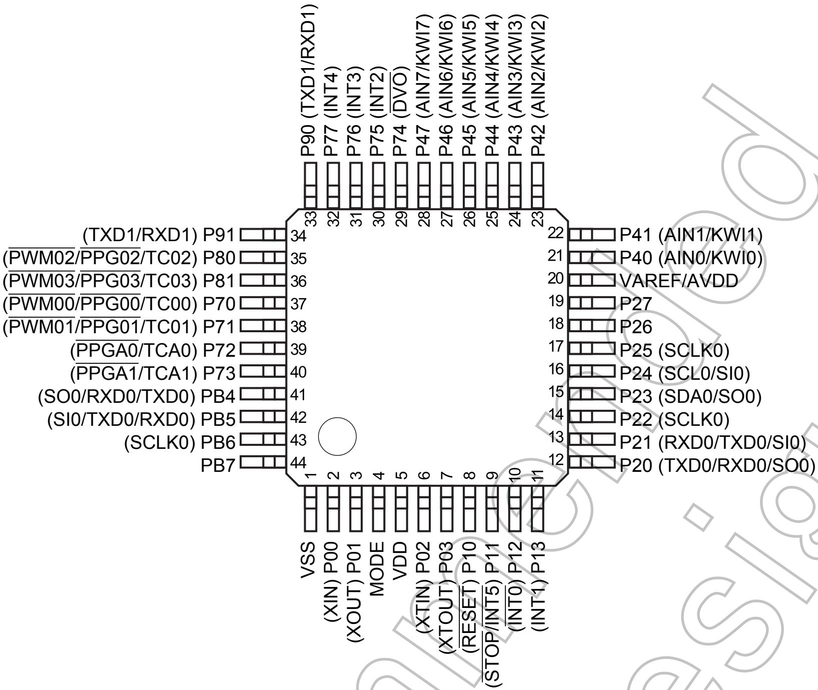


Figure 1-1 Pin Assignment

1.3 Block Diagram

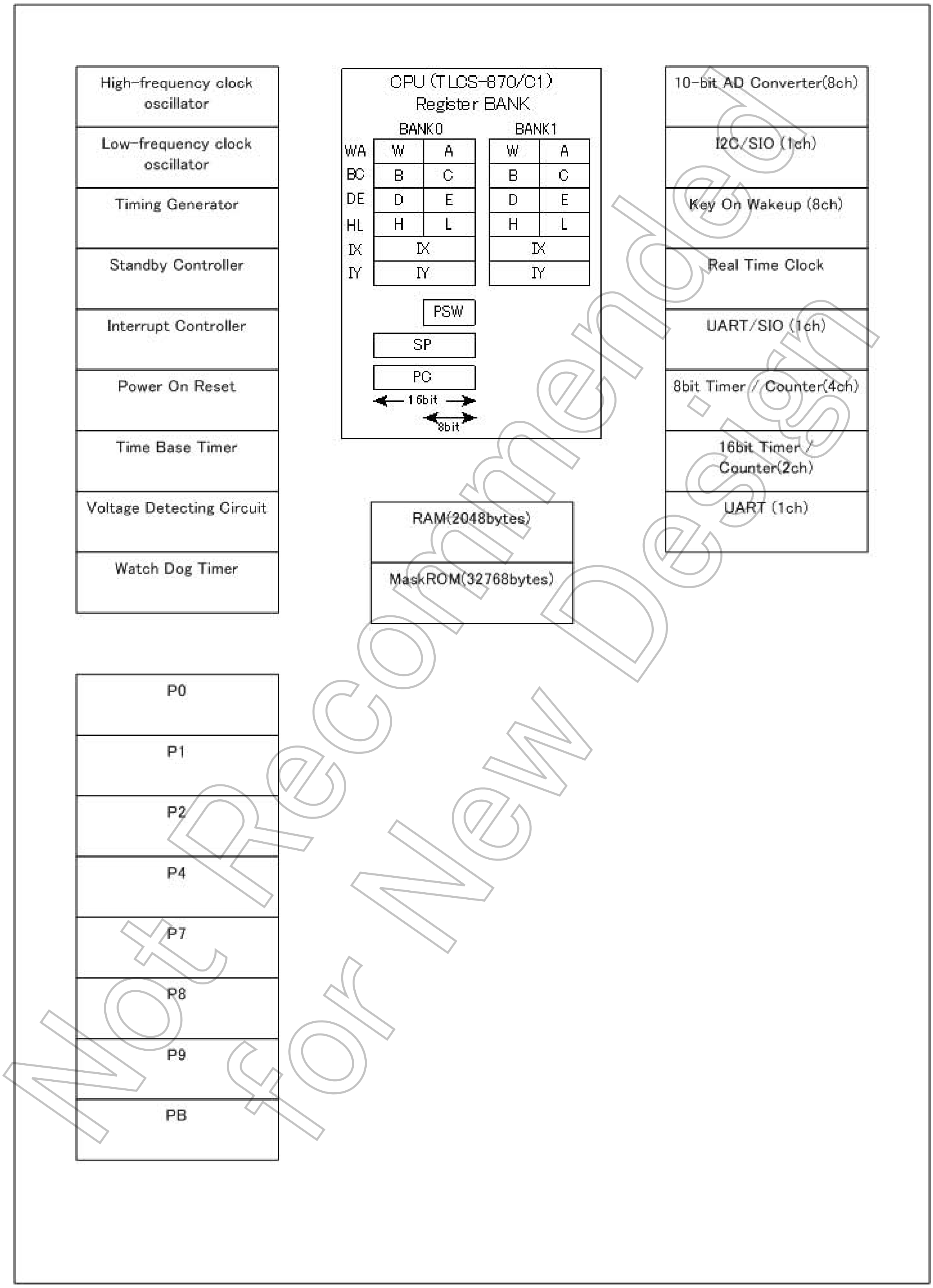


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

Table 1-1 Pin Names and Functions (1/3)

Pin Name	Input/Output	Functions
P03 XTOUT	IO O	PORT03 Low frequency OSC output
P02 XTIN	IO I	PORT02 Low frequency OSC input
P01 XOUT	IO O	PORT01 High frequency OSC output
P00 XIN	IO I	PORT00 High frequency OSC input
P13 INT1	IO I	PORT13 External interrupt 1 input
P12 $\overline{\text{INT0}}$	IO I	PORT12 External interrupt 0 input
P11 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	IO I I	PORT11 External interrupt 5 input STOP mode release input
P10 $\overline{\text{RESET}}$	IO I	PORT10 Reset signal input
P27	IO	PORT27
P26	IO	PORT26
P25 SCLK0	IO IO	PORT25 Serial clock input/output 0
P24 SCL0 SI0	IO IO I	PORT24 I2C bus clock input/output 0 Serial data input 0
P23 SDA0 SO0	IO IO O	PORT23 I2C bus data input/output 0 Serial data output 0
P22 SCLK0	IO IO	PORT22 Serial clock input/output 0
P21 RXD0 TXD0 SI0	IO I O I	PORT21 UART data input 0 UART data output 0 Serial data input 0
P20 TXD0 RXD0 SO0	IO O I O	PORT20 UART data output 0 UART data input 0 Serial data output 0

Table 1-2 Pin Names and Functions (2/3)

Pin Name	Input/Output	Functions
P47 AIN7 KWI7	IO I I	PORT47 Analog input 7 Key-on wake-up input 7
P46 AIN6 KWI6	IO I I	PORT46 Analog input 6 Key-on wake-up input 6
P45 AIN5 KWI5	IO I I	PORT45 Analog input 5 Key-on wake-up input 5
P44 AIN4 KWI4	IO I I	PORT44 Analog input 4 Key-on wake-up input 4
P43 AIN3 KWI3	IO I I	PORT43 Analog input 3 Key-on wake-up input 3
P42 AIN2 KWI2	IO I I	PORT42 Analog input 2 Key-on wake-up input 2
P41 AIN1 KWI1	IO I I	PORT41 Analog input 1 Key-on wake-up input 1
P40 AIN0 KWI0	IO I I	PORT40 Analog input 0 Key-on wake-up input 0
P77 INT4	IO I	PORT77 External interrupt 4 input
P76 INT3	IO I	PORT76 External interrupt 3 input
P75 INT2	IO I	PORT75 External interrupt 2 input
P74 $\overline{\text{DVO}}$	IO O	PORT74 Divider output
P73 TCA1 $\overline{\text{PPGA1}}$	IO I O	PORT73 TCA1 input PPGA1 output
P72 TCA0 $\overline{\text{PPGA0}}$	IO I O	PORT72 TCA0 input PPGA0 output
P71 TC01 $\overline{\text{PPG01}}$ $\overline{\text{PWM01}}$	IO I O O	PORT71 TC01 input PPG01 output PWM01 output

Table 1-2 Pin Names and Functions (3/3)

Pin Name	Input/Output	Functions
P70 TC00 $\overline{PPG00}$ $\overline{PWM00}$	IO I O O	PORT70 TC00 input PPG00 output PWM00 output
P81 TC03 $\overline{PPG03}$ $\overline{PWM03}$	IO I O O	PORT81 TC03 input PPG03 output PWM03 output
P80 TC02 $\overline{PPG02}$ $\overline{PWM02}$	IO I O O	PORT80 TC02 input PPG02 output PWM02 output
P91 RXD1 TXD1	IO I O	PORT91 UART data input 1 UART data output 1
P90 TXD1 RXD1	IO O I	PORT90 UART data output 1 UART data input 1
PB7	IO	PORTB7
PB6 SCLK0	IO IO	PORTB6 Serial clock input/output 0
PB5 RXD0 TXD0 SI0	IO I O I	PORTB5 UART data input 0 UART data output 0 Serial data input 0
PB4 TXD0 RXD0 SO0	IO O I O	PORTB4 UART data output 0 UART data input 0 Serial data output 0
MODE	I	Test pin for out-going test (fix to Low level).
VAREF / AVDD	I	Analog reference voltage input pin for A/D conversion. / Analog power supply pin.
VDD	I	VDD pin
VSS	I	GND pin

2. CPU Core

2.1 Configuration

The CPU core consists of a CPU, a system clock controller and a reset circuit.

This chapter describes the CPU core address space, the system clock controller and the reset circuit.

2.2 Memory space

The 870/C1 CPU memory space consists of a code area to be accessed as instruction operation codes and operands and a data area to be accessed as sources and destinations of transfer and calculation instructions.

Both the code and data areas have independent 64-Kbyte address spaces.

2.2.1 Code area

The code area stores operation codes, operands, vector tables for vector call instructions and interrupt vector tables.

The RAM and the MaskROM are mapped in the code area.

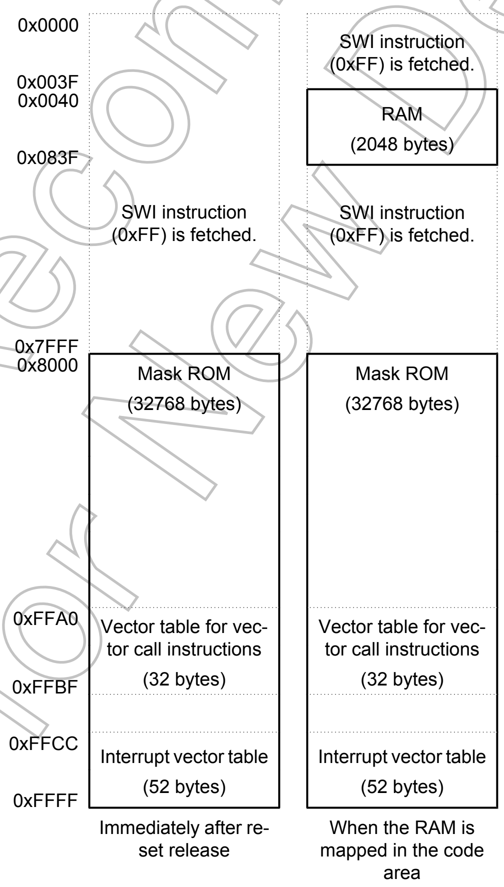


Figure 2-1 Memory Map in the Code Area

2.2.1.1 RAM

The RAM is mapped in the data area immediately after reset release.

By setting SYSCR3<RAREA> to "1" and writing 0xD4 to SYSCR4, RAM can be mapped to 0x0040to 0x083F in the code area to execute the program.

At this time, by setting SYSCR<RVCTR> to "1" and writing 0xD4 to SYSCR4, vector table for vector call instructions and interrupt except reset can be mapped to RAM.

Note 1: When the RAM is not mapped in the code area, the SWI instruction is fetched from 0x0040 to 0x083F.

Note2: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

System control register 3

SYSCR3 (0x0FDE)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	RVCTR	RAREA	(RSTDIS)
	Read/Write	R	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RAREA	Specifies mapping of the RAM in the code area	0 :	The RAM is not mapped from 0x0040 to 0x083F in the code area.	
		1 :	The RAM is mapped from 0x0040 to 0x083F in the code area.	
RVCTR	Specifies mapping of the vector table for vector call instructions and interrupts		Vector table for vector call instructions	Vector table for interrupt
		0 :	0xFFA0 to 0xFFBF in the code area	0xFFCC to 0xFFFF in the code area
		1 :	0x01A0 to 0x01BF in the code area	0x01CC to 0x01FD in the code area

Note 1: The value of SYSCR3<RAREA> is invalid until 0xD4 is written into SYSCR4.

Note 2: To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1" and SYSCR3<RAREA> to "1".

Note 3: Bits 7 to 3 of SYSCR3 are read as "0".

System control register 4

SYSCR4 (0x0FDF)		7	6	5	4	3	2	1	0
	Bit Symbol	SYSCR4							
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the SYSCR3 data control code.	0xB2 :	Enables the contents of SYSCR3<RSTDIS>.
		0xD4 :	Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR>.
		0x71 :	Enables the contents of IRSTSR<FCLR>
			Others : Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 3: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

System control status register 4

SYSSR4 (0x0FDF)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	RVCTRS	RAREAS	(RSTDIS)
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

RAREAS	Status of mapping of the RAM in the code area	0 : 1 :	The enabled SYSCR3<RAREA> data is "0". The enabled SYSCR3<RAREA> data is "1".
RVCTRS	Status of mapping of the vector address in the area	0 : 1 :	The enabled SYSCR3<RVCTR> data is "0". The enabled SYSCR3<RVCTR> data is "1".

Note: Bits 7 to 3 of SYSSR4 are read as "0".

Example: Program transfer (Transfer the program saved in the data area to the RAM.)

```
            LD    HL, TRANSFER_START_ADDRESS   ;Destination RAM address
            LD    DE, PROGRAM_START_ADDRESS    ;Source ROM address
            LD    BC, BYTE_OF_PROGRAM          ;Number of bytes of the program to be executed -1
TRANS_RAM:  LD    A, (DE)                      ;Reading the program to be transferred
            LD    (HL), A                      ;Writing the program to be transferred
            INC   HL                           ;Destination address increment
            INC   DE                           ;Source address increment
            DEC   BC                           ;Have all the programs been transferred?
            J     F, TRANS_RAM
```

2.2.1.2 MaskROM

The MaskROM is mapped to 0x8000 to 0xFFFF in the code area after reset release.

2.2.2 Data area

The data area stores the data to be accessed as sources and destinations of transfer and calculation instructions.

The SFR, the RAM and MaskROM are mapped in the data area.

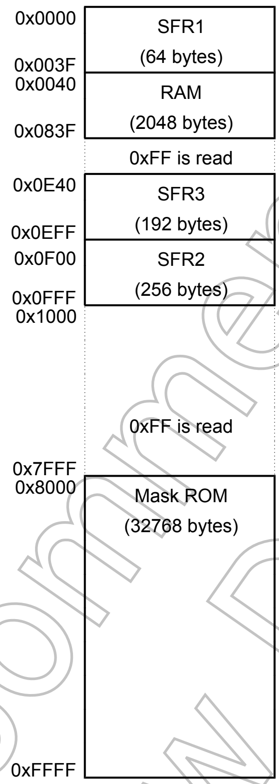


Figure 2-2 Memory Map in the Data Area

2.2.2.1 SFR

The SFR is mapped to 0x0000 to 0x003F (SFR1), 0x0F00 to 0x0FFF (SFR2) and 0x0E40 to 0x0EFF (SFR3) in the data area after reset release.

Note: Don't access the reserved SFR.

2.2.2.2 RAM

The RAM is mapped to 0x0040 to 0x083F in the data area after reset release.

Note: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

Example: RAM initialization program

```
            LD      HL, RAM_TOP_ADDRESS         ;Head of address of the RAM to be initialized
            LD      A, 0x00                     ;Initialization data
            LD      BC, BYTE_OF_CLEAR_BYTES     ;Number of bytes of RAM to be initialized -1
CLR_RAM:    LD      (HL), A                     ;Initialization of the RAM
            INC     HL                          ;Initialization address increment
            DEC     BC                          ;Have all the RAMs been initialized?
            J       F, CLR_RAM
```

2.2.2.3 MaskROM

The MaskROM is mapped to 0x8000 to 0xFFFF in the data area after reset release.

2.3 System clock controller

2.3.1 Configuration

The system clock controller consists of a clock generator, a clock gear, a timing generator, a warm-up counter and an operation mode control circuit.

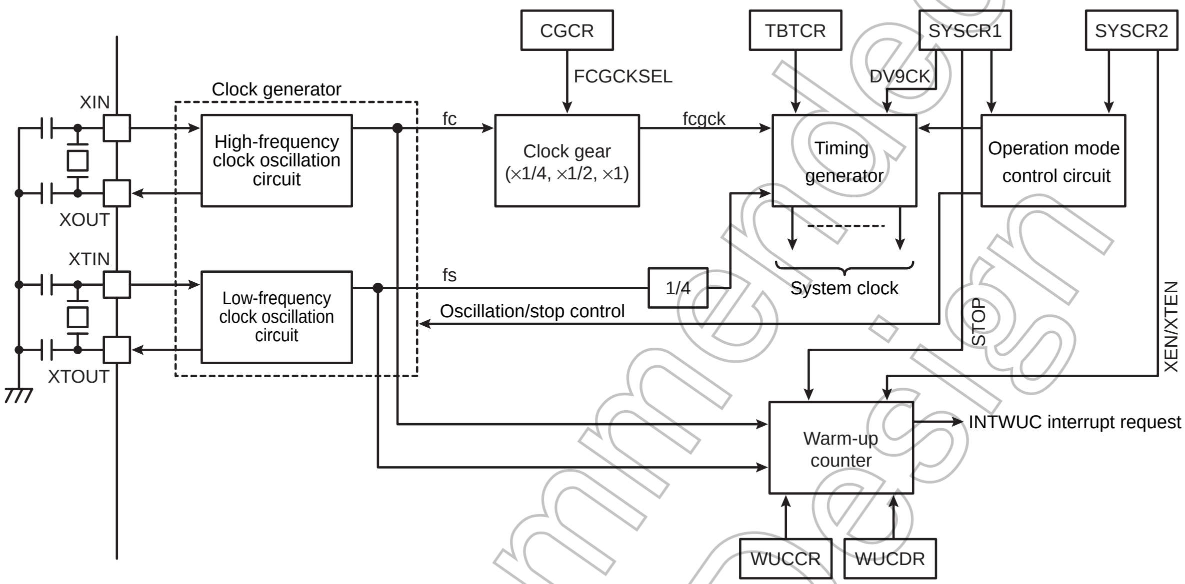


Figure 2-3 System Clock Controller

2.3.2 Control

The system clock controller is controlled by system control register 1 (SYSCR1), system control register 2 (SYSCR2), the warm-up counter control register (WUCCR), the warm-up counter data register (WUCDR) and the clock gear control register (CGCR).

System control register 1

SYSCR1 (0x0FDC)		7	6	5	4	3	2	1	0
	Bit Symbol	STOP	RELM	OUTEN	DV9CK	-	-	-	-
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	1	0	0	0

STOP	Activates the STOP mode	0 :	Operate the CPU and the peripheral circuits
		1 :	Stop the CPU and the peripheral circuits (activate the STOP mode)
RELM	Selects the STOP mode release method	0 :	Edge-sensitive release mode (Release the STOP mode at the rising edge of the STOP mode release signal)
		1 :	Level-sensitive release mode (Release the STOP mode at the "H" level of the STOP mode release signal)
OUTEN	Selects the port output state in the STOP mode	0 :	High impedance
		1 :	Output hold
DV9CK	Selects the input clock to stage 9 of the divider	0 :	fcgck/2^9
		1 :	fs/4

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Bits 2, 1 and 0 of SYSCR1 are read as "0". Bit 3 is read as "1".

Note 3: If the STOP mode is activated with SYSCR1<OUTEN> set at "0", the port internal input is fixed to "0". Therefore, an external interrupt may be set at the falling edge, depending on the pin state when the STOP mode is activated.

Note 4: The P11 pin is also used as the STOP pin. When the STOP mode is activated, the pin reverts to high impedance state and is put in input mode, regardless of the state of SYSCR1<OUTEN>.

Note 5: Writing of the second byte data will be executed improperly if the operation is switched to the STOP state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 6: Don't set SYSCK1<DV9CK> to "1" before the oscillation of the low-frequency clock oscillation circuit becomes stable.

Note 7: In the SLOW1/2 or SLEEP1 mode, fs/4 is input to stage 9 of the divider, regardless of the state of SYSCR1< DV9CK >.

System control register 2

SYSCR2 (0x0FDD)

	7	6	5	4	3	2	1	0
Bit Symbol	-	XEN	XTEN	SYSCK	IDLE	TGHALT	-	-
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R	R
After reset	0	1	0	0	0	0	0	0

XEN	Controls the high-frequency clock oscillation circuit	0 : 1 :	Stop oscillation Continue or start oscillation
XTEN	Controls the low-frequency clock oscillation circuit	0 : 1 :	Stop oscillation Continue or start oscillation
SYSCK	Selects a system clock	0 : 1 :	Gear clock (fcgck) (NORMAL1/2 or IDLE1/2 mode) Low-frequency clock (fs/4) (SLOW1/2 or SLEEP1 mode)
IDLE	CPU and WDT control (IDLE1/2 or SLEEP1 mode)	0 : 1 :	Operate the CPU and the WDT Stop the CPU and the WDT (Activate IDLE1/2 or SLEEP1 mode)
TGHALT	TG control (IDLE0 or SLEEP0 mode)	0 : 1 :	Enable the clock supply from the TG to all the peripheral circuits Disable the clock supply from the TG to the peripheral circuits except the TBT (Activate IDLE0 or SLEEP0 mode)

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: WDT: Watchdog timer, TG: Timing generator

Note 3: Don't set both SYSCR2<IDLE> and SYSCR2<TGHALT> to "1" simultaneously.

Note 4: Writing of the second byte data will be executed improperly if the operation is switched to the IDLE state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 5: When the IDLE1/2 or SLEEP1 mode is released, SYSCR2<IDLE> is cleared to "0" automatically.

Note 6: When the IDLE0 or SLEEP0 mode is released, SYSCR2<TGHALT> is cleared to "0" automatically.

Note 7: Bits 7, 1 and 0 of SYSCR2 are read as "0".

Warm-up counter control register

WUCCR (0x0FCD)

	7	6	5	4	3	2	1	0
Bit Symbol	WUCRST	-	-	-	WUCDIV		WUCSEL	-
Read/Write	W	R	R	R	R/W		R/W	R
After reset	0	0	0	0	1	1	0	1

WUCRST	Resets and stops the warm-up counter	0 : 1 :	- Clear and stop the counter
WUCDIV	Selects the frequency division of the warm-up counter source clock	00 : 01 : 10 : 11 :	Source clock Source clock / 2 Source clock / 2^2 Source clock / 2^3
WUCSEL	Selects the warm-up counter source clock	0 : 1 :	Select the high-frequency clock (fc) Select the low-frequency clock (fs)

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]

Note 2: WUCCR<WUCRST> is cleared to "0" automatically, and need not be cleared to "0" after being set to "1".

Note 3: Bits 7 to 4 of WUCCR are read as "0". Bit 0 is read as "1".

Note 4: Before starting the warm-up counter operation, set the source clock and the frequency division rate at WUCCR and set the warm-up time at WUCDR.

Warm-up counter data register

WUCDR (0x0FCE)

	7	6	5	4	3	2	1	0
Bit Symbol	WUCDR							
Read/Write	R/W							
After reset	0	1	1	0	0	1	1	0

WUCDR	Warm-up time setting

Note 1: Don't start the warm-up counter operation with WUCDR set at "0x00".

Clock gear control register

CGCR (0x0FCF)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	FCGCKSEL	
Read/Write	R	R	R	R	R	R	R/W	
After reset	0	0	0	0	0	0	0	0

FCGCKSEL	Clock gear setting	00 : fcgck = fc / 4 01 : fcgck = fc / 2 10 : fcgck = fc 11 : Reserved

Note 1: fcgck: Gear clock [Hz], fc: High-frequency clock [Hz]

Note 2: Don't change CGCR<FCGCKSEL> in the SLOW mode.

Note 3: Bits 7 to 2 of CGCR are read as "0".

2.3.3 Functions

2.3.3.1 Clock generator

The clock generator generates the basic clock for the system clocks to be supplied to the CPU core and peripheral circuits.

It contains two oscillation circuits: one for the high-frequency clock and the other for the low-frequency clock.

The oscillation circuit pins are also used as ports P0. For the setting to use them as ports, refer to the chapter of I/O Ports.

To use ports P00 and P01 as the high-frequency clock oscillation circuits (the XIN and XOUT pins), set P0FC0 to "1" and then set SYSCR2<XEN> to "1".

To use ports P02 and P03 as the low-frequency clock oscillation circuits (the XTIN and XTOUT pins), set P0FC2 to "1" and then set SYSCR2<XTEN> to "1".

The high-frequency (fc) clock and the low-frequency (fs) clock can easily be obtained by connecting an oscillator between the XIN and XOUT pins and between the XTIN and XTOUT pins respectively.

Clock input from an external oscillator is also possible. In this case, external clocks are applied to the XIN/XTIN pins and the XOUT/XTOUT pins are kept open.

Enabling/disabling the oscillation of the high-frequency clock oscillation circuit and the low-frequency clock oscillation circuit and switching the pin function to ports are controlled by the software and hardware.

The software control is executed by SYSCR2<XEN>, SYSCR2<XTEN> and the P0 port function control register P0FC.

The hardware control is executed by reset release and the operation mode control circuit when the operation is switched to the STOP mode as described in "2.3.5 Operation mode control circuit".

Note: No hardware function is available for external direct monitoring of the basic clock. The oscillation frequency can be adjusted by programming the system to output pulses at a certain frequency to a port (for example, a clock output) with interrupts disabled and the watchdog timer disabled and monitoring the output. An adjustment program must be created in advance for a system that requires adjustment of the oscillation frequency.

To prevent the dead lock of the CPU core due to the software-controlled enabling/disabling of the oscillation, an internal factor reset is generated depending on the combination of values of the clock selected as the main system clock, SYSCR2<XEN>, SYSCR2<XTEN> and the P0 port function control register P0FC0.

Table 2-1 Prohibited Combinations of Oscillation Enable Register Conditions

P0FC0	SYSCR2 <XEN>	SYSCR2 <XTEN>	SYSCR2 <SYSCK>	State
Don't Care	0	0	Don't Care	All the oscillation circuits are stopped.
Don't Care	Don't Care	0	1	The low-frequency clock (fs) is selected as the main system clock, but the low-frequency clock oscillation circuit is stopped.
Don't Care	0	Don't Care	0	The high-frequency clock (fc) is selected as the main system clock, but the high-frequency clock oscillation circuit is stopped.
0	1	Don't Care	Don't Care	The high-frequency clock oscillation circuit is allowed to oscillate, but the port is set as a general-purpose port.

Note: It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6 Operation Mode Control".

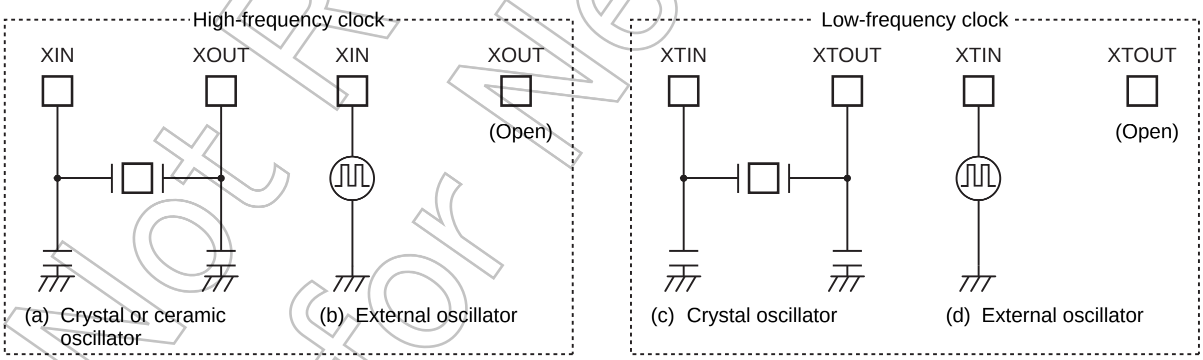


Figure 2-4 Examples of Oscillator Connection

2.3.3.2 Clock gear

The clock gear is a circuit that selects a gear clock (fcgck) obtained by dividing the high-frequency clock (fc) and inputs it to the timing generator.

Selects a divided clock at CGCR<FCGCKSEL>.

Two machine cycles are needed after CGCR<FCGCKSEL> is changed before the gear clock (fcgck) is changed.

The gear clock (fcgck) may be longer than the set clock width, immediately after CGCR<FCGCKSEL> is changed.

Immediately after reset release, the gear clock (fcgck) becomes the clock that is a quarter of the high-frequency clock (fc).

Table 2-2 Gear Clock (fcgck)

CGCR<FCGCKSEL>	fcgck
00	fc / 4
01	fc / 2
10	fc
11	Reserved

Note: Don't change CGCR<FCGCKSEL> in the SLOW mode. This may stop the gear clock (fcgck) from being changed.

2.3.3.3 Timing generator

The timing generator is a circuit that generates system clocks to be supplied to the CPU core and the peripheral circuits, from the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs). The timing generator has the following functions:

1. Generation of the main system clock (fm)
2. Generation of clocks for the timer counter, the time base timer and other peripheral circuits

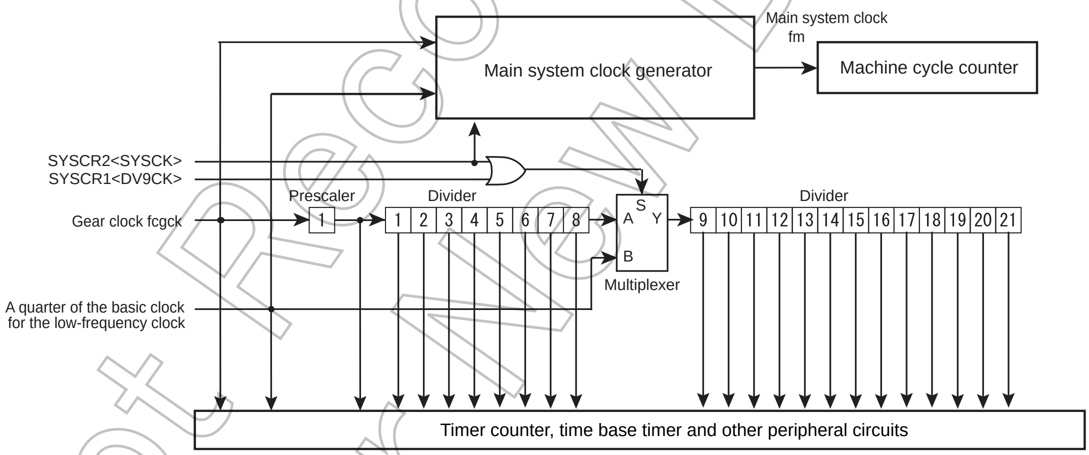


Figure 2-5 Configuration of Timing Generator

(1) Configuration of timing generator

The timing generator consists of a main system clock generator, a prescaler, a 21-stage divider and a machine cycle counter.

1. Main system clock generator

   This circuit selects the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs) for the main system clock (fm) to operate the CPU core.

   Clearing SYSCR2<SYSCK> to "0" selects the gear clock (fcgck). Setting it to "1" selects the clock that is a quarter of the low-frequency clock (fs).

It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6 Operation Mode Control".

2. Prescaler and divider

These circuits divide fcgck. The divided clocks are supplied to the timer counter, the time base timer and other peripheral circuits.

When both SYSCR1<DV9CK> and SYSCR2<SYSCK> are "0", the input clock to stage 9 of the divider becomes the output of stage 8 of the divider.

When SYSCR1<DV9CK> or SYSCR2<SYSCK> is "1", the input clock to stage 9 of the divider becomes fs/4. When SYSCR2<SYSCK> is "1", the outputs of stages 1 to 8 of the divider and prescaler are stopped.

The prescaler and divider are cleared to "0" at a reset and at the end of the warm-up operation that follows the release of STOP mode.

3. Machine cycle

Instruction execution is synchronized with the main system clock (fm).

The minimum instruction execution unit is called a "machine cycle". One machine cycle corresponds to one main system clock.

There are a total of 11 different types of instructions for the TLCS-870/C1 Series: 10 types ranging from 1-cycle instructions, which require one machine cycle for execution, to 10-cycle instructions, which require 10 machine cycles for execution, and 13-cycle instructions, which require 13 machine cycles for execution.

2.3.4 Warm-up counter

The warm-up counter is a circuit that counts the high-frequency clock (fc) and the low-frequency clock (fs), and it consists of a source clock selection circuit, a 3-stage frequency division circuit and a 14-stage counter.

The warm-up counter is used to secure the time after a power-on reset is released before the supply voltage becomes stable and secure the time after the STOP mode is released or the operation mode is changed before the oscillation by the oscillation circuit becomes stable.

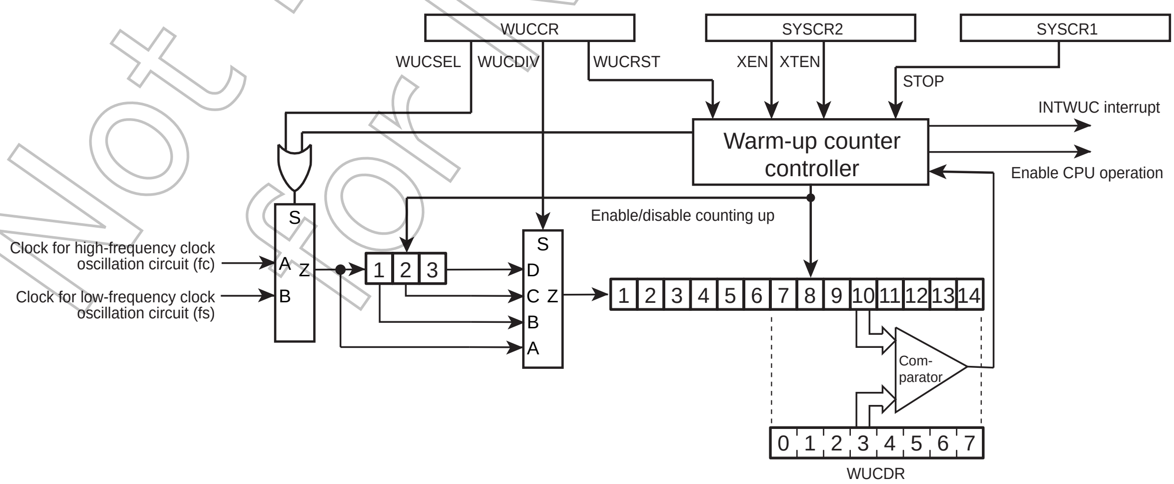


Figure 2-6 Warm-up Counter Circuit

2.3.4.1 Warm-up counter operation when the oscillation is enabled by the hardware

(1) When a power-on reset is released or a reset is released

The warm-up counter serves to secure the time after a power-on reset is released before the supply voltage becomes stable and the time after a reset is released before the oscillation by the high-frequency clock oscillation circuit becomes stable.

When the power is turned on and the supply voltage exceeds the power-on reset release voltage, the warm-up counter reset signal is released. At this time, the CPU and the peripheral circuits are held in the reset state.

A reset signal initializes WUCCR<WUCSEL> to "0" and WUCCR<WUCDIV> to "11", which selects the high-frequency clock (fc) as the input clock to the warm-up counter.

When a reset is released for the warm-up counter, the high-frequency clock (fc) is input to the warm-up counter, and the 14-stage counter starts counting the high-frequency clock (fc).

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and a reset is released for the CPU and the peripheral circuits.

WUCDR is initialized to 0x66 after reset release, which makes the warm-up time 0x66 × 2^9/fc[s].

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable.

(2) When the STOP mode is released

The warm-up counter serves to secure the time after the oscillation is enabled by the hardware before the oscillation becomes stable at the release of the STOP mode.

The high-frequency clock (fc) or the low-frequency clock (fs), which generates the main system clock when the STOP mode is activated, is selected as the input clock for frequency division circuit, regardless of WUCCR<WUCSEL>.

Before the STOP mode is activated, select the division rate of the input clock to the warm-up counter at WUCCR<WUCDIV> and set the warm-up time at WUCDR.

When the STOP mode is released, the 14-stage counter starts counting the input clock selected in the frequency division circuit.

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and the operation is restarted by an instruction that follows the STOP mode activation instruction.

Clock that generates the main system clock when the STOP mode is activated	WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
fc	Don't Care	00	fc	2^6 / fc to 255 x 2^6 / fc
		01	fc / 2	2^7 / fc to 255 x 2^7 / fc
		10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
		11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
fs	Don't Care	00	fs	2^6 / fs to 255 x 2^6 / fs
		01	fs / 2	2^7 / fs to 255 x 2^7 / fs
		10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
		11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note 1: When the operation is switched to the STOP mode during the warm-up for the oscillation enabled by the software, the warm-up counter holds the value at the time, and restarts counting after the STOP mode is released. In this case, the warm-up time at the release of the STOP mode becomes insufficient. Don't switch the operation to the STOP mode during the warm-up for the oscillation enabled by the software.

Note 2: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

2.3.4.2 Warm-up counter operation when the oscillation is enabled by the software

The warm-up counter serves to secure the time after the oscillation is enabled by the software before the oscillation becomes stable, at a mode change from NORMAL1 to NORMAL2 or from SLOW1 to SLOW2.

Select the input clock to the frequency division circuit at WUCCR<WUCSEL>.

Select the input clock to the 14-stage counter at WUCCR<WUCDIV>.

After the warm-up time is set at WUCDR, setting SYSCR2<XEN> or SYSCR2<XTEN> to "1" allows the stopped oscillation circuit to start oscillation and the 14-stage counter to start counting the selected input clock.

When the upper 8 bits of the counter become equal to WUCDR, an INTWUC interrupt occurs, counting is stopped and the counter is cleared.

Set WUCCR<WUCRST> to "1" to discontinue the warm-up operation.

By setting it to "1", the count-up operation is stopped, the warm-up counter is cleared, and WUCCR<WUCRST> is cleared to "0".

SYSCR2<XEN> and SYSCR2<XTEN> hold the values when WUCCR<WUCRST> is set to "1". To restart the warm-up operation, SYSCR2<XEN> or SYSCR2<XTEN> must be cleared to "0".

Note: The warm-up counter starts counting when SYSCR2<XEN> or SYSCR2<XTEN> is changed from "0" to "1". The counter will not start counting by writing "1" to SYSCR2<XEN> or SYSCR2<XTEN> when it is in the state of "1".

WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
0	00	fc	2^6 / fc to 255 x 2^6 / fc
	01	fc / 2	2^7 / fc to 255 x 2^7 / fc
	10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
	11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
1	00	fs	2^6 / fs to 255 x 2^6 / fs
	01	fs / 2	2^7 / fs to 255 x 2^7 / fs
	10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
	11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

2.3.5 Operation mode control circuit

The operation mode control circuit starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock (fm).

There are three operating modes: the single-clock mode, the dual-clock mode and the STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).

Figure 2-7 shows the operating mode transition diagram.

2.3.5.1 Single-clock mode

Only the gear clock (fcgck) is used for the operation in the single-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck). Therefore, the machine cycle time is 1/fcgck [s].

The gear clock (fcgck) is generated from the high-frequency clock (fc).

In the single-clock mode, the low-frequency clock generation circuit pins P02 (XTIN) and P03 (XTOUT) can be used as the I/O ports.

(1) NORMAL1 mode

In this mode, the CPU core and the peripheral circuits operate using the gear clock (fcgck).

The NORMAL1 mode becomes active after reset release.

(2) IDLE1 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck).

The IDLE1 mode is activated by setting SYSCR2<IDLE> to "1" in the NORMAL1 mode.

When the IDLE1 mode is activated, the CPU and the watchdog timer stop.

When the interrupt latch enabled by the interrupt enable register EFR becomes "1", the IDLE1 mode is released to the NORMAL1 mode.

When the IMF (interrupt master enable flag) is "1" (interrupts enabled), the operation returns normal after the interrupt processing is completed.

When the IMF is "0" (interrupts disabled), the operation is restarted by the instruction that follows the IDLE1 mode activation instruction.

(3) IDLE0 mode

In this mode, the CPU and the peripheral circuits stop, except the oscillation circuits and the time base timer.

In the IDLE0 mode, the peripheral circuits stop in the states when the IDLE0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the IDLE0 mode, refer to the section of each peripheral circuit.

The IDLE0 mode is activated by setting SYSCR2<TGHALT> to "1" in the NORMAL1 mode.

When the IDLE0 mode is activated, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

When the falling edge of the source clock selected at TBTCR<TBTCK> is detected, the IDLE0 mode is released, the timing generator starts the clock supply to all the peripheral circuits and the NORMAL1 mode is restored.

Note that the IDLE0 mode is activated and restarted, regardless of the setting of TBTCR<TBTEN>.

When the IDLE0 mode is activated with TBTCR<TBTEN> set at "1", the INTTBT interrupt latch is set after the NORMAL mode is restored.

When the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer) is "1", the operation returns normal after the interrupt processing is completed.

When the IMF is "0" or when the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer) is "0", the operation is restarted by the instruction that follows the IDLE0 mode activation instruction.

2.3.5.2 Dual-clock mode

The gear clock (fcgck) and the low-frequency clock (fs) are used for the operation in the dual-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck) in the NORMAL2 or IDLE2 mode, and generated from the clock that is a quarter of the low-frequency clock (fs) in the SLOW1/2 or SLEEP0/1 mode. Therefore, the machine cycle time is 1/fcgck [s] in the NORMAL2 or IDLE2 mode and is 4/fs [s] in the SLOW1/2 or SLEEP0/1 mode.

P02 (XTIN) and P03 (XTOUT) are used as the low-frequency clock oscillation circuit pins. (These pins cannot be used as I/O ports in the dual-clock mode.)

The operation of the TLCS-870/C1 Series becomes the single-clock mode after reset release. To operate it in the dual-clock mode, allow the low-frequency clock to oscillate at the beginning of the program.

(1) NORMAL2 mode

In this mode, the CPU core operates using the gear clock (fcgck), and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

(2) SLOW2 mode

In this mode, the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<SYSCK> to switch the operation mode from NORMAL2 to SLOW2 or from SLOW2 to NORMAL2.

In the SLOW2 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(3) SLOW1 mode

In this mode, the high-frequency clock oscillation circuit stops operation and the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

This mode requires less power to operate the high-frequency clock oscillation circuit than in the SLOW2 mode.

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<XEN> to switch the operation between the SLOW1 and SLOW2 modes.

In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(4) IDLE2 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

The IDLE2 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the NORMAL2 mode after this mode is released.

(5) SLEEP1 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the CPU and the watchdog timer stop, and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLEEP1 mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP1 mode, refer to the section of each peripheral circuit.

The SLEEP1 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(6) SLEEP0 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the time base timer operates using the clock that is a quarter of the low-frequency clock (fs), and the core and the peripheral circuits stop.

In the SLEEP0 mode, the peripheral circuits stop in the states when the SLEEP0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP0 mode, refer to the section of each peripheral circuit.

The SLEEP0 mode can be activated and released in the same way as for the IDLE0 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLEEP0 mode, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

2.3.5.3 STOP mode

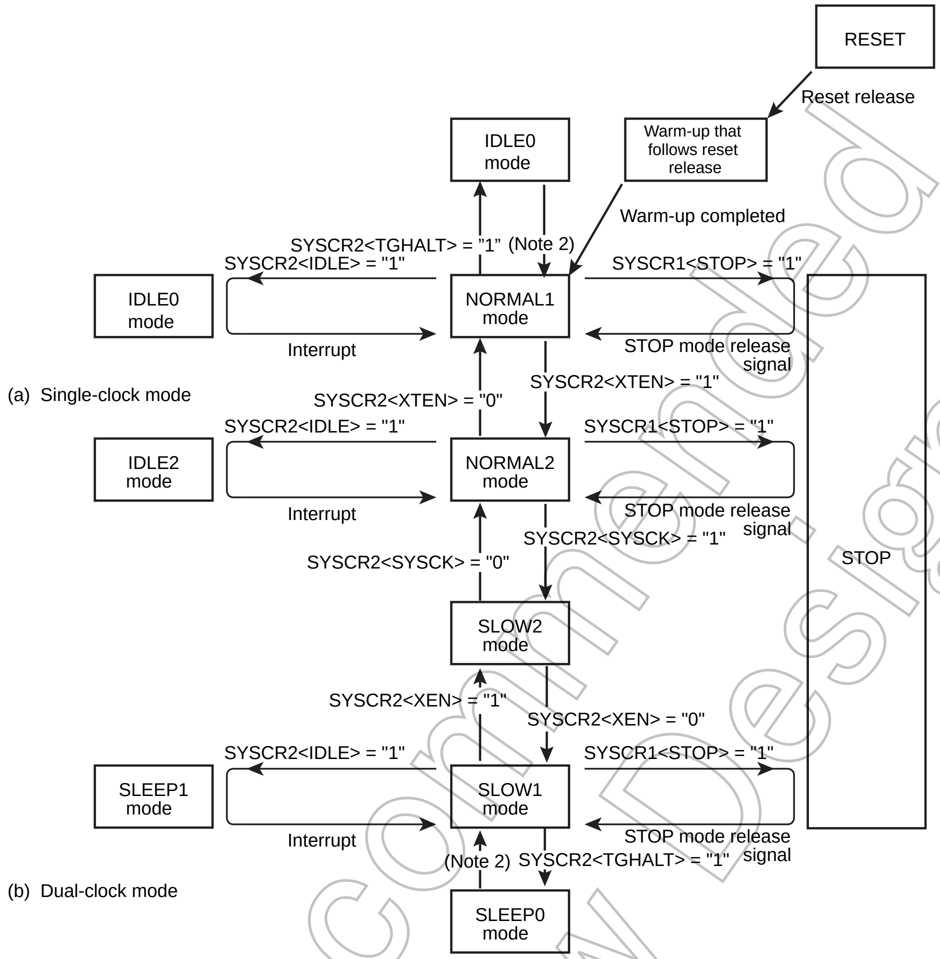
In this mode, all the operations in the system, including the oscillation circuits, are stopped and the internal states in effect before the system was stopped are held with low power consumption.

In the STOP mode, the peripheral circuits stop in the states when the STOP mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the STOP mode, refer to the section of each peripheral circuit.

The STOP mode is activated by setting SYSCR1<STOP> to "1".

The STOP mode is released by the STOP mode release signals. After the warm-up time has elapsed, the operation returns to the mode that was active before the STOP mode, and the operation is restarted by the instruction that follows the STOP mode activation instruction.

2.3.5.4 Transition of operation modes



Note 1: The NORMAL1 and NORMAL2 modes are generically called the NORMAL mode; the SLOW1 and SLOW2 modes are called the SLOW mode; the IDLE0, IDLE1 and IDLE2 modes are called the IDLE mode; and the SLEEP0 and SLEEP1 are called the SLEEP mode.

Note 2: The mode is released by the falling edge of the source clock selected at TBTCR<TBTCK>.

Figure 2-7 Operation Mode Transition Diagram

Table 2-3 Operation Modes and Conditions

<table>
<tr><th colspan="2" rowspan="2">Operation mode</th><th colspan="2">Oscillation circuit</th><th rowspan="2">CPU core</th><th rowspan="2">Watchdog timer</th><th rowspan="2">Time base timer</th><th rowspan="2">AD converter</th><th rowspan="2">Other peripheral circuits</th><th rowspan="2">Machine cycle time</th></tr>
<tr><th>High-frequency</th><th>Low-frequency</th></tr>
<tr><td rowspan="5">Single clock</td><td>RESET</td><td rowspan="4">Oscillation</td><td rowspan="5">Stop</td><td>Reset</td><td>Reset</td><td>Reset</td><td>Reset</td><td>Reset</td><td rowspan="4">1 / fcgck [s]</td></tr>
<tr><td>NORMAL1</td><td>Operate</td><td>Operate</td><td rowspan="3">Operate</td><td rowspan="2">Operate</td><td rowspan="2">Operate</td></tr>
<tr><td>IDLE1</td><td rowspan="3">Stop</td><td rowspan="3">Stop</td></tr>
<tr><td>IDLE0</td><td rowspan="2">Stop</td><td rowspan="2">Stop</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Stop</td><td>–</td></tr>
<tr><td rowspan="7">Dual clock</td><td>NORMAL2</td><td rowspan="3">Oscillation</td><td rowspan="6">Oscillation</td><td>Operate with the high frequency</td><td>Operate with the high / low frequency</td><td rowspan="6">Operate</td><td rowspan="2">Operate</td><td rowspan="5">Operate</td><td rowspan="2">1 / fcgck [s]</td></tr>
<tr><td>IDLE2</td><td>Stop</td><td>Stop</td></tr>
<tr><td>SLOW2</td><td>Operate with the low frequency</td><td>Operate with the low frequency</td><td rowspan="5">Stop</td><td rowspan="4">4/ fs [s]</td></tr>
<tr><td>SLOW1</td><td rowspan="4">Stop</td><td>Operate with the low frequency</td><td>Operate with the low frequency</td></tr>
<tr><td>SLEEP1</td><td rowspan="3">Stop</td><td rowspan="3">Stop</td></tr>
<tr><td>SLEEP0</td><td rowspan="2">Stop</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Stop</td><td>–</td></tr>
</table>

2.3.6 Operation Mode Control

2.3.6.1 STOP mode

The STOP mode is controlled by system control register 1 (SYSCR1) and the STOP mode release signals.

(1) Start the STOP mode

The STOP mode is started by setting SYSCR1<STOP> to "1". In the STOP mode, the following states are maintained:

1. Both the high-frequency and low-frequency clock oscillation circuits stop oscillation and all internal operations are stopped.
2. The data memory, the registers and the program status word are all held in the states in effect before STOP mode was started. The port output latch is determined by the value of SYSCR1<OUTEN>.
3. The prescaler and the divider of the timing generator are cleared to "0".
4. The program counter holds the address of the instruction 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started the STOP mode.

(2) Release the STOP mode

The STOP mode is released by the following STOP mode release signals. It is also released by a reset by the RESET pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

1. Release by the STOP pin
2. Release by key-on wakeup
3. Release by the voltage detection circuits

Note: During the STOP period (from the start of the STOP mode to the end of the warm-up), due to changes in the external interrupt pin signal, interrupt latches may be set to "1" and interrupts may be accepted immediately after the STOP mode is released. Before starting the STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

1. Release by the $\overline{\text{STOP}}$ pin

Release the $\overline{\text{STOP}}$ mode by using the STOP pin.

The STOP mode release by the STOP pin includes the level-sensitive release mode and the edge-sensitive release mode, either of which can be selected at SYSCR1<RELM>.

The $\overline{\text{STOP}}$ pin is also used as the P11 port and the $\overline{\text{INT5}}$ (external interrupt input 5) pin.

- Level-sensitive release mode

The STOP mode is released by setting the $\overline{\text{STOP}}$ pin high.

Setting SYSCR1<RELM> to "1" selects the level-sensitive release mode.

This mode is used for the capacitor backup when the main power supply is cut off and the long term battery backup.

Even if an instruction for starting the STOP mode is executed while the $\overline{\text{STOP}}$ pin input is high, the STOP mode does not start. Thus, to start the STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low.

This can be confirmed by testing the port by the software or using interrupts

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Example: Starting the STOP mode from NORMAL mode after testing P00 port.
(Warm-up time at release of the STOP mode is about 300μs at fc= 10MHz.)

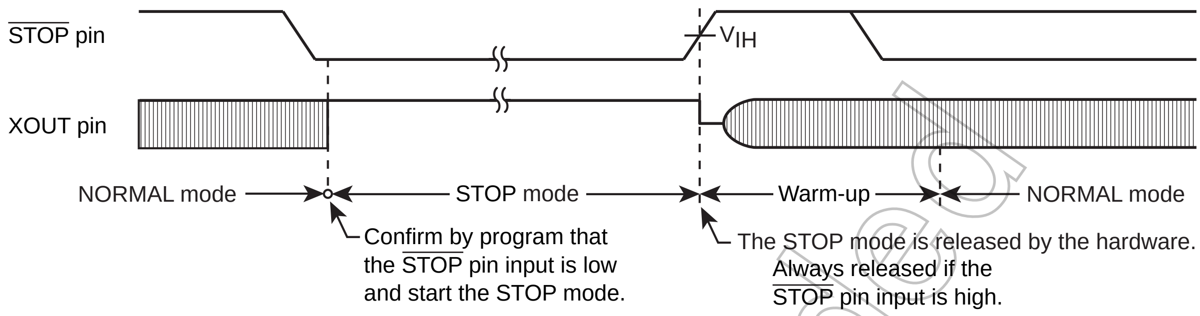
```
            LD      (SYSCR1), 0x40       ;Sets up the level-sensitive release mode
SSTOPH:     TEST    (P0PRD). 5           ;Wait until STOP pin becomes L level.
            J       F, SSTOPH
            LD      (WUCCR), 0x01        ;WUCCR<WUCDIV> = 00 (No division) (Note)
            LD      (WUCDR),0x2F         ;Sets the warm-up time
                                         ;300μs / 6.4μs = 46.9 → round up to 0x2F
            DI                           ;IMF = 0
            SET     (SYSCR1).7           ;Starts the STOP mode
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Example: Starting the STOP mode from the SLOW mode with an INT5 interrupt
(Warm-up time at release of the STOP mode is about 450ms at fs=32.768 kHz.)

```
PINT5:      TEST    (P0PRD).5            ;To reject noise, the STOP mode does not start
            J       F, SINT5             ;if the STOP pin input is high.
            LD      (SYSCR1), 0x40       ;Sets up the level-sensitive release mode
            LD      (WUCCR), 0x03        ;WUCCR<WUCDIV> = 00 (No division) (Note)
            LD      (WUCDR),0xE8         ;Sets the warm-up time
                                         ;450 ms/1.953 ms = 230.4 → round up to 0xE8
            DI                           ;IMF = 0
            SET     (SYSCR1).7           ;Starts the STOP mode
SINT5:      RETI
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.



Note: Even if the $\overline{\text{STOP}}$ pin input returns to low after the warm-up starts, the STOP mode is not restarted.

Figure 2-8 Level-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

- Edge-sensitive release mode

In this mode, the STOP mode is released at the rising edge of the $\overline{\text{STOP}}$ pin input.

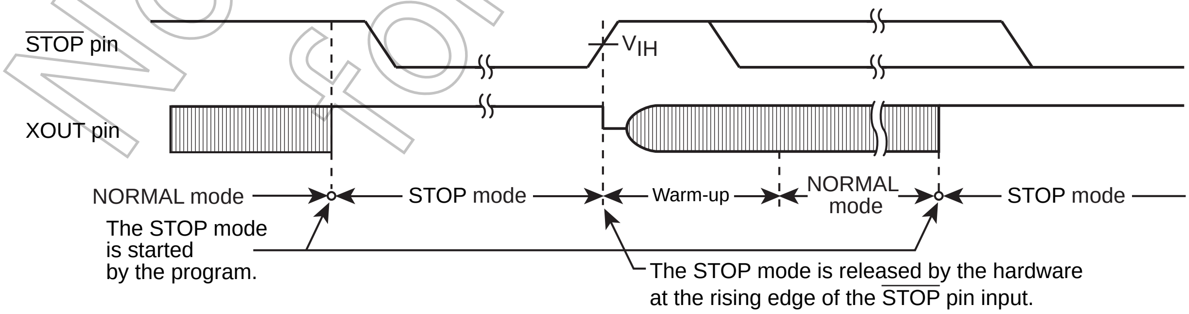
Setting SYSCR1<RELM> to "0" selects the edge-sensitive release mode.

This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{\text{STOP}}$ pin. In the edge-sensitive release mode, the STOP mode is started even when the $\overline{\text{STOP}}$ pin input is high

Example: Starting the STOP mode from the NORMAL mode
(Warm-up time at release of the STOP mode is about 200μs at fc=10 MHz.)

```
LD    (WUCCR),0x01        ;WUCCR<WUCDIV> = 00 (No division) (Note)
LD    (WUCDR),0x20        ;Sets the warm-up time
                          ;200μs / 6.4μs = 31.25 → round up to 0x20
DI                        ;IMF = 0
LD    (SYSCR1) , 0x80     ;Starts the STOP mode with the edge-sensitive release mode selected
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.



Note: If the rising edge is input to the $\overline{\text{STOP}}$ pin within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

Figure 2-9 Edge-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

2. Release by the key-on wakeup

The STOP mode is released by inputting the prescribed level to the key-on wakeup pin.

The level to release the STOP mode can be selected from "H" and "L".

For release by the key-on wakeup, refer to section "Key-on Wakeup".

Note: If the key-on wakeup pin input becomes the opposite level to the release level after the warm-up starts, the STOP mode is not restarted.

3. Release by the voltage detection circuits

   The STOP mode is released by the supply voltage detection by the voltage detection circuits.

   If the voltage detection operation mode of the voltage detection circuits is set to "Generates a voltage detection reset signal", the STOP mode is released and a reset is applied as soon as the supply voltage becomes lower than the detection voltage.

   When the supply voltage becomes equal to or higher than the detection voltage of the voltage detection circuits, the reset is released and the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

   For details, refer to the section of the voltage detection circuits.

   Note: If the supply voltage becomes equal to or higher than the detection voltage within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

(3) STOP mode release operation

The STOP mode is released in the following sequence:

1. Oscillation starts. For the oscillation start operation in each mode, refer to "Table 2-4 Oscillation Start Operation at Release of the STOP Mode".
2. Warm-up is executed to secure the time required to stabilize oscillation. The internal operations remain stopped during warm-up. The warm-up time is set by the warm-up counter, depending on the oscillator characteristics.
3. After the warm-up time has elapsed, the normal operation is restarted by the instruction that follows the STOP mode start instruction. At this time, the prescaler and the divider of the timing generator are cleared to "0".

Note: When the STOP mode is released with a low hold voltage, the following cautions must be observed.

The supply voltage must be at the operating voltage level before releasing the STOP mode. The $\overline{\text{RESET}}$ pin input must also be "H" level, rising together with the supply voltage. In this case, if an external time constant circuit has been connected, the $\overline{\text{RESET}}$ pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if the input voltage level of the $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-4 Oscillation Start Operation at Release of the STOP Mode

Operation mode before the STOP mode is started		High-frequency clock	Low-frequency clock	Oscillation start operation after release
Single-clock mode	NORMAL1	High-frequency clock oscillation circuit	-	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit stops oscillation.
Dual-clock mode	NORMAL2	High-frequency clock oscillation circuit	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit starts oscillation.
	SLOW1	-	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit stops oscillation. The low-frequency clock oscillation circuit starts oscillation.

Note: When the operation returns to the NORMAL2 mode, fc is input to the frequency division circuit of the warm-up counter.

2.3.6.2 IDLE1/2 and SLEEP1 modes

The IDLE1/2 and SLEEP1 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following states are maintained during these modes.

1. The CPU and the watchdog timer stop their operations. The peripheral circuits continue to operate.
2. The data memory, the registers, the program status word and the port output latches are all held in the status in effect before IDLE1/2 or SLEEP1 mode was started.
3. The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE1/2 or SLEEP1 mode.

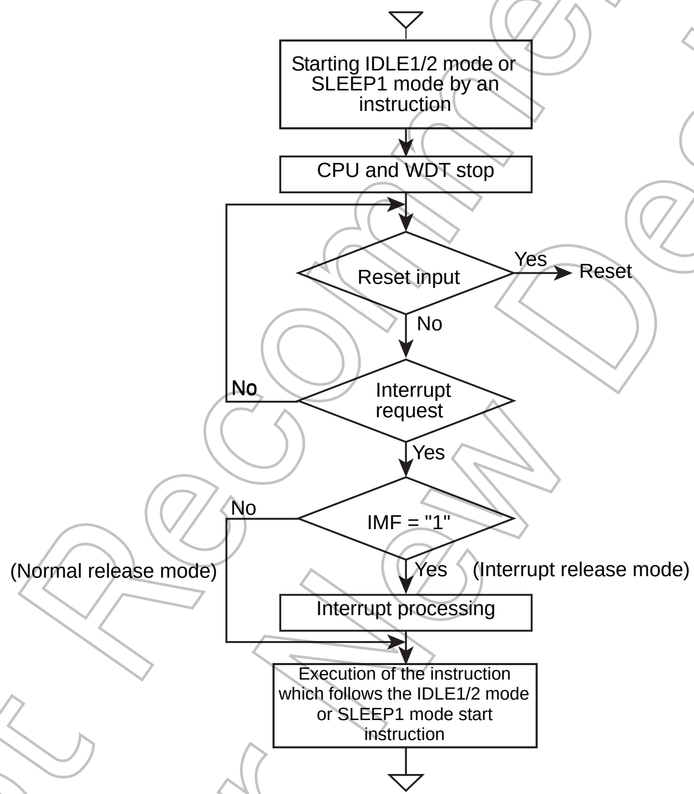


Figure 2-10 IDLE1/2 and SLEEP 1 Modes

(1) Start the IDLE1/2 and SLEEP1 modes

After the interrupt master enable flag (IMF) is set to "0", set the individual interrupt enable flag (EF) to "1", which releases IDLE1/2 and SLEEP1 modes.

To start the IDLE1/2 or SLEEP1 mode, set SYSCR2<IDLE> to "1".

If the release condition is satisfied when it is attempted to start the IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> remains cleared and the IDLE1/2 or SLEEP1 mode will not be started.

Note 1: When a watchdog timer interrupt is generated immediately before the IDLE1/2 or SLEEP1 mode is started, the watchdog timer interrupt will be processed but the IDLE1/2 or SLEEP1 mode will not be started.

Note 2: Before starting the IDLE1/2 or SLEEP1 mode, enable the interrupt request signals to be generated to release the IDLE1/2 or SLEEP1 mode and set the individual interrupt enable flag.

(2) Release the IDLE1/2 and SLEEP1 modes

The IDLE1/2 and SLEEP1 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF). After releasing IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE1/2 or SLEEP1 mode.

The IDLE1/2 and SLEEP1 modes are also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. After releasing the reset, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

- Normal release mode (IMF = "0")

  The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". The operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction. Normally, the interrupt latch (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

- Interrupt release mode (IMF = "1")

  The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". After the interrupt is processed, the operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction.

2.3.6.3 IDLE0 and SLEEP0 modes

The IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR). The following states are maintained during the IDLE0 and SLEEP0 modes:

- The timing generator stops the clock supply to the peripheral circuits except the time base timer.
- The data memory, the registers, the program status word and the port output latches are all held in the states in effect before the IDLE0 or SLEEP0 mode was started.
- The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE0 or SLEEP0 mode.

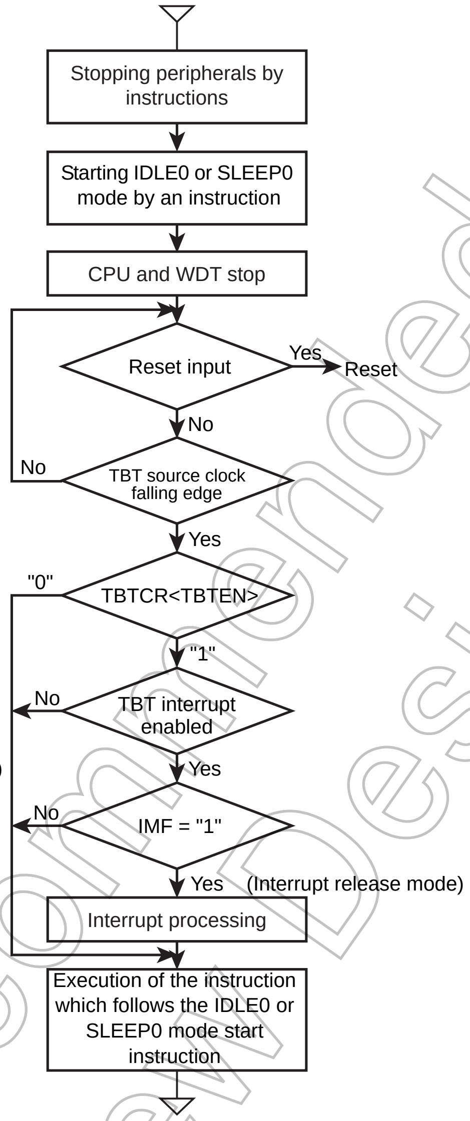


Figure 2-11 IDLE0 and SLEEP0 Modes

- Start the IDLE0 and SLEEP0 modes

  Stop (disable) the peripherals such as a timer counter.

  To start the IDLE0 or SLEEP0 mode, set SYSCR2<TGHALT> to "1".

- Release the IDLE0 and SLEEP0 modes

  The IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF), the individual interrupt enable flag (EF5) for the time base timer and TBTCR<TBTEN>. After releasing the IDLE0 or SLEEP0 mode, SYSCR2<TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE0 or SLEEP0 mode. If TBTCR<TBTEN> has been set at "1", the INTTBT interrupt latch is set.

  The IDLE0 and SLEEP0 modes are also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

(1) Normal release mode (IMF, EF5, TBTCR<TBTEN> = "0")

The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK> is detected. After the IDLE0 or SLEEP0 mode is released, the operation is restarted by the instruction that follows the IDLE0 or SLEEP0 mode start instruction.

When TBTCR<TBTEN> is "1", the time base timer interrupt latch is set.

(2) Interrupt release mode (IMF, EF5, TBTCR<TBTEN> = "1")

The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK> is detected. After the release, the INTTBT interrupt processing is started.

Note 1: The IDLE0 or SLEEP0 mode is released to the NORMAL1 or SLOW1 mode by the asynchronous internal clock selected at TBTCR<TBTCK>. Therefore, the period from the start to the release of the mode may be shorter than the time specified at TBTCR<TBTCK>.

Note 2: When a watchdog timer interrupt is generated immediately before the IDLE0 or SLEEP0 mode is started, the watchdog timer interrupt will be processed but the IDLE0 or SLEEP0 mode will not be started.

2.3.6.4 SLOW mode

The SLOW mode is controlled by system control register 2 (SYSCR2).

(1) Switching from the NORMAL2 mode to the SLOW1 mode

Set SYSCR2<SYSCK> to "1".

When a maximum of 2/fcgck + 10/fs [s] has elapsed since SYSCR2<SYSCK> is set to "1", the main system clock (fm) is switched to fs/4.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XEN> to "0" to turn off the high-frequency clock oscillator.

If the oscillation of the low-frequency clock (fs) is unstable, confirm the stable oscillation at the warm-up counter before implementing the procedure described above.

Note 1: Be sure to follow this procedure to switch the operation from the NORMAL2 mode to the SLOW1 mode.

Note 2: It is also possible to allow the basic clock for the high-frequency clock to oscillate continuously to return to NORMAL2 mode. However, be sure to turn off the oscillation of the basic clock for the high-frequency clock when the STOP mode is started from the SLOW mode.

Note 3: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 4: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 10/fs or shorter.

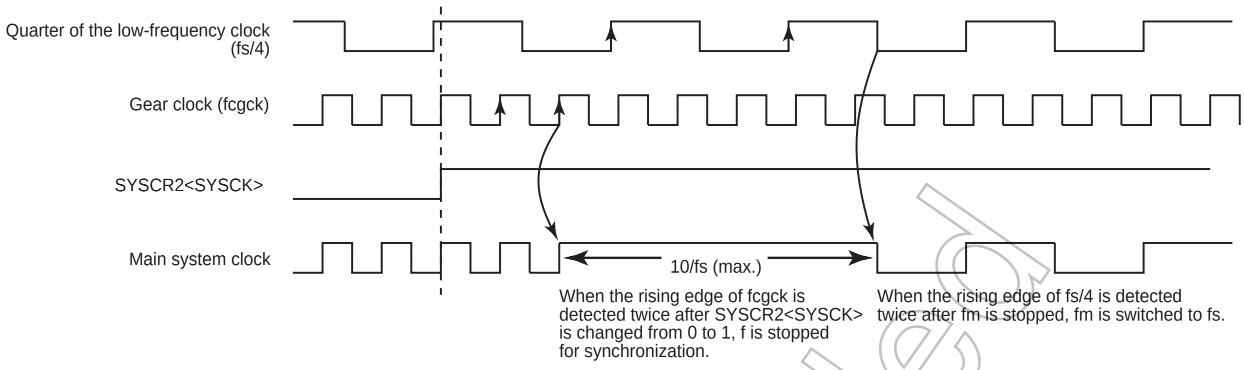


Figure 2-12 Switching of the Main System Clock (fm) (Switching from fcgck to fs/4)

Example 1: Switching from the NORMAL2 mode to the SLOW1 mode (when fc is used as the basic clock for the high-frequency clock)

```
        SET     (SYSCR2).4              ;SYSCR2<SYSCK> = 1
                                        ;(Switches the main system clock to the basic clock for the
                                        ;low-frequency clock for the SLOW2 mode)
        NOP                             ;Waits for 2 machine cycles
        NOP
        CLR     (SYSCR2).6              ;SYSCR2<XEN> = 0
                                        ;(Turns off the high-frequency clock oscillation circuit)
```

Example 2: Switching to the SLOW1 mode after the stable oscillation of the low-frequency clock oscillation circuit is confirmed at the warm-up counter (fs=32.768kHz, warm-up time = about 100 ms)

```
        ; #### Initialize routine ####
        SET     (P0FC).2                ;P0FC2 = 1 (Uses P02/03 as oscillators)
        ¦
        ¦
        LD      (WUCCR), 0x02           ;WUCCR<WUCDIV> = 00 (No division)
                                        ;WUCCR<WUCSEL> = 1 (Selects fs as the source clock)
        LD      (WUCDR), 0x33           ;Sets the warm-up time
                                        ;(Determines the time depending on the oscillator characteristics)
                                        ;100 ms/1.95 ms = 51.2 → round up to 0x33
        SET     (EIRL).4                ;Enables INTWUC interrupts
        SET     (SYSCR2).5              ;SYSCR2<XTEN> = 1
                                        ;(Starts the low-frequency clock oscillation and starts the warm-up
                                        ;counter)
        ¦
        ; #### Interrupt service routine of warm-up counter interrupts ####
PINTWUC: SET    (SYSCR2).4              ;SYSCR2<SYSCK> = 1
                                        ;(Switches the main system clock to the low-frequency clock)
        NOP                             ;Waits for 2 machine cycles
        NOP
        CLR     (SYSCR2).6              ;SYSCR2<XEN> = 0
                                        ;(Turns off the high-frequency clock oscillation circuit)
        RETI
        ¦
```

```
VINTWUC:    DW    PINTWUC                         ;INTWUC vector table
```

(2) Switching from the SLOW1 mode to the NORMAL1 mode

Set SYSCR2<XEN> to "1" to enable the high-frequency clock (fc) to oscillate. Confirm at the warm-up counter that the oscillation of the basic clock for the high-frequency clock has stabilized, and then clear SYSCR2<SYSCK> to "0".

When a maximum of 8/fs + 2.5/fcgck [s] has elapsed since SYSCR2<SYSCK> is cleared to "0", the main system clock (fm) is switched to fcgck.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XTEN> to "0" to turn off the low-frequency clock oscillator.

The SLOW mode is also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

Note 1: Be sure to follow this procedure to switch the operation from the SLOW1 mode to the NORMAL1 mode.

Note 2: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XTEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 3: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 2.5/fcgck [s] or shorter.

Note 4: When P0FC0 is "0", setting SYSCR2<XEN> to "1" causes a system clock reset.

Note 5: When SYSCR2<XEN> is set at "1", writing "1" to SYSCR2<XEN> does not cause the warm-up counter to start counting the source clock.

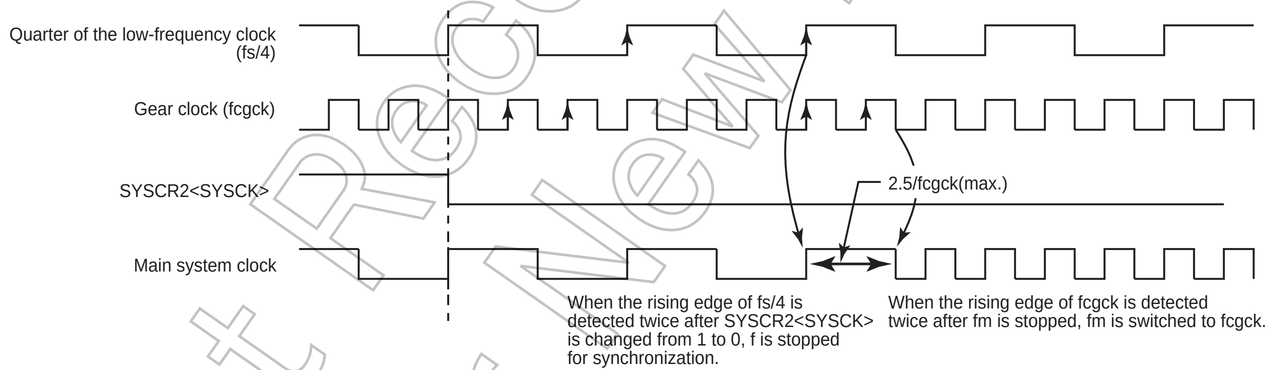


Figure 2-13 Switching the Main System Clock (fm) (Switching from fs/4 to fcgck)

Example : Switching from the SLOW1 mode to the NORMAL1 mode after the stability of the high-frequency clock oscillation circuit is confirmed at the warm-up counter (fc = 10 MHz, warm-up time = 4.0 ms)

```
; #### Initialize routine ####
SET     (P0FC).2                ;P0FC2 = 1 (Uses P02/03 as oscillators)
¦
¦
LD      (WUCCR), 0x09           ;WUCCR<WUCDIV> = 10 (Divided by 2)
                                ;WUCCR<WUCSEL> = 0 (Selects fc as the source clock)
LD      (WUCDR), 0x9D           ;Sets the warm-up time
                                ;(Determine the time depending on the frequency and the oscillator
                                ;characteristics)
                                ;4ms / 25.6us = 156.25 → round up to 0x9D
```

```
                SET     (EIRL). 4               ;Enables INTWUC interrupts
                SET     (SYSCR2) .6             ;SYSCR2<XEN> = 1
                                                ;(Starts the oscillation of the high-frequency clock oscillation circuit)
                ¦
                ; #### Interrupt service routine of warm-up counter interrupts ####
PINTWUC:        CLR     (SYSCR2). 4             ;SYSCR2<SYSCK> = 0
                                                ;(Switches the main system clock to the gear clock)
                NOP                             ;Waits for 2 machine cycles
                NOP
                CLR     (SYSCR2). 5             ;SYSCR2<XTEN> = 0
                                                ;(Turns off the low-frequency clock oscillation circuit)
                RETI
                ¦
VINTWUC:        DW      PINTWUC                 ;INTWUC vector table
```

2.4 Reset Control Circuit

The reset circuit controls the external and internal factor resets and initializes the system.

2.4.1 Configuration

The reset control circuit consists of the following reset signal generation circuits:

1. External reset input (external factor)
2. Power-on reset (internal factor)
3. Voltage detection reset 1 (internal factor)
4. Voltage detection reset 2 (internal factor)
5. Watchdog timer reset (internal factor)
6. System clock reset (internal factor)
7. Trimming data reset (internal factor)

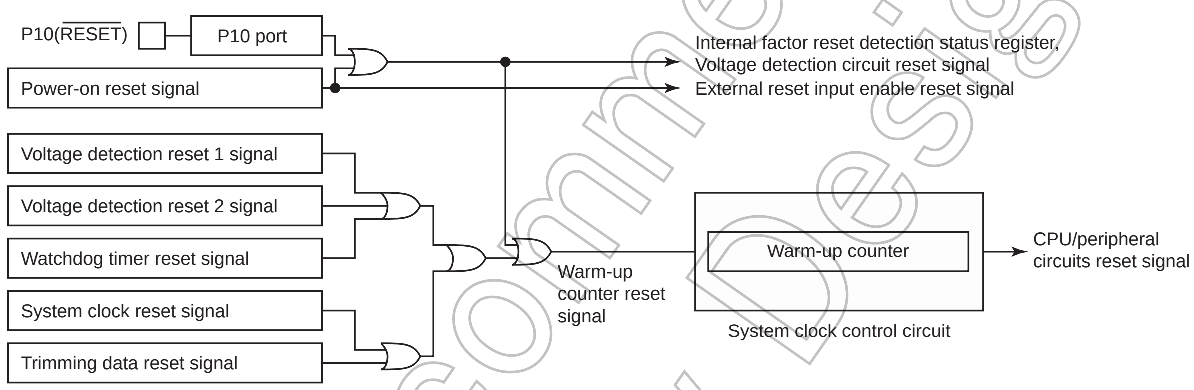


Figure 2-14 Reset Control Circuit

2.4.2 Control

The reset control circuit is controlled by system control register 3 (SYSCR3), system control register 4 (SYSCR4), system control status register (SYSSR4) and the internal factor reset detection status register (IRSTSR).

System control register 3

SYSCR3 (0x0FDE)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	(RVCTR)	(RAREA)	RSTDIS
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RSTDIS	External reset input enable register	0 : 1 :	Enables the external reset input. Disables the external reset input.

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by an external reset input or internal factor reset. The value written in SYSCR3 is reset by a power-on reset, external reset input or internal factor reset.

Note 2: The value of SYSCR3<RSTDIS> is invalid until 0xB2 is written into SYSCR4.

Note 3: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 4: Bits 7 to 3 of SYSCR3 are read as "0".

System control register 4

SYSCR4 (0x0FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	SYSCR4							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the SYSCR3 data control code.	0xB2 : 0xD4 : 0x71 : Others :	Enables the contents of SYSCR3<RSTDIS> Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR> Enables the contents of IRSTSR<FCLR> Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 3: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

System control status register 4

SYSSR4 (0x0FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	(RVCTRS)	(RAREAS)	RSTDISS
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

RSTDISS	External reset input enable status	0 : 1 :	The enabled SYSCR3<RSTDIS> data is "0". The enabled SYSCR3<RSTDIS> data is "1".

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by any other reset signals. The value written in SYSCR3 is reset by a power-on reset and other reset signals.

Note 2: Bits 7 to 3 of SYSCR4 are read as "0".

Internal factor reset detection status register

IRSTSR (0x0FCC)		7	6	5	4	3	2	1	0
	Bit Symbol	FCLR	-	TRMDS	TRMRF	LVD2RF	LVD1RF	SYSRF	WDTRF
	Read/Write	W	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

FCLR	Flag initialization control	0 : 1 :	- Clears the internal factor reset flag to "0".
TRMDS	Trimming data status	0 : 1 :	- Detect state of abnormal trimming data
TRMRF	Trimming data reset detection flag	0 : 1 :	- Detects the trimming data reset.
LVD2RF	Voltage detection reset 2 detection flag	0 : 1 :	- Detects the voltage detection 2 reset.
LVD1RF	Voltage detection reset 1 detection flag	0 : 1 :	- Detects the voltage detection 1 reset.
SYSRF	System clock reset detection flag	0 : 1 :	- Detects the system clock reset.
WDTRF	Watchdog timer reset detection flag	0 : 1 :	- Detects the watchdog timer reset.

Note 1: Internal reset factor flag (IRSTSR<TRMDS, TRMRF, LVD2RF, LVD1RF, SYSRF, WDTRF>) is initialized only by a power-on reset, an external reset input or IRSTSR <FCLR>. It is not initialized by an internal factor reset.

Note 2: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 3: If SYSCR4 is set to 0x71 after IRSTSR<FCLR> is set to "1", internal factor reset flag is cleared to "0" and IRSTSR<FCLR> is automatically cleared to "0".

Note 4: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

Note 5: Bit 7, 6 of IRSTSR is read as "0".

2.4.3 Functions

The power-on reset, external reset input and internal factor reset signals are input to the warm-up circuit of the clock generator.

During reset, the warm-up counter circuit is reset, and the CPU and the peripheral circuits are reset.

After reset is released, the warm-up counter starts counting the high frequency clock (fc), and executes the warm-up operation that follows reset release.

During the warm-up operation that follows reset release, the trimming data is loaded from the non-volatile exclusive use memory for adjustment of the ladder resistor that generates the comparison voltage for the power-on reset and the voltage detection circuits.

When the warm-up operation that follows reset release is finished, the CPU starts execution of the program from the reset vector address stored in addresses 0xFFFE to 0xFFFF.

When a reset signal is input during the warm-up operation that follows reset release, the warm-up counter circuit is reset.

The reset operation is common to the power-on reset, external reset input and internal factor resets, except for the initialization of some special function registers and the initialization of the voltage detection circuits.

When a reset is applied, the peripheral circuits become the states as shown in Table 2-5.

Table 2-5 Initialization of Built-in Hardware by Reset Operation and Its Status after Release

Built-in hardware	During reset	During the warm-up operation that follows reset release	Immediately after the warm-up operation that follows reset release
Program counter (PC)	0xFFFE	0xFFFE	0xFFFE
Stack pointer (SP)	0x00FF	0x00FF	0x00FF
RAM	Indeterminate	Indeterminate	Indeterminate
General-purpose registers (W, A, B, C, D, E, H, L, IX and IY)	Indeterminate	Indeterminate	Indeterminate
Register bank selector (RBS)	0	0	0
Jump status flag (JF)	Indeterminate	Indeterminate	Indeterminate
Zero flag (ZF)	Indeterminate	Indeterminate	Indeterminate
Carry flag (CF)	Indeterminate	Indeterminate	Indeterminate
Half carry flag (HF)	Indeterminate	Indeterminate	Indeterminate
Sign flag (SF)	Indeterminate	Indeterminate	Indeterminate
Overflow flag (VF)	Indeterminate	Indeterminate	Indeterminate
Interrupt master enable flag (IMF)	0	0	0
Individual interrupt enable flag (EF)	0	0	0
Interrupt latch (IL)	0	0	0
High-frequency clock oscillation circuit	Oscillation enabled	Oscillation enabled	Oscillation enabled
Low-frequency clock oscillation circuit	Oscillation disabled	Oscillation disabled	Oscillation disabled
Warm-up counter	Reset	Start	Stop
Timing generator prescaler and divider	0	0	0
Watchdog timer	Disabled	Disabled	Enabled
Voltage detection circuit	Disabled or enabled	Disabled or enabled	Disabled or enabled
I/O port pin status	HiZ	HiZ	HiZ
Special function register	Refer to the SFR map.	Refer to the SFR map.	Refer to the SFR map.

Note: The voltage detection circuits are disabled by an external reset input or power-on reset only.

2.4.4 Reset Signal Generating Factors

Reset signals are generated by each factor as follows:

2.4.4.1 Power-on reset

The power-on reset is an internal reset that occurs when power is turned on.

During power-up, a power-on reset signal is generated while the supply voltage is below the power-on reset release voltage. When the supply voltage rises above the power-on reset release voltage, the power-on reset signal is released.

During power-down, a power-on reset signal is generated when the supply voltage falls below the power-on reset detection voltage.

Refer to "Power-on Reset circuit".

2.4.4.2 External reset input ($\overline{\text{RESET}}$ pin input)

This is an external reset that is generated by the $\overline{\text{RESET}}$ pin input. Port P10 is also used as the $\overline{\text{RESET}}$ pin, and it is configured as the $\overline{\text{RESET}}$ pin at power-up.

- During power-up
  - When the supply voltage rises rapidly

    When the power supply rise time (t_{VDD}) is shorter than 5 [ms] with enough margin, the reset can be released by a power-on reset or an external reset ($\overline{\text{RESET}}$ pin input).

    The power-on reset logic and external reset ($\overline{\text{RESET}}$ pin input) logic are ORed. This means that the TMP89CM42 is reset when either or both of these reset sources are asserted.

    Therefore, the reset time is determined by the reset source with a longer reset period. If the $\overline{\text{RESET}}$ pin level changes from Low to High before the supply voltage rises above the power-on-reset release voltage (V_{PROFF}) (or if the $\overline{\text{RESET}}$ pin level is High from the beginning), the reset time depends on the power-on reset. If the $\overline{\text{RESET}}$ pin level changes from Low to High after the supply voltage rises above V_{PROFF}, the reset time depends on the external reset.

    In the former case, a warm-up period begins when the power-on reset signal is released. In the latter case, a warm-up period begins when the $\overline{\text{RESET}}$ pin level becomes High. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-15).

  - When the supply voltage rises slowly

    When the power supply rise time (t_{VDD}) is longer than 5 [ms], the reset must be released by using the $\overline{\text{RESET}}$ pin. In this case, hold the $\overline{\text{RESET}}$ pin Low until the supply voltage rises to the operating voltage range and oscillation is stabilized. When this state is achieved, wait at least 5 [µs] and then pull the $\overline{\text{RESET}}$ pin High. Changing the $\overline{\text{RESET}}$ pin level to High starts a warm-up period. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-15).

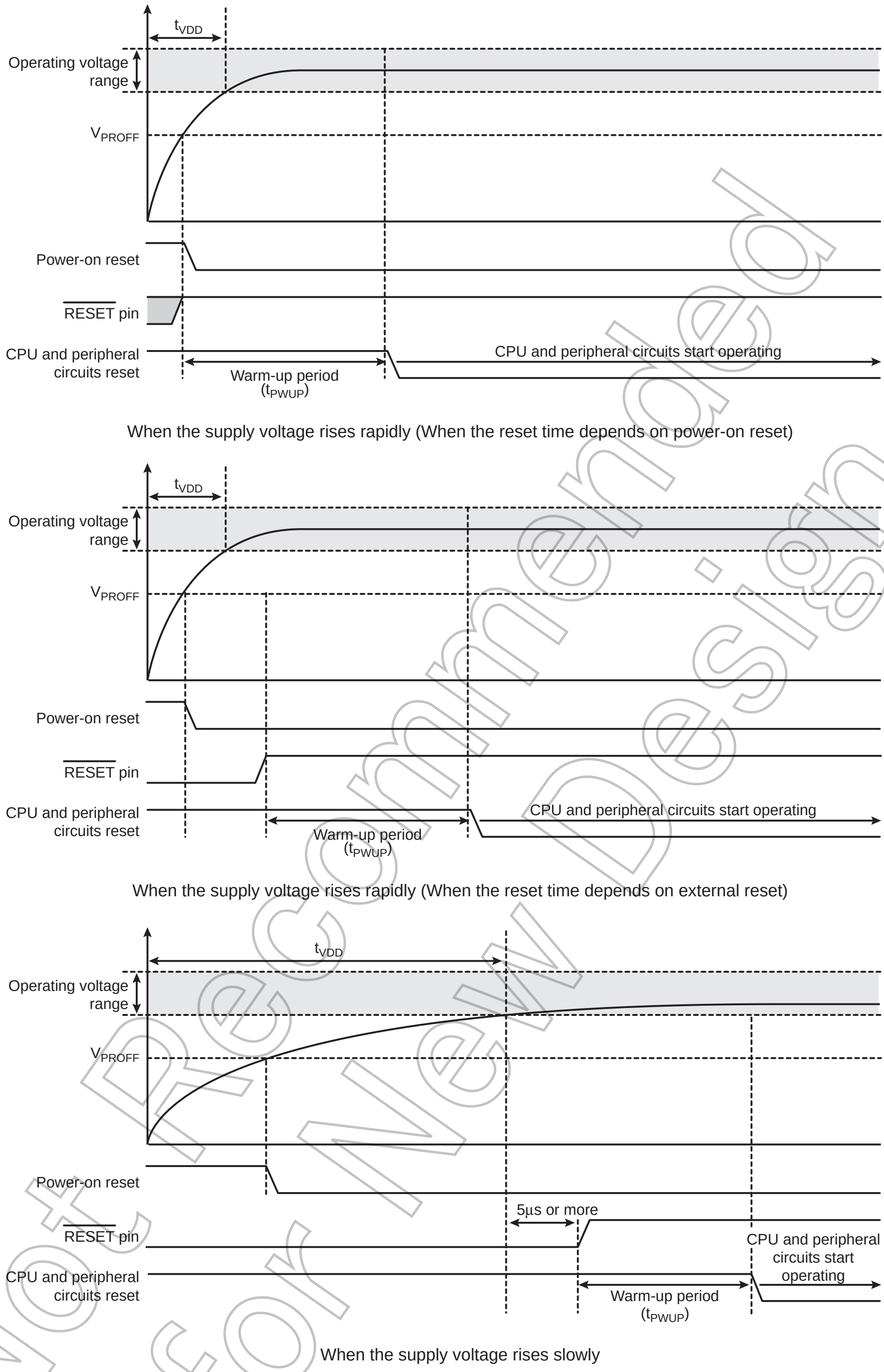


Figure 2-15 External Reset Input (During Power-Up)

・ When the supply voltage is within the operating voltage range

When the supply voltage is within the operating voltage range and stable oscillation is achieved, holding the $\overline{\text{RESET}}$ pin Low for 5 [µs] or longer generates a reset. Then, changing the $\overline{\text{RESET}}$ pin level to High starts a warm-up period. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-16).

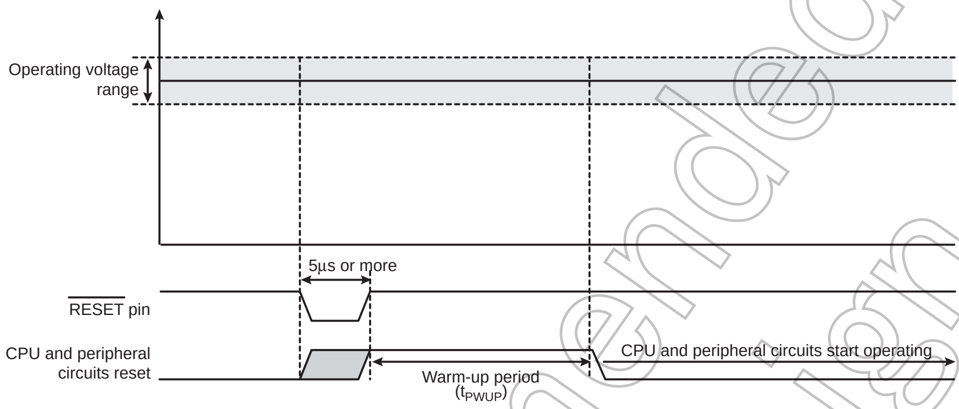


Figure 2-16 External Reset Input (When the Power Supply Is Stable)

2.4.4.3 Voltage detection reset

The voltage detection reset is an internal factor reset that occurs when it is detected that the supply voltage has reached a predetermined detection voltage.

Refer to "Voltage Detection Circuit".

2.4.4.4 Watchdog timer reset

The watchdog timer reset is an internal factor reset that occurs when an overflow of the watchdog timer is detected.

Refer to "Watchdog Timer".

2.4.4.5 System clock reset

The system clock reset is an internal factor reset that occurs when it is detected that the oscillation enable register is set to a combination that puts the CPU into deadlock.

Refer to "Clock Control Circuit".

2.4.4.6 Trimming data reset

The trimming data reset is an internal factor reset that occurs when the trimming data latched in the internal circuit is broken down during operation due to noise or other factors.

The trimming data is a data bit provided for adjustment of the ladder resistor that generates the comparison voltage for the power-on reset and the voltage detection circuits.

This bit is loaded from the non-volatile exclusive use memory during the warm-up time that follows reset release (tPWUP) and latched into the internal circuit.

If the trimming data loaded from the non-volatile exclusive use memory during the warm-up operation that follows reset release is abnormal, IRSTSR<TRMDS> is set to "1".

When IRSTSR<TRMDS> is read as "1" in the initialize routine immediately after reset release, the trimming data need to be reloaded by generating an internal factor reset, such as a system clock reset, and activating the warm-up operation again.

If IRSTSR<TRMDS> is still set to "1" after repeated reading, the detection voltage of the voltage detection circuit and power-on reset circuit does not satisfy the characteristic specified in the electric characteristics. Design the system so that the system will not be damaged in such a case.

2.4.4.7 Internal factor reset detection status register

By reading the internal factor reset detection status register IRSTSR after the release of an internal factor reset, except the power-on reset, the factor which causes a reset can be detected.

The internal factor reset detection status register is initialized by an external reset input or power-on reset.

Set IRSTSR<FCLR> to "1" and write 0x71 to SYSCR4. This enables IRSTSR<FCLR> and the internal factor reset detection status register is clear to "0". IRSTSR<FCLR> is cleared to "0" automatically after initializing the internal factor reset detection status register.

Note 1: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 2: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

2.4.4.8 How to use the external reset input pin as a port

To use the external reset input pin as a port, keep the external reset input pin at the "H" level until the power is turned on and the warm-up operation that follows reset release is finished.

After the warm-up operation that follows reset release is finished, set P1PU0 to "1" and P1CR0 to "0", and connect a pull-up resistor for a port. Then set SYSCR3<RSTDIS> to "1" and write 0xB2 to SYSCR4. This disables the external reset function and makes the external reset input pin usable as a normal port.

To use the pin as an external reset pin when it is used as a port, set P1PU0 to "1" and P1CR0 to "0" and connect the pull-up resistor to put the pin to the input mode. Then clear SYSCR3<RSTDIS> to "0" and write 0xB2 to SYSCR4. This enables the external reset function and makes the pin usable as the external reset input pin.

Note 1: If you switch the external reset input pin to a port or switch the pin used as a port to the external reset input pin, do it when the pin is stabilized at the "H" level. Switching the pin function when the "L" level is input may cause a reset.

Note 2: If the external reset input is used as a port, the statement which clears SYSCR3<RSTDIS> to "0" is not written in a program. By the abnormal execution of program, the external reset input set as a port may be changed as the external reset input at unexpected timing.

Note 3: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

2.5 Revision History

Rev	Description
RA002	"Table 2-3 Operation Modes and Conditions" Added AD converter condition.
	"(2) Release the STOP mode" Added new example program and note to Level-sensitive release mode.
RA003	"Table 2-3 Operation Modes and Conditions" Revised character code error. "Table 2-3 Operation Modes and Conditions" Added AD converter condition. "(2) Release the STOP mode" Added new example program.
RA004	"2.3.6 Operation Mode Control" Revised register name from VDCR2<VDSS> to VDCR2<SRSS>. " Internal factor reset detection status register" Revised Note. "2.4.4.2 External reset input (RESET pin input)" Revised description.
RB000	Revised P03 (XTIN) and P04 (XTOUT) to P02 (XTIN) and P03 (XTOUT). Deleted SRSS function.

3. Interrupt Control Circuit

The TMP89CM42 has a total of 25 interrupt sources excluding reset. Interrupts can be nested with priorities. Three of the internal interrupt sources are non-maskable while the rest are maskable.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and have independent vector addresses. When a request for an interrupt is generated, its interrupt latch is set to "1", which requests the CPU to accept the interrupt. Acceptance of interrupts is enabled or disabled by software using the interrupt master enable flag (IMF) and individual enable flag (EF) for each interrupt source. If multiple maskable interrupts are generated simultaneously, the interrupts are accepted in order of descending priority. The priorities are determined by the interrupt priority change control register (ILPRS1-ILPRS6) as Levels and determined by the hardware as the basic priorities.

However, there are no prioritized interrupt sources among non-maskable interrupts.

Interrupt sources		Enable condition	Interrupt latch	Vector Address (MCU mode)		Basic priority
				RVCTR=0 enabled	RVCTR=1 enabled	
Internal/External	(Reset)	Non-maskable	-	0xFFFE	-	1
Internal	INTSWI	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTUNDEF	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTWDT	Non-maskable	ILL<IL3>	0xFFF8	0x01F8	2
Internal	INTWUC	IMF AND EIRL<EF4> = 1	ILL<IL4>	0xFFF6	0x01F6	5
Internal	INTTBT	IMF AND EIRL<EF5> = 1	ILL<IL5>	0xFFF4	0x01F4	6
Internal	INTRXD0 / INTSIO0	IMF AND EIRL<EF6> = 1	ILL<IL6>	0xFFF2	0x01F2	7
Internal	INTTXD0	IMF AND EIRL<EF7> = 1	ILL<IL7>	0xFFF0	0x01F0	8
External	$\overline{\text{INT5}}$	IMF AND EIRH<EF8> = 1	ILH<IL8>	0xFFEE	0x01EE	9
Internal	INTVLTD	IMF AND EIRH<EF9> = 1	ILH<IL9>	0xFFEC	0x01EC	10
Internal	INTADC	IMF AND EIRH<EF10> = 1	ILH<IL10>	0xFFEA	0x01EA	11
Internal	INTRTC	IMF AND EIRH<EF11> = 1	ILH<IL11>	0xFFE8	0x01E8	12
Internal	INTTC00	IMF AND EIRH<EF12> = 1	ILH<IL12>	0xFFE6	0x01E6	13
Internal	INTTC01	IMF AND EIRH<EF13> = 1	ILH<IL13>	0xFFE4	0x01E4	14
Internal	INTTCA0	IMF AND EIRH<EF14> = 1	ILH<IL14>	0xFFE2	0x01E2	15
Internal	INTSBI0/INTSIO0	IMF AND EIRH<EF15> = 1	ILH<IL15>	0xFFE0	0x01E0	16
External	INT0	IMF AND EIRE<EF16> = 1	ILE<IL16>	0xFFDE	0x01DE	17
External	INT1	IMF AND EIRE<EF17> = 1	ILE<IL17>	0xFFDC	0x01DC	18
External	INT2	IMF AND EIRE<EF18> = 1	ILE<IL18>	0xFFDA	0x01DA	19
External	INT3	IMF AND EIRE<EF19> = 1	ILE<IL19>	0xFFD8	0x01D8	20
External	INT4	IMF AND EIRE<EF20> = 1	ILE<IL20>	0xFFD6	0x01D6	21
Internal	INTTCA1	IMF AND EIRE<EF21> = 1	ILE<IL21>	0xFFD4	0x01D4	22
Internal	INTRXD1	IMF AND EIRE<EF22> = 1	ILE<IL22>	0xFFD2	0x01D2	23
Internal	INTTXD1	IMF AND EIRE<EF23> = 1	ILE<IL23>	0xFFD0	0x01D0	24
Internal	INTTC02	IMF AND EIRD<EF24> = 1	ILD<IL24>	0xFFCE	0x01CE	25
Internal	INTTC03	IMF AND EIRD<EF25> = 1	ILD<IL25>	0xFFCC	0x01CC	26
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Note 1: To use the watchdog timer interrupt (INTWDT), clear WDCTR<WDTOUT> to "0" (It is set for the "Reset request" after reset is released). For details, see "Watchdog Timer".

Note 2: Vector address areas can be changed by the SYSCR3<RVCTR> setting. To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1" and SYSCR3<RAREA> to "1".

3.1 Configuration

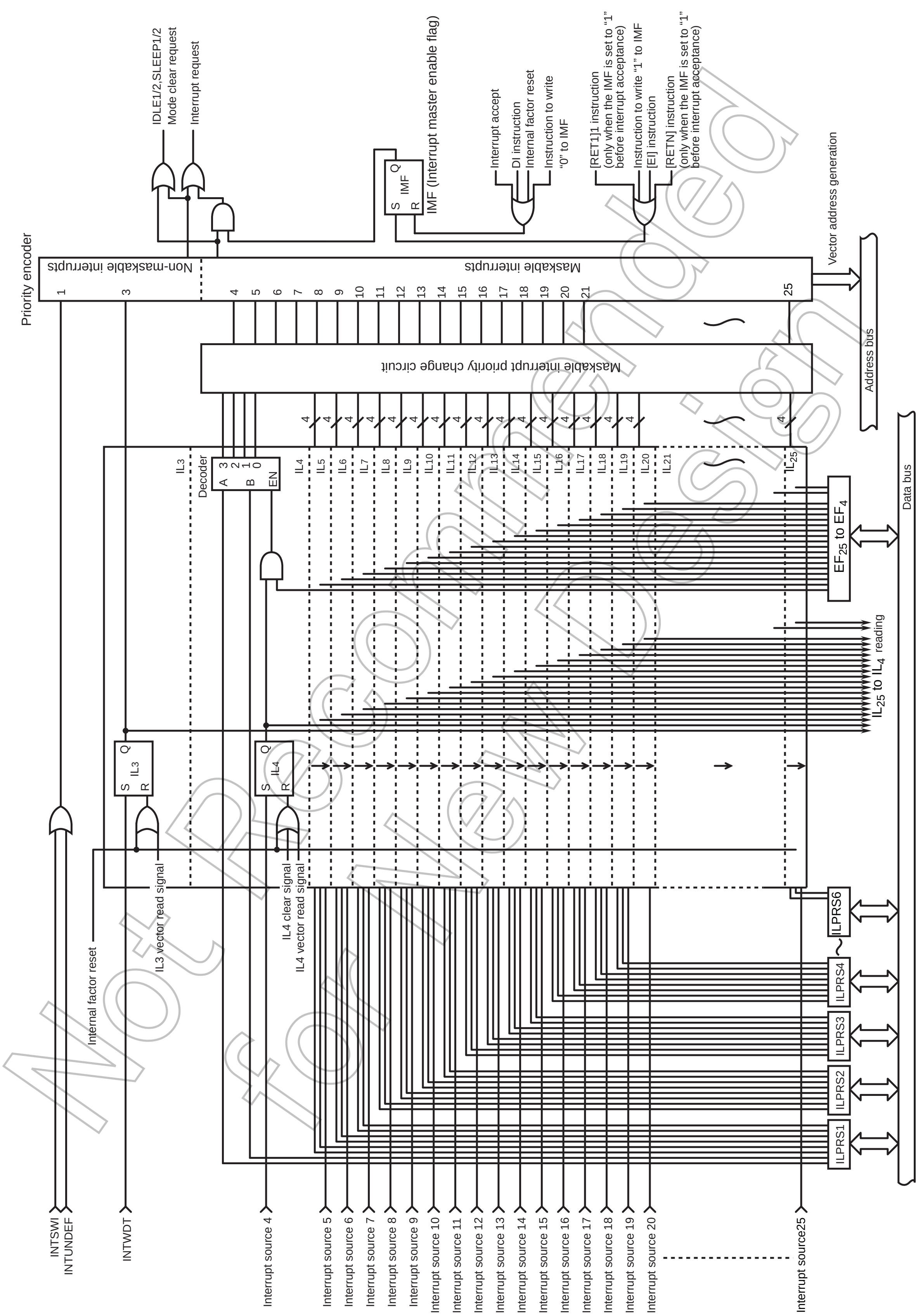


Figure 3-1 Interrupt Control Circuit

3.2 Interrupt Latches (IL25 to IL3)

An interrupt latch is provided for each interrupt source, except for a software interrupt and an undefined instruction execution interrupt. When an interrupt request is generated, the latch is set to "1", and the CPU is requested to accept the interrupt if its acceptance is enabled. The interrupt latch is cleared to "0" immediately after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are located at addresses 0x0FE0, 0x0FE1, 0x0FE2, 0x0FE3 in SFR area. Each latch can be cleared to "0" individually by an instruction. However, IL2 and IL3 interrupt latches cannot be cleared by instructions.

Do not use any read-modify-write instruction, such as a bit manipulation or operation instruction, because it may clear interrupt requests generated while the instruction is executed.

Interrupt latches cannot be set to "1" by using an instruction. Writing "1" to an interrupt latch is equivalent to denying clearing of the interrupt latch, and not setting the interrupt latch.

Since interrupt latches can be read by instructions, the status of interrupt requests can be monitored by software.

Note: In the main program, before manipulating an interrupt latch (IL), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

Example 1:Clears interrupt latches

```
DI                               ;IMF ← 0
LD      (ILL), 0y00111111        ;IL7 to IL6 ← 0
LD      (ILH), 0y11101000        ;IL12, IL10 to IL8 ← 0
EI                               ;IMF ← 1
```

Example 2:Reads interrupt latches

```
LD      WA, (ILL)                ;W ← ILH, A ← ILL
```

Example 3:Tests interrupt latches

```
TEST    (ILL). 7                 ;if IL7=1 then jump
JR      F, SSET                  ;
```

3.3 Interrupt Enable Register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (software interrupt, undefined instruction interrupt and watchdog interrupt). Non-maskable interrupts are accepted regardless of the contents of the EIR.

The EIR consists of the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located at addresses 0x003A, 0x003B, 0x003C, 0x003D in the SFR area, and they can be read and written by instructions (including read-modify-write instructions such as bit manipulation or operation instructions).

3.3.1 Interrupt master enable flag (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all maskable interrupts. Clearing the IMF to "0" disables the acceptance of all maskable interrupts. Setting the IMF to "1" enables the acceptance of the interrupts that are specified by the individual interrupt enable flags.

When an interrupt is accepted, the IMF is stacked and then cleared to "0", which temporarily disables the subsequent maskable interrupts. After the interrupt service routine is executed, the stacked data, which was the status before interrupt acceptance, reloaded on the IMF by return interrupt instruction [RETI]/[RETN].

The IMF is located on bit 0 in EIRL (Address: 0x03A in SFR), and can be read and written by instructions. The IMF is normally set and cleared by [EI] and [DI] instructions respectively. During reset, the IMF is initialized to "0".

3.3.2 Individual interrupt enable flags (EF25 to EF4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of its interrupt, and setting the bit to "0" disables acceptance.

During reset, all the individual interrupt enable flags are initialized to "0" and no maskable interrupts are accepted until the flags are set to "1".

Note: In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Example:Enables interrupts individually and sets IMF

```
DI                                     ;IMF ← 0
LDW     (EIRL), 0y1110100010100000     ;EF15 to EF13, EF11, EF7, EF5 ← 1
:                                      ;Note: IMF should not be set.
:
EI                                     ;IMF ← 1
```

Interrupt latch (ILL)

ILL		7	6	5	4	3	2	1	0
(0x0FE0)	Bit Symbol	IL7	IL6	IL5	IL4	IL3	-	-	-
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0 / INTSIO0	INTTBT	INTWUC	INTWDT			

Interrupt latch (ILH)

ILH		7	6	5	4	3	2	1	0
(0x0FE1)	Bit Symbol	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSBI0/IN-TSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt latch (ILE)

ILE		7	6	5	4	3	2	1	0
(0x0FE2)	Bit Symbol	IL23	IL22	IL21	IL20	IL19	IL18	IL17	IL16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1	INTTCA1	INT4	INT3	INT2	INT1	INT0

Interrupt latch (ILD)

ILD		7	6	5	4	3	2	1	0
(0x0FE3)	Bit Symbol	-	-	-	-	-	-	IL25	IL24
	Read/Write	R	R	R	R	R	R	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function							INTTC03	INTTC02

			Read	Write
IL25 to IL4	Interrupt latch	0:	No interrupt request	Clears the interrupt request (Notes 2 and 3)
		1:	Interrupt request	Does not clear the interrupt request (Interrupt is not set by writing "1".)
IL3		0: 1:	No interrupt request Interrupt request	-

Note 1: IL3 is a read-only register. Writing the register does not affect interrupt latch.

Note 2: In the main program, before manipulating an interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

Note 3: Do not clear IL with read-modify-write instructions such as bit operations.

Note 4: When a read instruction is executed on ILL, bits 0 to 2 are read as "0". Other unused bits are read as "0".

Interrupt enable register (EIRL)

EIRL		7	6	5	4	3	2	1	0
(0x003A)	Bit Symbol	EF7	EF6	EF5	EF4	-	-	-	IMF
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0 / INTSIO0	INTTBT	INTWUC				Interrupt master enable flag

Interrupt enable register (EIRH)

EIRH		7	6	5	4	3	2	1	0
(0x003B)	Bit Symbol	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSBI0/INTSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt enable register (EIRE)

EIRE		7	6	5	4	3	2	1	0
(0x003C)	Bit Symbol	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1	INTTCA1	INT4	INT3	INT2	INT1	INT0

Interrupt enable register (EIRD)

EIRD		7	6	5	4	3	2	1	0
(0x003D)	Bit Symbol	-	-	-	-	-	-	EF25	EF24
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function							INTTC03	INTTC02

EF25 to EF4	Individual interrupt enable flag (Specified for each bit)	0: 1:	Disables the acceptance of each maskable interrupt. Enables the acceptance of each maskable interrupt.
IMF	Interrupt master enable flag	0: 1:	Disables the acceptance of all maskable interrupts. Enables the acceptance of all maskable interrupts.

Note 1: Do not set the IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 2: In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction)

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Note 3: When a read instruction is executed on EIRL, bits 3 to 1 are read as "0". Other unused bits are read as "0".

3.4 Maskable Interrupt Priority Change Function

The priority of maskable interrupts (IL4 to IL25) can be changed to four levels, Levels 0 to 3, regardless of the basic priorities 5 to 26. Interrupt priorities can be changed by the interrupt priority change control register (ILPRS1 to ILPRS6). To raise the interrupt priority, set the Level to a larger number. To lower the interrupt priority, set the Level to a smaller number. When different maskable interrupts are generated simultaneously at the same level, the interrupt with higher basic priority is processed preferentially. For example, when the ILPRS1 register is set to 0xC0 and interrupts IL4 and IL7 are generated at the same time, IL7 is preferentially processed (provided that EF4 and EF7 have been enabled).

After reset is released, all maskable interrupts are set to priority level 0 (the lowest priority).

Note: In the main program, before manipulating the interrupt priority change control register (ILPRS1 to 6), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction).
Set the IMF to "1" as required after operating ILPRS1 to 6 (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate ILPRS1 to 6 before setting the IMF to "1".

Interrupt priority change control register 1

ILPRS1 (0x0FF0)

	7	6	5	4	3	2	1	0
Bit Symbol	IL07P		IL06P		IL05P		IL04P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL07P	Sets the interrupt priority of IL7.	00:	Level 0 (lower priority)
IL06P	Sets the interrupt priority of IL6.	01:	Level 1
IL05P	Sets the interrupt priority of IL5.	10:	Level 2
IL04P	Sets the interrupt priority of IL4.	11:	Level 3 (higher priority)

Interrupt priority change control register 2

ILPRS2 (0x0FF1)

	7	6	5	4	3	2	1	0
Bit Symbol	IL11P		IL10P		IL09P		IL08P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL11P	Sets the interrupt priority of IL11.	00:	Level 0 (lower priority)
IL10P	Sets the interrupt priority of IL10.	01:	Level 1
IL09P	Sets the interrupt priority of IL9.	10:	Level 2
IL08P	Sets the interrupt priority of IL8.	11:	Level 3 (higher priority)

Interrupt priority change control register 3

ILPRS3 (0x0FF2)

	7	6	5	4	3	2	1	0
Bit Symbol	IL15P		IL14P		IL13P		IL12P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL15P	Sets the interrupt priority of IL15.	00:	Level 0 (lower priority)
IL14P	Sets the interrupt priority of IL14.	01:	Level 1
IL13P	Sets the interrupt priority of IL13.	10:	Level 2
IL12P	Sets the interrupt priority of IL12.	11:	Level 3 (higher priority)

Interrupt priority change control register 4

ILPRS4 (0x0FF3)

	7	6	5	4	3	2	1	0
Bit Symbol	IL19P		IL18P		IL17P		IL16P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL19P	Sets the interrupt priority of IL19.	00:	Level 0 (lower priority)
IL18P	Sets the interrupt priority of IL18.	01:	Level 1
IL17P	Sets the interrupt priority of IL17.	10:	Level 2
IL16P	Sets the interrupt priority of IL16.	11:	Level 3 (higher priority)

Interrupt priority change control register 5

ILPRS5 (0x0FF4)

	7	6	5	4	3	2	1	0
Bit Symbol	IL23P		IL22P		IL21P		IL20P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL23P	Sets the interrupt priority of IL23.	00:	Level 0 (lower priority)
IL22P	Sets the interrupt priority of IL22.	01:	Level 1
IL21P	Sets the interrupt priority of IL21.	10:	Level 2
IL20P	Sets the interrupt priority of IL20.	11:	Level 3 (higher priority)

Interrupt priority change control register 6

ILPRS6 (0x0FF5)

	7	6	5	4	3	2	1	0
Bit Symbol	-		-		IL25P		IL24P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

-	-	00:	Level 0 (lower priority)
-	-	01:	Level 1
IL25P	Sets the interrupt priority of IL25.	10:	Level 2
IL24P	Sets the interrupt priority of IL24.	11:	Level 3 (higher priority)

3.5 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8-machine cycles after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts).

3.5.1 Initial Setting

Using an interrupt requires specifying an SP (stack pointer) for it in advance. The SP is a 16-bit register pointing at the start address of a stack. The SP is post-decremented when a subroutine call or a push instruction is executed or when an interrupt request is accepted. It is pre-incremented when a return or pop instruction is executed. Therefore, the stack becomes deeper toward lower stack location addresses. Be sure to reserve a stack area having an appropriate size based on the SP setting.

The SP is initialized to 00FFH after a reset. If you need to change the SP, do so right after a reset or when the interrupt master enable flag (IMF) is “0”.

Example :SP setting

```
LD      SP, 023FH     ; SP = 023FH
LD      SP, SP+04H    ; SP = SP + 04H
ADD     SP, 0010H     ; SP = SP + 0010H
```

3.5.2 Interrupt acceptance processing

Interrupt acceptance processing is packaged as follows.

1. The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.
2. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.
3. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.
4. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
5. The instruction stored at the entry address of the interrupt service program is executed.

Note: When the contents of PSW are saved on the stack, the contents of register bank and IMF are also saved.

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program

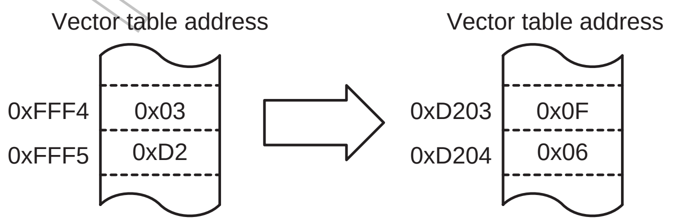


Figure 3-2 Vector table address and Entry address

A maskable interrupt is not accepted until the IMF is set to “1” even if the maskable interrupt is requested in the interrupt service routine.

In order to utilize nested interrupt service, the IMF must be set to “1” in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to “1”. As for non-maskable interrupt, keep interrupt service shorter compared with length between interrupt requests.

3.5.3 Saving/restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the general purpose registers are not. These registers must be saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same data memory area for saving registers. The following methods are used to save/restore the general-purpose registers.

3.5.3.1 Using PUSH and POP instructions

To save only a specific register, PUSH and POP instructions are available.

Example :Using PUSH and POP instructions

```
PINTxx   PUSH                    WA     ; Save WA register
         Interrupt processing
         POP                     WA     ; Restore WA register
         RETI                           ; RETURN
```

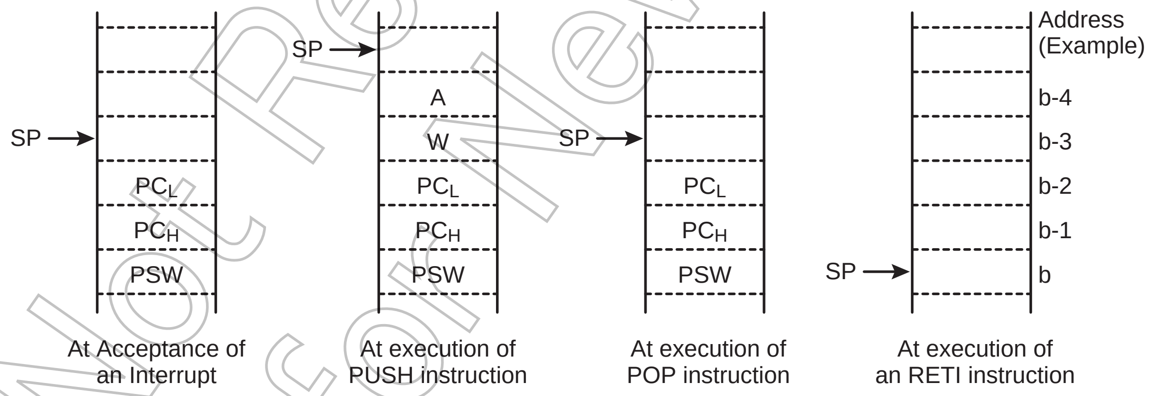


Figure 3-3 Saving/restoring general-purpose registers

3.5.3.2 Using data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

Example :Save/store register using data transfer instructions

```
PINTxx:   LD      (GSAVA), A      ; Save A register
          Interrupt processing
          LD      A, (GSAVA)      ; Restore A register
          RETI                    ; RETURN
```

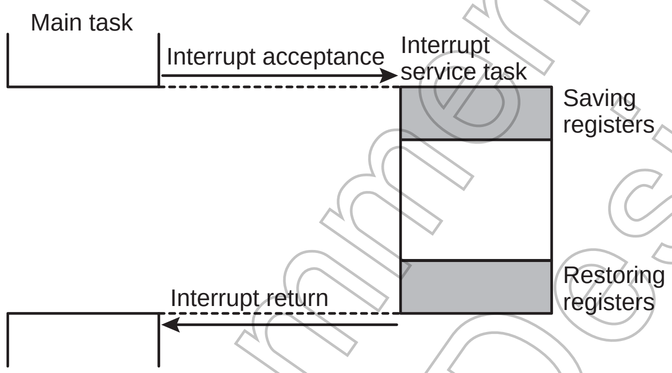


Figure 3-4 Saving/Restoring General-purpose Registers under Interrupt Processing

3.5.3.3 Using a register bank to save/restore general-purpose registers

In non-multiple interrupt handling, the register bank function can be used to save/restore the general-purpose registers at a time. The register bank function saves (switches) the general-purpose registers by executing a register bank manipulation instruction (such as LD RBS,1) at the beginning of an interrupt service task. It is unnecessary to re-execute the register bank manipulation instruction at the end of the interrupt service task because executing the RETI instruction makes a return automatically to the register bank that was being used by the main task according to the content of the PSW.

Note: Two register banks (BANK0 and BANK1) are available. Each bank consists of 8-bit general-purpose registers (W, A, B, C, D, E, H, and L) and 16-bit general-purpose registers (IX and IY).

Example :Saving/restoring registers, using an instruction for transfer with data memory (with the main task using the register bank BANK0)

```
PINTxx:   LD      RBS, 1          ; Switches to the register bank BANK1
          Interrupt processing
          RETI                    ; RETURN
                                  (Makes a return automatically to BANK0 that
                                  was being used by the main task when the
                                  PSW is restored)
```

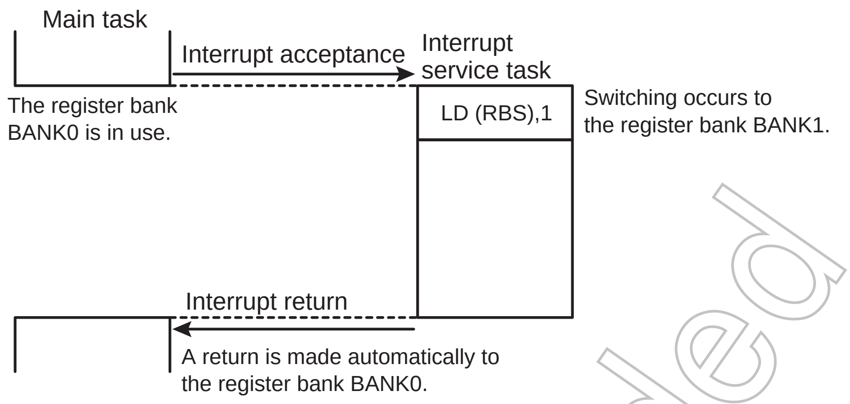


Figure 3-5 Saving/Restoring General-purpose Registers under Interrupt Processing

3.5.4 Interrupt return

Interrupt return instructions [RETI]/[RETN] perform as follows.

[RETI]/[RETN] Interrupt Return
1. Program counter (PC) and program status word (register bank) are restored from tha stack. 2. Stack pointer (SP) is incremented by 3.

3.6 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is the top-priority interrupt).

Use the SWI instruction only for address error detection or for debugging described below.

3.6.1 Address error detection

0xFF is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code 0xFF is an SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing 0xFF to unused areas in the program memory.

3.6.2 Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

3.7 Undefined Instruction Interrupt (INTUNDEF)

When the CPU tries to fetch and execute an instruction that is not defined, INTUNDEF is generated and starts the interrupt processing. INTUNDEF is accepted even if another non-maskable interrupt is in process. The current process is discontinued and the INTUNDEF interrupt process starts soon after it is requested.

Note: The undefined instruction interrupt (INTUNDEF) forces the CPU to jump into the interrupt vector address, as software interrupt (SWI) does.

3.8 Revision History

Rev	Description
RA003	Revised from WDTCR1<WDTOUT> to WDCTR<WDTOUT>
	Added chapter "3.5 Interrupt Sequence"
	"Figure 3-3 Saving/restoring general-purpose registers" Revised SP position

4. External Interrupt control circuit

External interrupts detects the change of the input signal and generates an interrupt request. Noise can be removed by the built-in digital noise canceller.

4.1 Configuration

The external interrupt control circuit consists of a noise canceller, an edge detection circuit, a level detection circuit and an interrupt signal generation circuit.

Externally input signals are input to the rising edge or falling edge or level detection circuit for each external interrupt, after noise is removed by the noise canceller.

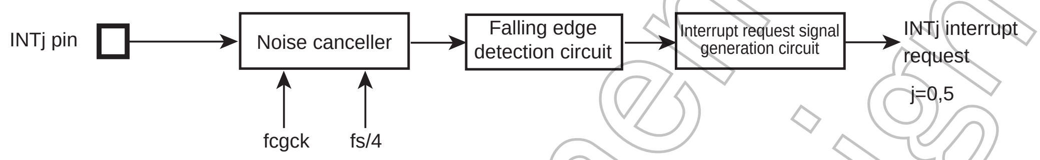


Figure 4-1 External Interrupts 0/5

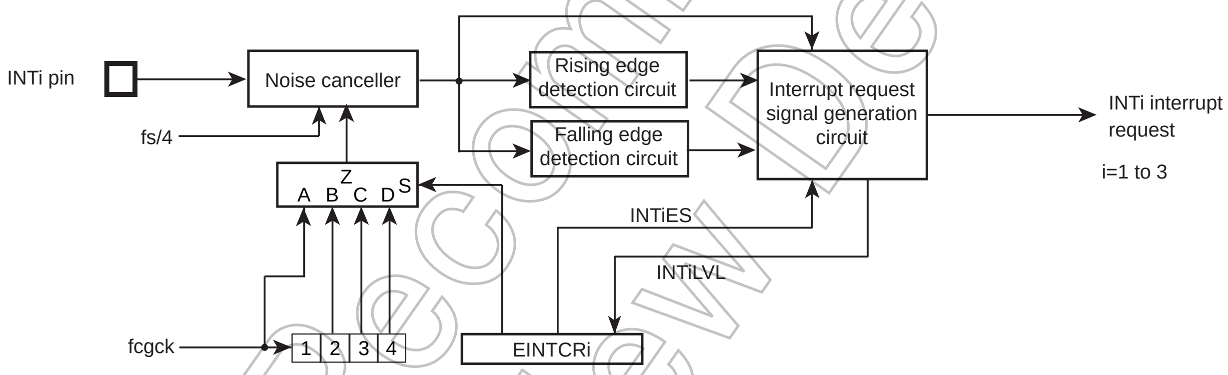


Figure 4-2 External Interrupts 1/2/3

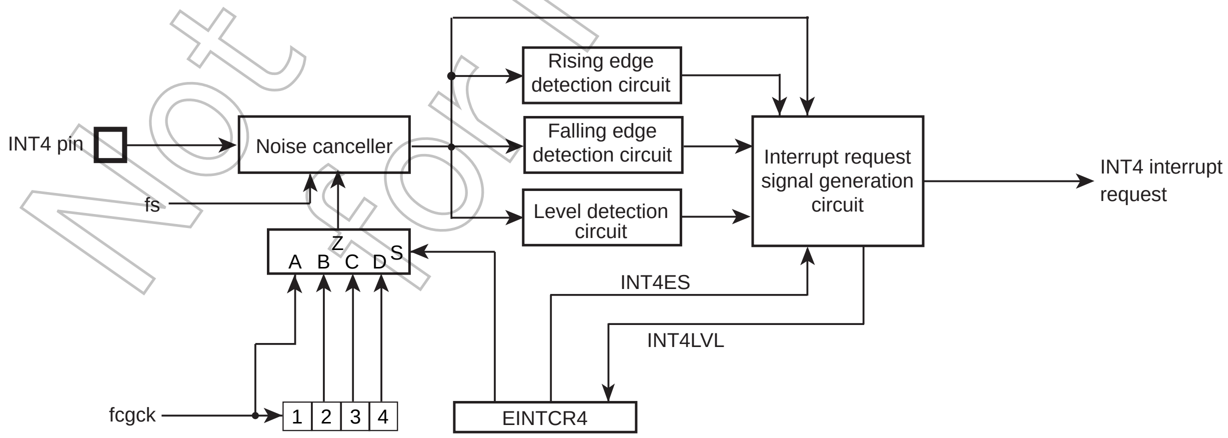


Figure 4-3 External Interrupt 4

4.2 Control

External interrupts are controlled by the following registers:

Low power consumption register 3

POFFCR3 (0x0F77)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 control	0 1	Disable Enable
INT4EN	INT4 control	0 1	Disable Enable
INT3EN	INT3 control	0 1	Disable Enable
INT2EN	INT2 control	0 1	Disable Enable
INT1EN	INT1 control	0 1	Disable Enable
INT0EN	INT0 control	0 1	Disable Enable

Note 1: Clearing INTxEN(x=0 to 5) to "0" stops the clock supply to the external interrupts. This invalidates the data written in the control register for each external interrupt. When using the external interrupts, set INTxEN to "1" and then write data into the control register for each external interrupt.

Note 2: Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Bits 7 and 6 of POFFSET3 are read as "0".

External interrupt control register 1

EINTCR1 (0x0FD8)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	INT1LVL	INT1ES		INT1NC	
	Read/Write	R	R	R	R	R/W		R/W	
	After reset	0	0	0	0	0		0	

INI1LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 1	0 : 1 :	Initial state or signal level "L" Signal level "H"		
INT1ES	Selects the interrupt request generating condition for external interrupt 1	00 : 01 : 10 : 11 :	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal Reserved		
INT1NC	Sets the noise canceller sampling interval for external interrupt 1		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 : 01 : 10 : 11 :	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00 : 01 : 10 : 11 :	fs/4 [Hz] fs/4 [Hz] fs/4 [Hz] fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR1 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR1 are read as "0".

External interrupt control register 2

EINTCR1 (0x0FD9)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT2LVL	INT2ES		INT2NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI2LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 2	0 :	Initial state or signal level "L"
		1 :	Signal level "H"
INT2ES	Selects the interrupt request generating condition for external interrupt 2	00 :	An interrupt request is generated at the rising edge of the noise canceller pass signal
		01 :	An interrupt request is generated at the falling edge of the noise canceller pass signal
		10 :	An interrupt request is generated at both edges of the noise canceller pass signal
		11 :	Reserved

			NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
INT2NC	Sets the noise canceller sampling interval for external interrupt 2	00 :	fcgck [Hz]	00 :	fs/4 [Hz]
		01 :	fcgck / 2^2 [Hz]	01 :	fs/4 [Hz]
		10 :	fcgck / 2^3 [Hz]	10 :	fs/4 [Hz]
		11 :	fcgck / 2^4 [Hz]	11 :	fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR2 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR2 are read as "0".

External interrupt control register 3

EINTCR3 (0x0FDA)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT3LVL	INT3ES		INT3NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI3LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 3	0 : 1 :	Initial state or signal level "L" Signal level "H"		
INT3ES	Selects the interrupt request generating condition for external interrupt 3	00 : 01 : 10 : 11 :	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal Reserved		
INT3NC	Sets the noise canceller sampling interval for external interrupt 3		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 : 01 : 10 : 11 :	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00 : 01 : 10 : 11 :	fs/4 [Hz] fs/4 [Hz] fs/4 [Hz] fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR3 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR3 are read as "0".

External interrupt control register 4

EINTCR4 (0x0FDB)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT4LVL	INT4ES		INT4NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI4LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 4	0 : 1 :	Initial state or signal level "L" Signal level "H"		
INT4ES	Selects the interrupt request generating condition for external interrupt 4	00 : 01 : 10 : 11 :	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal An interrupt request is generated at "H" of the noise canceller pass signal		
INT4NC	Sets the noise canceller sampling interval for external interrupt 4		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 : 01 : 10 : 11 :	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00 : 01 : 10 : 11 :	fs/4 [Hz] fs/4 [Hz] fs/4 [Hz] fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR4 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: The contents of EINTCRx<INTxLVL> are updated each time an interrupt request signal is generated.

Note 5: Bits 7 to 5 of EINTCR4 are read as "0".

4.3 Function

The condition for generating interrupt request signals and the noise cancel time can be set for external interrupts 1 to 4.

The condition for generating interrupt request signals and the noise cancel time are fixed for external interrupts 0 and 5.

Table 4-1 External Interrupts

Source	Pin	Enable conditions	Interrupt request signal generated at	External interrupt pin input signal width and noise removal	
				NORMAL1/2, IDLE1/2	SLOW1/2, SLEEP1
INT0	$\overline{\text{INT0}}$	IMF AND EF16 = 1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT1	INT1	IMF AND EF17 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT2	INT2	IMF AND EF18 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT3	INT3	IMF AND EF19 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT4	INT4	IMF AND EF20 = 1	Falling edge Rising edge Both edges "H" level	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT5	$\overline{\text{INT5}}$	IMF AND EF8 = 1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal

Note 1: fcgck, Gear clock [Hz]; fs, low frequency clock [Hz]; fspl, Sampling interval [Hz]

4.3.1 Low power consumption function

External interrupts have a function that saves power by using the low power consumption register (POFFCR3) when they are not used.

Setting POFFCR3<INTxEN> to "0" stops (disables) the basic clock for external interrupts and helps save power. Note that this makes external interrupts unavailable. Setting POFFCR3<INTxEN> to "1" supplies (enables) the basic clock for external interrupts and makes external interrupts available.

After reset, POFFCR3<INTxEN> is initialized to "0" and external interrupts become unavailable. When using the external interrupt function for the first time, be sure to set POFFCR3<INTxEN> to "1" in the initial setting of software (before operating the external interrupt control registers).

Note: Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

4.3.2 External interrupt 0

External interrupt 0 detects the falling edge of the $\overline{\text{INT0}}$ pin and generates interrupt request signals.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

4.3.3 External interrupts 1/2/3

External interrupts 1/2/3 detect the falling edge, the rising edge or both edges of the INT1, INT2 and INT3 pins and generate interrupt request signals.

4.3.3.1 Interrupt request signal generating condition detection function

Select interrupt request signal generating conditions at EINTCRx<INTxES> for external interrupts 1/2/3.

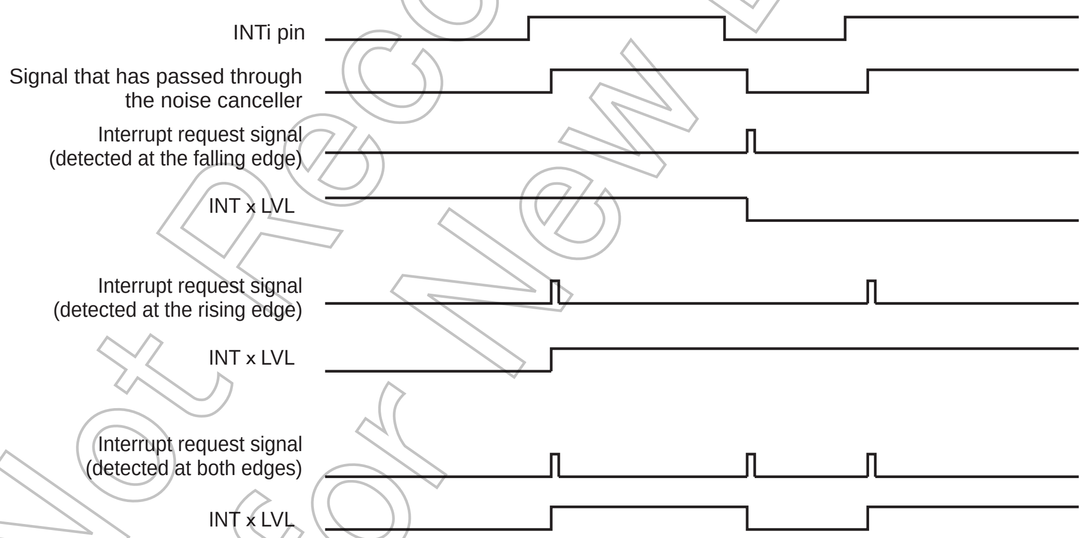
Table 4-2 Selection of Interrupt Request Generation Edge

EINTCRx<INTxES>	Detected at
00	Rising edge
01	Falling edge
10	Both edges
11	Reserved

Note: x=1 to 3

4.3.3.2 A noise canceller pass signal monitoring function when interrupt request signals are generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCRx<INTxLVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCRx<INTxLVL>.



Note: The contents of EINTCRx<INTxLVL> are updated each time an interrupt request signal is generated.

Figure 4-4 Interrupt Request Generation and EINTCRx<INTxLVL>

4.3.3.3 Noise cancel time selection function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INTxNC>. If the same level is detected three consecutive times, the signal is recognized as a signal. If not, the signal is removed as noise.

Table 4-3 Noise Canceller Sampling Lock

EINTCRx<INTxNC>	Sampling interval
00	fcgck
01	$fcgck/2^2$
10	$fcgck/2^3$
11	$fcgck/2^4$

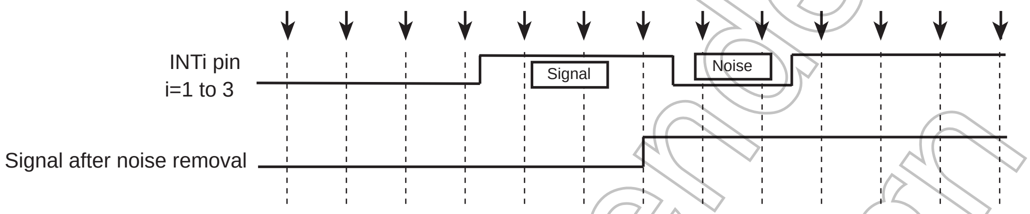


Figure 4-5 Noise Cancel Operation

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low frequency clock divided by 4. If the same level is detected twice consecutively, the signal is recognized as a signal.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: If noise is input consecutively during sampling of external interrupt pins, the noise cancel function does not work properly. Set EINTCRx<INTxNC> according to the cycle of externally input noise.

Note 2: If an external interrupt pin is used as an output port, the input signal to the port is fixed to "L" when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.

Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

4.3.4 External interrupt 4

External interrupt 4 detects the falling edge, the rising edge, both edges or "H" level of the INT4 pin and generates interrupt request signals.

4.3.4.1 Interrupt request signal generating condition detection function

Select an interrupt request signal generating condition at EINTCR4<INT4ES> for external interrupt 4.

Table 4-4 Selection of Interrupt Request Generation Edge

EINTCR4<INT4ES>	Detected at
00	Rising edge
01	Falling edge
10	Both edges
11	"H" level interrupt

4.3.4.2 A noise canceller pass signal monitoring function when interrupt request signals are generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCR4<INT4LVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCR4<INT4LVL>.

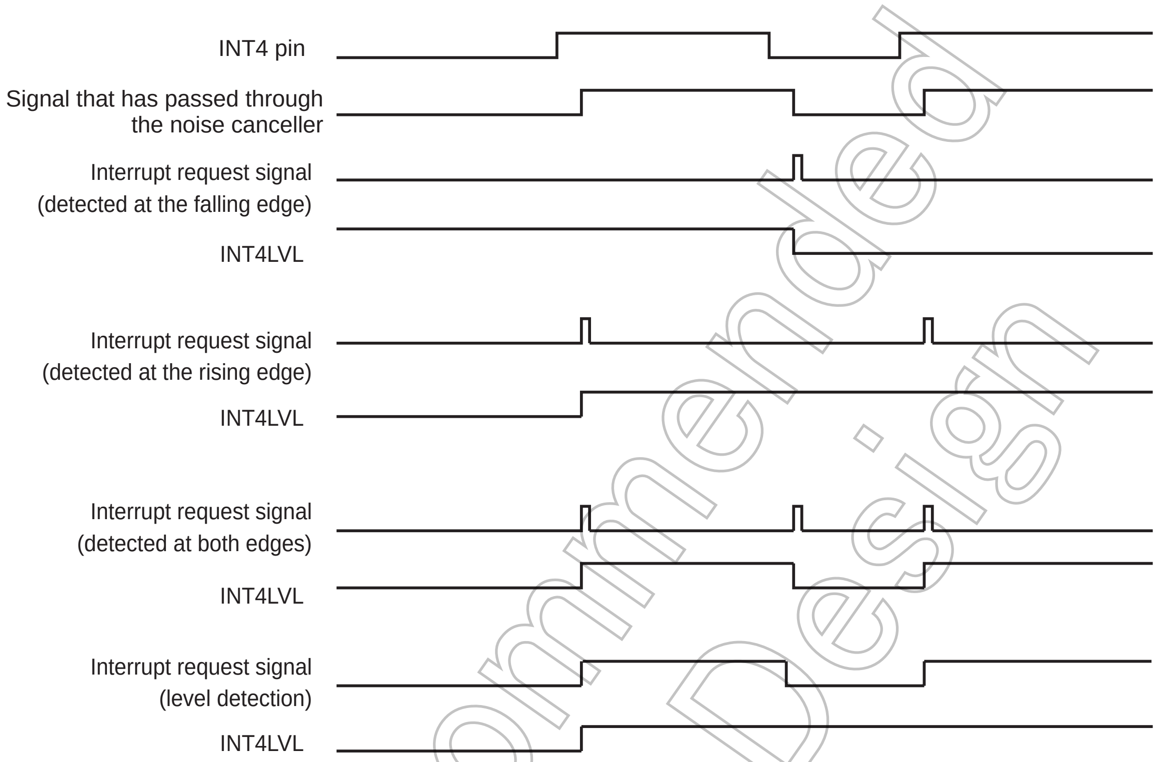


Figure 4-6 Interrupt Request Generation and EINTCR4<INT4LVL>

4.3.4.3 Noise cancel time selection function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INT4NC>. If the same level is detected three consecutive times, the signal is recognized as a signal. If not, the signal is removed as noise.

Table 4-5 Noise Canceller Sampling Lock

EINTCR4<INT4NC>	Sampling interval
00	fcgck
01	fcgck/2^2
10	fcgck/2^3
11	fcgck/2^4

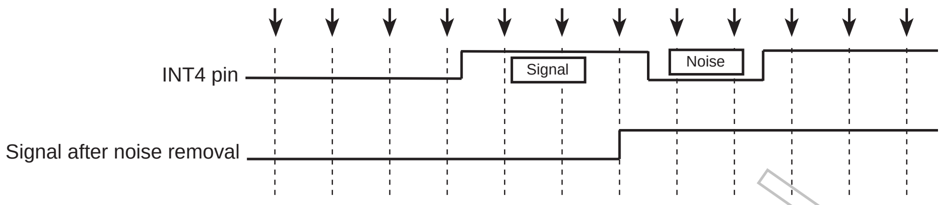


Figure 4-7 Noise Cancel Operation

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low frequency clock divided by 4. If the same level is detected twice consecutively, the signal is recognized as a signal.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: When noise is input consecutively during sampling external interrupt pins, the noise cancel function does not work properly. Set EINTCRx<INTxNC> according to the cycle of externally input noise.

Note 2: When an external interrupt pin is used as an output port, the input signal to the port is fixed to "L" when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.

Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

4.3.5 External interrupt 5

External interrupt 5 detects the falling edge of the $\overline{\text{INT5}}$ pin and generates interrupt request signals.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

5. Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and return the CPU to a system recovery routine.

The watchdog timer signals used for detecting malfunctions can be programmed as watchdog interrupt request signals or watchdog timer reset signals.

Note: Care must be taken in system designing since the watchdog timer may not fulfill its functions due to disturbing noise and other effects.

5.1 Configuration

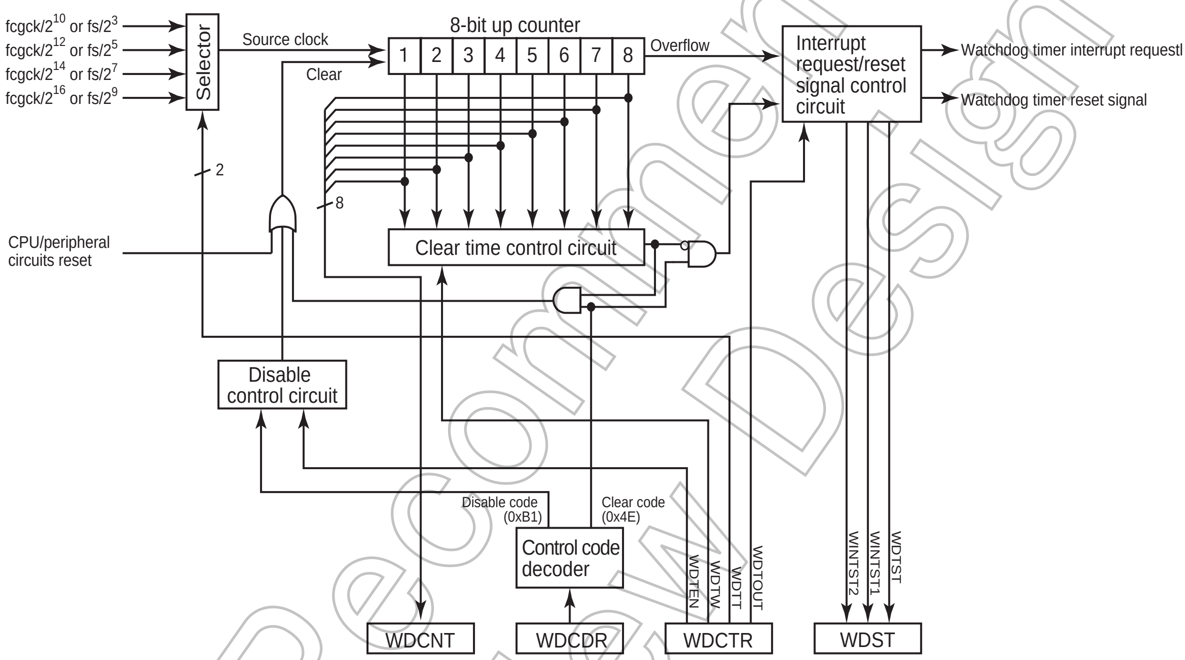


Figure 5-1 Watchdog Timer Configuration

5.2 Control

The watchdog timer is controlled by the watchdog timer control register (WDCTR), the watchdog timer control code register (WDCDR), the watchdog timer counter monitor (WDCNT) and the watchdog timer status (WDST).

The watchdog timer is enabled automatically just after the warm-up operation that follows reset is finished.

Watchdog timer control register

WDCTR (0x0FD4)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WDTEN	WDTW		WDTT		WDTOUT
Read/Write	R	R	R/W	R/W		R/W		R/W
After reset	1	0	1	0	0	1	1	0

WDTEN	Enables/disables the watchdog timer operation.	0 : 1 :	Disable Enable		
WDTW	Sets the clear time of the 8-bit up counter.	00 :	The 8-bit up counter is cleared by writing the clear code at any point within the overflow time of the 8-bit up counter.		
		01 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first quarter of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first quarter of the overflow time has elapsed.		
		10 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first half of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first half of the overflow time has elapsed.		
		11 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first three quarters of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first three quarters of the overflow time have elapsed.		
WDTT	Sets the overflow time of the 8-bit up counter.		NORMAL mode		SLOW mode
			DV9CK=0	DV9CK=1	
		00 :	2^{18}/fcgck	2^{11}/fs	2^{11}/fs
		01:	2^{20}/fcgck	2^{13}/fs	2^{13}/fs
		10:	2^{22}/fcgck	2^{15}/fs	2^{15}/fs
		11:	2^{24}/fcgck	2^{17}/fs	2^{17}/fs
WDTOUT	Selects an overflow detection signal of the 8-bit up counter.	0 : 1 :	Watchdog timer interrupt request signal Watchdog timer reset request signal		

Note 1: fcgck, Gear clock [Hz]; fs, Low frequency clock [Hz]

Note 2: WDCTR<WDTW>, WDCTR<WDTT> and WDCTR<WDTOUT> cannot be changed when WDCTR<WDTEN> is "1". If WDCTR<WDTEN> is "1", clear WDCTR<WDTEN> to "0" and write the disable code (0xB1) into WDCDR to disable the watchdog timer operation. Note that WDCTR<WDTW>, WDCTR<WDTT> and WDCTR<WDTOUT> can be changed at the same time as setting WDCTR<WDTEN> to "1".

Note 3: Bit 7 and bit 6 of WDCTR are read as "1" and "0" respectively.

Watchdog timer control code register

WDCDR (0x0FD5)

	7	6	5	4	3	2	1	0
Bit Symbol	WDTCR2							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

WDTCR2	Writes watchdog timer control codes.	0x4E :	Clears the watchdog timer. (Clear code)
		0xB1 :	Disables the watchdog timer operation and clears the 8-bit up counter when WDCTR<WDTEN> is "0". (Disable code)
		Others :	Invalid

Note: WDCDR is a write-only register and must not be accessed by using a read-modify-write instruction, such as a bit operation.

8-bit up counter monitor

WDCNT (0x0FD6)

	7	6	5	4	3	2	1	0
Bit Symbol	WDCNT							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0

WDCNT	Monitors the count value of the 8-bit up counter	The count value of the 8-bit up counter is read.

Watchdog timer status

WDST (0x0FD7)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	WINTST2	WINTST1	WDTST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	1	0	1	1	0	0	1

WINTST2	Watchdog timer interrupt request signal factor status 2	0 :	No watchdog timer interrupt request signal has occurred.
		1 :	A watchdog timer interrupt request signal has occurred due to the overflow of the 8-bit up counter.
WINTST1	Watchdog timer interrupt request signal factor status 1	0 :	No watchdog timer interrupt request signal has occurred.
		1 :	A watchdog timer interrupt request signal has occurred due to releasing of the 8-bit up counter outside the clear time.
WDTST	Watchdog timer operating state status	0 :	Operation disabled
		1 :	Operation enabled

Note 1: WDST<WINTST2> and WDST<WINTST1> are cleared to "0" by reading WDST.

Note 2: Values after reset are read from bits 7 to 3 of WDST.

5.3 Functions

The watchdog timer can detect the CPU malfunctions and deadlock by detecting the overflow of the 8-bit up counter and detecting releasing of the 8-bit up counter outside the clear time.

The watchdog timer stoppage and other abnormalities can be detected by reading the count value of the 8-bit up counter at random times and comparing the value to the last read value.

5.3.1 Setting of enabling/disabling the watchdog timer operation

Setting WDCTR<WDTEN> to "1" enables the watchdog timer operation, and the 8-bit up counter starts counting the source clock.

WDCTR<WDTEN> is initialized to "1" after the warm-up operation that follows reset is released. This means that the watchdog timer is enabled.

To disable the watchdog timer operation, clear WDCTR<WDTEN> to "0" and write 0xB1 into WDCDR. Disabling the watchdog timer operation clears the 8-bit up counter to "0".

Note: If the overflow of the 8-bit up counter occurs at the same time as 0xB1 (disable code) is written into WDCDR with WDCTR<WDTEN> set at "1", the watchdog timer operation is disabled preferentially and the overflow detection is not executed.

To re-enable the watchdog timer operation, set WDCTR<WDTEN> to "1". There is no need to write a control code into WDCDR.

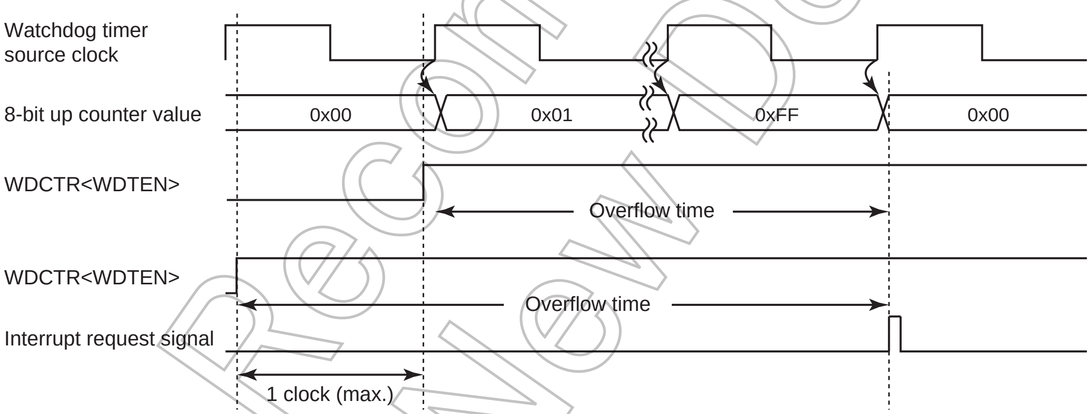


Figure 5-2 WDCTR<WDTEN> Set Timing and Overflow Time

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within the period of the overflow time minus 1 source clock cycle.

5.3.2 Setting the clear time of the 8-bit up counter

WDCTR<WDTW> sets the clear time of the 8-bit up counter.

When WDCTR<WDTW> is "00", the clear time is equal to the overflow time of the 8-bit up counter, and the 8-bit up counter can be cleared at any time.

When WDCTR<WDTW> is not "00", the clear time is fixed to only a certain period within the overflow time of the 8-bit up counter. If the operation for releasing the 8-bit up counter is attempted outside the clear time, a watchdog timer interrupt request signal occurs.

At this time, the watchdog timer is not cleared but continues counting. If the 8-bit up counter is not cleared within the clear time, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs due to the overflow, depending on the WDCTR<WDTOUT> setting.

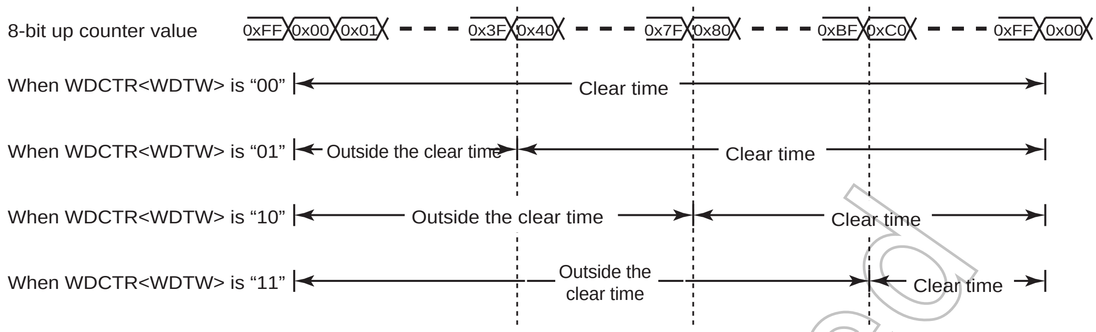


Figure 5-3 WDCTR<WDTW> and the 8-bit up Counter Clear Time

5.3.3 Setting the overflow time of the 8-bit up counter

WDCTR<WDTT> sets the overflow time of the 8-bit up counter.

When the 8-bit up counter overflows, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs, depending on the WDCTR<WDTOUT> setting.

If the watchdog timer interrupt request signal is selected as the malfunction detection signal, the watchdog counter continues counting, even after the overflow has occurred.

The watchdog timer temporarily stops counting up in the STOP mode (including warm-up) or in the IDLE/ SLEEP mode, and restarts counting up after the STOP/IDLE/SLEEP mode is released. To prevent the 8-bit up counter from overflowing immediately after the STOP/IDLE/SLEEP mode is released, it is recommended to clear the 8-bit up counter before the operation mode is changed.

Table 5-1 Watchdog Timer Overflow Time (fcgck=10.0 MHz; fs=32.768 kHz)

WDTT	Watchdog timer overflow time [s]		
	NORMAL mode		SLOW mode
	DV9CK = 0	DV9CK = 1	
00	26.21 m	62.50 m	62.50 m
01	104.86 m	250.00 m	250.00 m
10	419.43 m	1.000	1.000
11	1.678	4.000	4.000

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within a period of the overflow time minus 1 source clock cycle.

5.3.4 Setting an overflow detection signal of the 8-bit up counter

WDCTR<WDTOUT> selects a signal to be generated when the overflow of the 8-bit up counter is detected.

1. When the watchdog timer interrupt request signal is selected (when WDCTR<WDTOUT> is "0")

   Releasing WDCTR<WDTOUT> to "0" causes a watchdog timer interrupt request signal to occur when the 8-bit up counter overflows.

   A watchdog timer interrupt is a non-maskable interrupt, and its request is always accepted, regardless of the interrupt master enable flag (IMF) setting.

Note: When a watchdog timer interrupt is generated while another interrupt, including a watchdog timer interrupt, is already accepted, the new watchdog timer interrupt is processed immediately and the preceding interrupt is put on hold. Therefore, if watchdog timer interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

2. When the watchdog timer reset request signal is selected (when WDCTR<WDTOUT> is "1")

   Setting WDCTR<WDTOUT> to "1" causes a watchdog timer reset request signal to occur when the 8-bit up counter overflows.

   This watchdog timer reset request signal resets the TMP89CM42 and starts the warm-up operation.

5.3.5 Writing the watchdog timer control codes

The watchdog timer control codes are written into WDCDR.

By writing 0x4E (clear code) into WDCDR, the 8-bit up counter is cleared to "0" and continues counting the source clock.

When WDCTR<WDTEN> is "0", writing 0xB1 (disable code) into WDCDR disables the watchdog timer operation.

To prevent the 8-bit up counter from overflowing, clear the 8-bit up counter in a period shorter than the overflow time of the 8-bit up counter and within the clear time.

By designing the program so that no overflow will occur, the program malfunctions and deadlock can be detected through interrupts generated by watchdog timer interrupt request signals.

By applying a reset to the microcomputer using watchdog timer reset request signals, the CPU can be restored from malfunctions and deadlock.

Example: When WDCTR<WDTEN> is "0", set the watchdog timer detection time to 2^{20}/fcgck [s], set the counter clear time to half of the overflow time, and allow a watchdog timer reset request signal to occur if a malfunction is detected.

```
        LD      (WDCTR), 0y00110011        ;WDTW←10, WDTT←01, WDTOUT←1
```

Clear the 8-bit up counter at a point after half of its overflow time and within a period of the overflow time minus 1 source clock cycle.

```
        LD      (WDCDR), 0x4E              ;Clear the 8-bit up counter
```

Clear the 8-bit up counter at a point after half of its overflow time and within a period of the overflow time minus 1 source clock cycle.

```
        LD      (WDCDR), 0x4E              ;Clear the 8-bit up counter
```

Note: If the overflow of the 8-bit up counter and writing of 0x4E (clear code) into WDCDR occur simultaneously, the 8-bit up counter is cleared preferentially and the overflow detection is not executed.

5.3.6 Reading the 8-bit up counter

The counter value of the 8-bit up counter can be read by reading WDCNT.

The stoppage of the 8-bit up counter can be detected by reading WDCNT at random times and comparing the value to the last read value.

5.3.7 Reading the watchdog timer status

The watchdog timer status can be read at WDST.

WDST<WDTST> is set to "1" when the watchdog timer operation is enabled, and it is cleared to "0" when the watchdog timer operation is disabled.

WDST<WINTST2> is set to "1" when a watchdog timer interrupt request signal occurs due to the overflow of the 8-bit up counter.

WDST<WINTST1> is set to "1" when a watchdog timer interrupt request signal occurs due to the operation for releasing the 8-bit up counter outside the clear time.

You can know which factor has caused a watchdog timer interrupt request signal by reading WDST<WINTST2> and WDST<WINTST1> in the watchdog timer interrupt service routine.

WDST<WINTST2> and WDST<WINTST1> are cleared to "0" when WDST is read. If WDST is read at the same time as the condition for turning WDST<WINTST2> or WDST<WINTST1> to "1" is satisfied, WDST<WINTST2> or WDST<WINTST1> is set to "1", rather than being cleared.

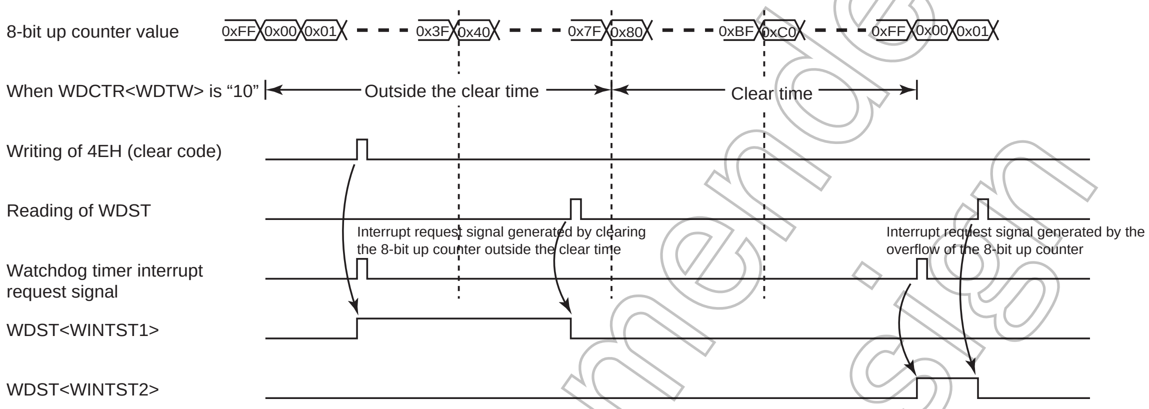


Figure 5-4 Changes in the Watchdog Timer Status

6. Power-on Reset Circuit

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

6.1 Configuration

The power-on reset circuit consists of a reference voltage generation circuit and a comparator.

The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.

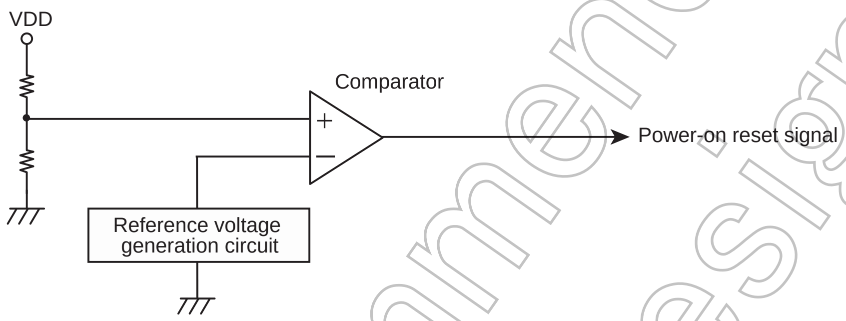


Figure 6-1 Power-on Reset Circuit

6.2 Function

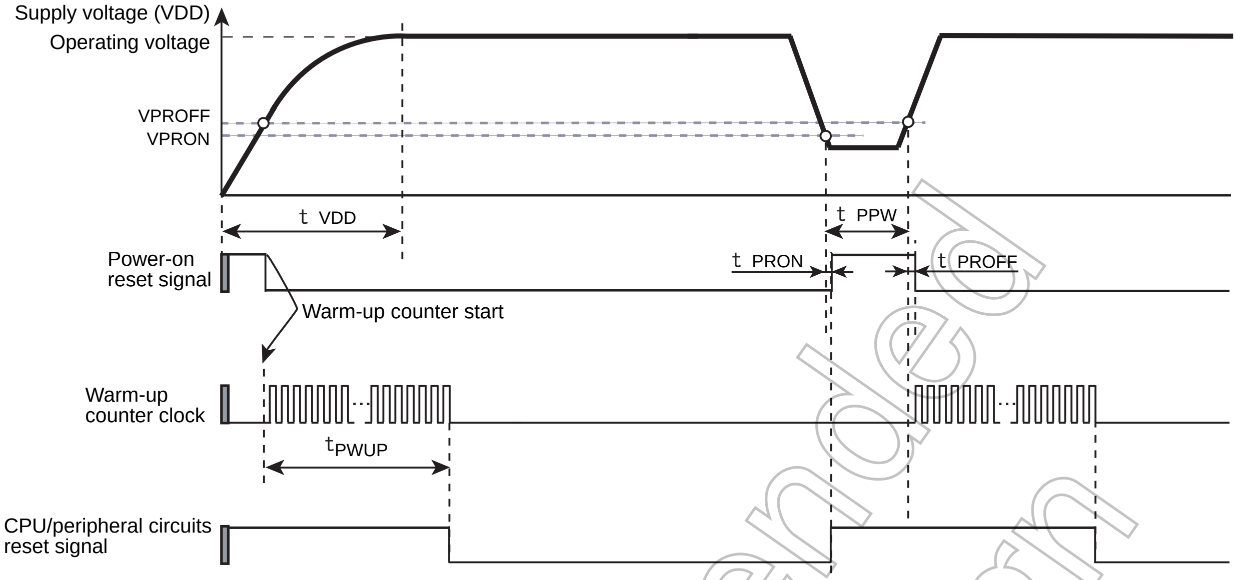
When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated and if it is higher than the releasing voltage of the power-on reset circuit, a power-on reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a power-on reset signal is generated.

Until the power-on reset signal is generated, a warm-up circuit and a CPU is reset.

When the power-on reset signal is released, the warm-up circuit is activated. The reset of the CPU and peripheral circuits is released after the warm-up time that follows reset release has elapsed.

Increase the supply voltage into the operating range during the period from detection of the power-on reset release voltage until the end of the warm-up time that follows reset release. If the supply voltage has not reached the operating range by the end of the warm-up time that follows reset release, the TMP89CM42 cannot operate properly.



Note 1: The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

Note 2: For the AC timing, refer to the electrical characteristics.

Figure 6-2 Operation Timing of Power-on Reset

7. Voltage Detection Circuit

The voltage detection circuit detects any decrease in the supply voltage and generates INTVLTD interrupt request signals and voltage detection reset signals.

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

7.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (VDD) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDxLVL) (x = 1 or 2), and the comparator compares it with the reference voltage. When the comparator detects the selected voltage, a voltage detection reset signal or an INTVLTD interrupt request signal can be generated.

Whether to generate a voltage detection reset signal or an INTVLTD interrupt request signal can be programmed by software. In the former case, a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL). In the latter case, an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage level.

Note: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently if the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

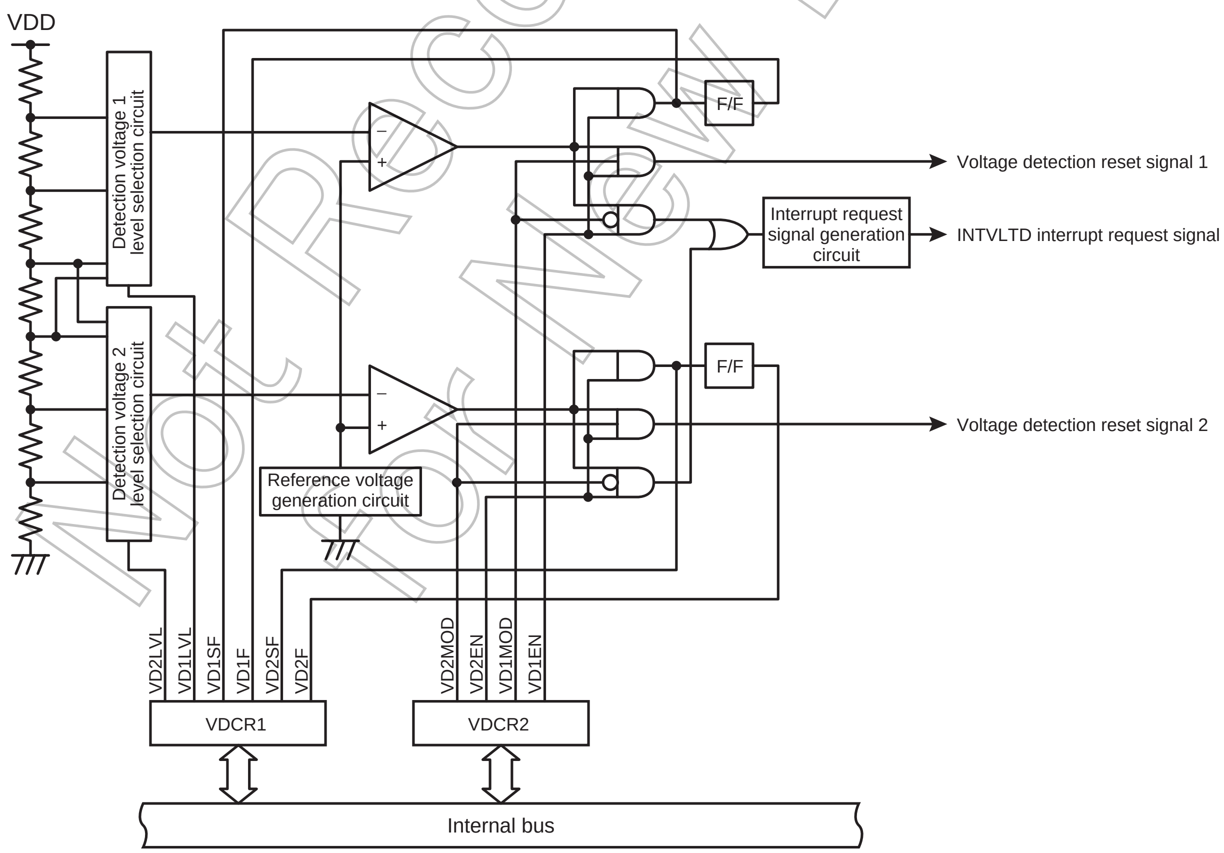


Figure 7-1 Voltage Detection Circuit

7.2 Control

The voltage detection circuit is controlled by voltage detection control registers 1 and 2.

Voltage detection control register 1

VDCR1 (0x0FC6)		7	6	5	4	3	2	1	0
	Bit Symbol	VD2F	VD2SF	VD2LVL		VD1F	VD1SF	VD1LVL	
	Read/Write	R/W	Read Only	R/W		R/W	Read Only	R/W	
	After reset	0	0	1	0	0	0	0	0

			Read	Write
VD2F	Voltage detection 2 flag (Retains the state when VDD<VD2LVL is detected)	0:	VDD ≥ VD2LVL	Clears VD2F to "0"
		1:	VDD < VD2LVL	-
VD2SF	Voltage detection 2 status flag (Magnitude relation of VDD and VD2LVL when they are read)	0:	VDD ≥ VD2LVL	
		1:	VDD < VD2LVL	
VD2LVL	Selection for detection voltage 2	00:	2.35 +0.15 / -0.15V	
		01:	3.15 +0.15 / -0.15V	
		10:	2.85 +0.15 / -0.15V	
		11:	2.65 +0.15 / -0.15V	
			Read	Write
VD1F	Voltage detection 1 flag (Retains the state when VDD < VD1LVL is detected)	0:	VDD ≥ VD1LVL	Clears VD1F to "0"
		1:	VDD < VD1LVL	-
VD1SF	Voltage detection 1 status flag (Magnitude relation of VDD and VD1LVL when they are read)	0:	VDD ≥ VD1LVL	
		1:	VDD < VD1LVL	
VD1LVL	Selection for detection voltage 1	00:	4.50 +0.2 / -0.2V	
		01:	4.20 +0.2 / -0.2V	
		10:	3.70 +0.2 / -0.2V	
		11:	3.15 +0.15 / -0.15V	

Note 1: VDCR1 is initialized by a power-on reset or an external reset input.

Note 2: When VD2F or VD1F is cleared by the software and is set due to voltage detection at the same time, the setting due to voltage detection is given priority.

Note 3: VD2F and VD1F cannot be programmed to "1" by the software.

Voltage detection control register 2

VDCR2 (0x0FC7)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	"0"	"0"	VD2MOD	VD2EN	VD1MOD	VD1EN
	Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

VD2MOD	Selects the operation mode of voltage detection 2	0:	Generate an INTVLTD interrupt request signal
		1:	Generate a voltage detection reset 2 signal
VD2EN	Enables/disables the operation of voltage detection 2	0:	Disables the operation of voltage detection 2
		1:	Enables the operation of voltage detection 2
VD1MOD	Selects the operation mode of voltage detection 1	0:	Generates an INTVLTD interrupt request signal
		1:	Generates a voltage detection reset signal
VD1EN	Enables/disables the operation of voltage detection 1	0:	Disables the operation of voltage detection 1
		1:	Enables the operation of voltage detection 1

Note 1: VDCR2 is initialized by a power-on reset or an external reset input.

Note 2: Bits 7 and 6 of VDCR2 are read as "0".

Note 3: Bit 5 and 4 of VDCR2 should be cleared to "0".

7.3 Function

Two detection voltages (VDxLVL, x = 1 to 2) can be set in the voltage detection circuit. For each voltage, enabling/disabling the voltage detection and the operation to be executed when the supply voltage (VDD) falls to or below the detection voltage (VDxLVL) can be programmed.

7.3.1 Enabling/disabling the voltage detection operation

Setting VDCR2<VDxEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR2<VDxEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (VDD) is lower than the detection voltage (VDxLVL), setting VDCR2<VDxEN> to "1" generates an INTVLTD interrupt request signal or a voltage detection reset signal at the time.

7.3.2 Selecting the voltage detection operation mode

When VDCR2<VDxMOD> is set to "0", the voltage detection operation mode is set to generate INTVLTD interrupt request signals. When VDCR2<VDxMOD> is set to "1", the operation mode is set to generate voltage detection reset signals.

- When the operation mode is set to generate INTVLTD interrupt signals (VDCR2<VDxMOD>="0")

  When VDCR2<VDxEN>="1", an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage (VDxLVL).

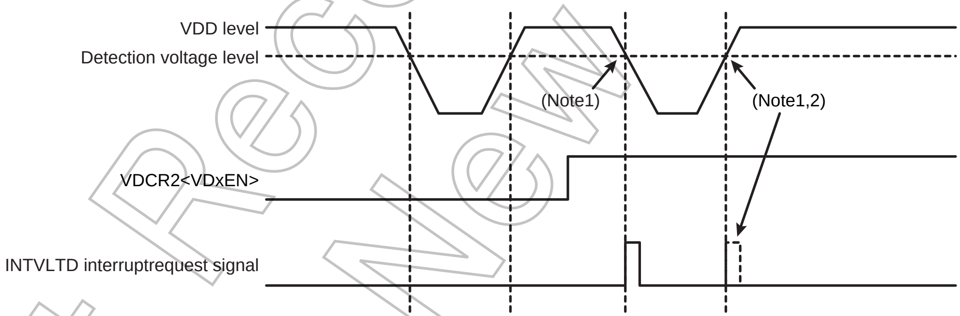


Figure 7-2 Voltage Detection Interrupt Request

Note1: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently when the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

Note2: In debug using the RTE870/C1 In-Circuit Emulator (ICE mode) with the TMP89C900 mounted on it, no interrupt is generated when the supply voltage rises to the detection voltage. Since the #!Undefined!# may operate differently, take account of this difference when debugging programs.

Note3: If the supply voltage (VDD) falls to the detection voltage (VDxLVL) during IDLE0 or SLEEP0 mode, an INTVLTD interrupt request signal is generated after the TBT counts the specified period and IDLE0 or SLEEP mode is released. In the case of STOP mode, an INTLVTD interrupt request signal is generated after STOP mode is released by the $\overline{\text{STOP}}$ pin.

- When the operation mode is set to generate voltage detection reset signals (VDCR2<VDxMOD>="1")

  When VDCR2<VDxEN> = "1", a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL).

VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. A voltage detection reset signal is generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

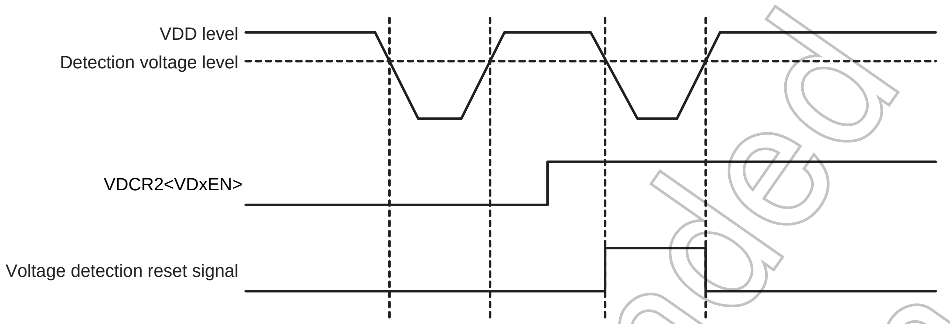


Figure 7-3 Voltage Detection Reset Signal

7.3.3 Selecting the detection voltage level

Select a detection voltage at VDCR1<VDxLVL>.

7.3.4 Voltage detection flag and voltage detection status flag

The magnitude relation between the supply voltage (VDD) and the detection voltage (VDxLVL) can be checked by reading VDCR1<VDxF> and VDCR1<VDxSF>.

If VDCR2<VDxEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL), VDCR1<VDxF> is set to "1" and is held in this state. VDCR1<VDxF> is not cleared to "0" when the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDxLVL).

When VDCR2<VDxEN> is cleared to "0" after VDCR1<VDxF> is set to "1", the previous state is still held. To clear VDCR1<VDxF>, "0" must be written to it.

If VDCR2<VDxEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL), VDCR1<VDxSF> is set to "1". When the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDxLVL), VDCR1<VDxSF> is cleared to "0".

Unlike VDCR1<VDxF>, VDCR1<VDxSF> does not hold the set state.

Note 1: When the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL) in the STOP, IDLE0 or SLEEP0 mode, the voltage detection flag and the voltage detection status flag are changed after the operation mode is returned to NORMAL or SLOW mode.

Note 2: Depending on the voltage detection timing, the voltage detection status flag (VDxSF) may be changed earlier than the voltage detection flag (VDxF) by a maximum of 2/fcgck[s].

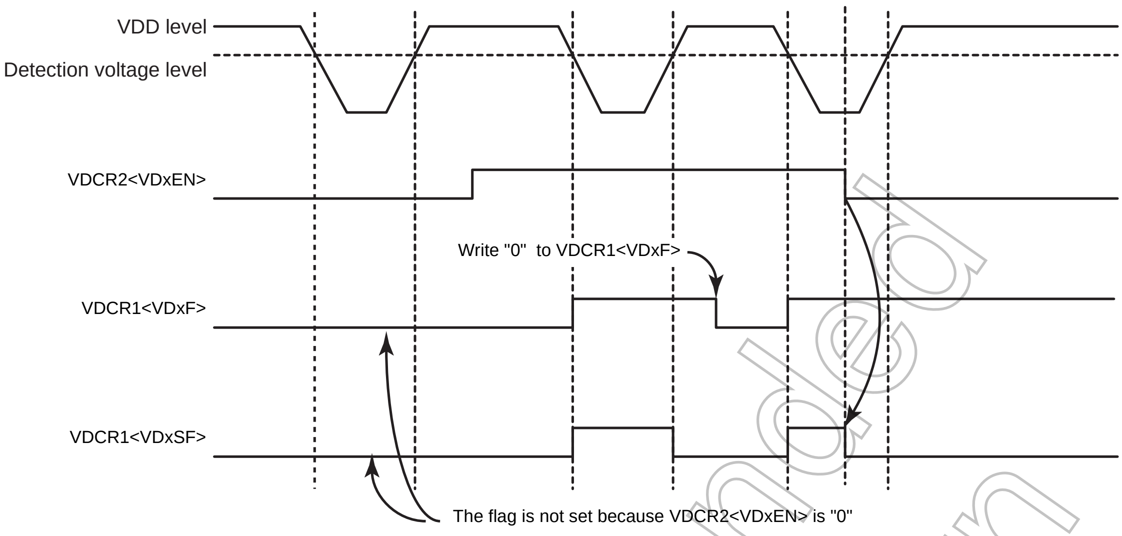


Figure 7-4 Changes in the Voltage Detection Flag and the Voltage Detection Status Flag

7.4 Register Settings

7.4.1 Setting procedure when the operation mode is set to generate INTVLTD interrupt request signals

When the operation mode is set to generate INTVLTD interrupt request signal, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1<VDxLVL>(x=1 to 2).
3. Clear VDCR2<VDxMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.
4. Set VDCR2<VDxEN> to "1" to enable the voltage detection operation.
5. Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1<VDxSF> is "0".
7. Clear the voltage detection circuit interrupt latch to "0" and set the interrupt enable flag to "1" to enable interrupts.

Note: When the supply voltage (VDD) is close to the detection voltage (VDxLVL), voltage detection request signals may be generated frequently. If this may pose any problem, execute appropriate wait processing depending on fluctuations in the system power supply and clear the interrupt latch before returning from the INTVLTD interrupt service routine.

To disable the voltage detection circuit while it is enabled with the INTVLTD interrupt request, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Clear VDCR2<VDxEN> to "0" to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

7.4.2 Setting procedure when the operation mode is set to generate voltage detection reset signals

When the operation mode is set to generate voltage detection reset signals, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1<VDxLVL>(x=1 to 2).
3. Clear VDCR2<VDxMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.
4. Set VDCR2<VDxEN> to "1" to enable the voltage detection operation.
5. Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1<VDxSF> is "0".
7. Clear VDCR1<VDxF> to "0".
8. Set VDCR2<VDxMOD> to "1" to set the operation mode to generate voltage detection reset signals.

Note 1: VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. If the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL) in the period from release of the voltage detection reset until clearing of VDCR2<VDxEN> to "0", a voltage detection reset signal is generated immediately.

Note 2: The voltage detection reset signals are generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

To disable the voltage detection circuit while it is enabled with the voltage detection reset, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".

2. Clear VDCR2<VDxMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.
3. Clear VDCR2<VDxEN> to "0" to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

7.5 Revision History

Rev	Description
RA001	" Voltage detection control register 1" Revised VD1LVL and VD2LVL.
RA002	Revised from VDCR2<VDxLVL> to VDCR1<VDxLVL>
RA003	"7.4.1 Setting procedure when the operation mode is set to generate INTVLTD interrupt request signals" Added description to disable the voltage detection circuit.
	"7.4.2 Setting procedure when the operation mode is set to generate voltage detection reset signals" Added description to disable the voltage detection circuit. Added step 7.
	" Voltage detection control register 2" Added Note 3.
	"7.3.5 Selecting the STOP mode release signal" Added Note 4.
RA004	Revised "voltage detection interrupt" to "INTVLTD interrupt". Revised initial value of VDCR1<VD2LVL> "00" to "10".
RB000	Deleted VDCR2<SRSS> function. Added note of INTVLTD interrupt. "7.3.2 Selecting the voltage detection operation mode" Revised description.

8. I/O Ports

The TMP89CM42 has 8 parallel input/output ports (40 pins) as follows:

Table 8-1 List of I/O Ports

Port name	Pin name	Number of pins	Input/output	Secondary functions
Port P0	P03 to P00 (Note)	4 (Note)	Input/output	Also used as the high-frequency oscillator connection pin and the low-frequency oscillator connection pin
Port P1	P13 to P10	4	Input/output	Also used as the external reset input, the external interrupt input and the STOP mode release signal input
Port P2	P27 to P20	8	Input/output	Also used as the UART input/output, the serial interface input/output and the serial bus interface input/output
Port P4	P47 to P40	8	Input/output	Also used as the analog input and the key-on wakeup input
Port P7	P77 to P70	8	Input/output	Also used as the timer counter input/output, the divider output and the external interrupt input
Port P8	P81 to P80	2	Input/output	Also used as the timer counter input/output
Port P9	P91 to P90	2	Input/output	Also used as the UART input/output
Port PB	PB7 to PB4	4	Input/output	Also used as the UART input/output and the serial interface input/output

Note: P00 and P01 pins can not be used for the I/O port, because they should be connected with the high frequency OSC input.

Each output port contains a latch, which holds the output data. No input port has a latch, so the external input data should be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 8-1 shows input/output timing examples.

External data is read from an I/O port in the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program. Data is output to an I/O port in the next cycle of the write cycle during execution of the write instruction.

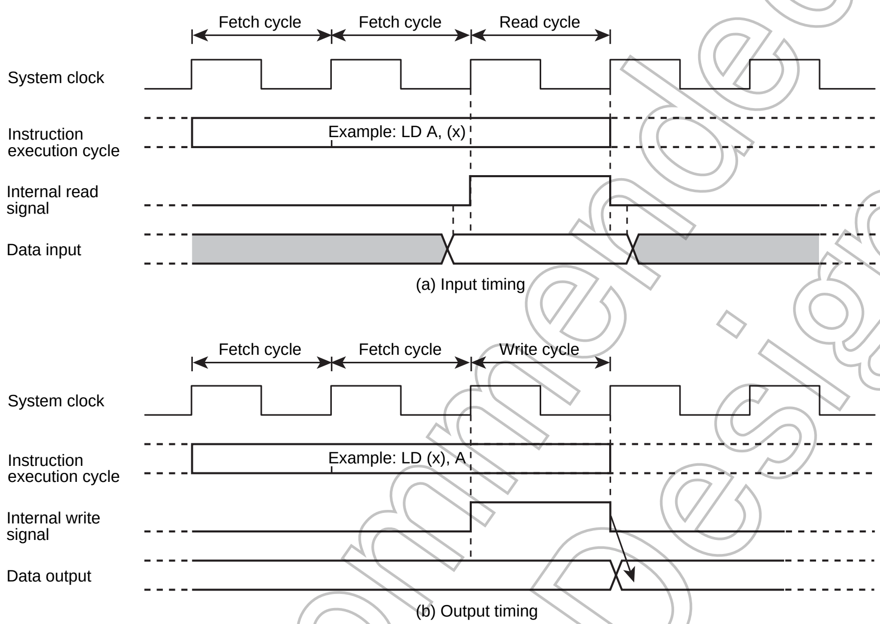


Figure 8-1 Input/Output Timing (Example)

Note: The positions of the read and write cycles may vary, depending on the instruction.

8.1 I/O Port Control Registers

The following control registers are used for I/O ports. (The port number is indicated in place of x.) Registers that can be set vary depending on the port. For details, refer to the description of each port.

- PxDR register

  This is the register for setting output data. When a port is set to the "output mode", the value specified at PxDR is output from the port.

- PxPRD register

  This is the register for reading input data. When a port is set to the "input mode", the current port input status can be read by reading PxPRD.

- PxCR register

  This register switches a port between input and output. A port can be switched between the "input mode" and the "output mode".

- PxFC register

  This register enables the secondary function output of each port. The secondary function output of each port can be enabled or disabled.

- PxOUTCR register

  This register switches the port output between the C-MOS output and the open drain output.

- PxPU register

  This register determines whether or not the built-in pull-up resistor is connected when a port is used in the input mode or as the open drain output.

8.2 List of I/O Port Settings

For the setting methods for individual I/O ports, refer to the following table.

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
Port P0	P03 to P00	Port input	0	Without register	0	
		Port output	1		0	
	P03	XTOUT	*		Without register	
	P02	XTIN	*		1	
	P01	XOUT	*		Without register	
	P00	XIN	*		1	
Port P1	P13 to P11	Port input	0	Without register	Without register	
		Port output	1			
	P10	Port input	0			Note 1
	P10	Port output	1			Note 1
	P13	INT1 input	0			
	P12	$\overline{INT0}$ input	0			
	P11	$\overline{INT5}$ input	0			
	P11	$\overline{STOP}$ input	0			
	P10	$\overline{RESET}$ input	*			Note 1
Port P2	P27 to P20	Port input	0	*	*	
		Port output	1	**	0	
	P25	SCLK0 input	0	*	*	SERSEL<SRSEL0>="01"
		SCLK0 output	1	**	1	SERSEL<SRSEL0>="01"
	P24	SCL0 input/output	1	Without register	1	SERSEL<SRSEL0>="*0"
		SI input	0		*	SERSEL<SRSEL0>="01"
	P23	SDA0 input/output	1	Without register	1	SERSEL<SRSEL0>="*0"
		SO output	1		1	SERSEL<SRSEL0>="01"
	P22	SCLK0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
		SCLK0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
	P21	RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="0"
		TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="1"
		SI0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
	P20	TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="0"
		RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="1"
		SO0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
Port P4	P47 to P40	Port input	0	Without register	0	
		Port output	1		0	
		AIN7 to AIN0	0		1	
		KWI7 to KWI4	*		*	KWUCR1
		KWI3 to KWI0	*		*	KWUCR0
Port P7	P77 to P70	Port input	0	Without register	*	
		Port output	1		0	
	P77	INT4 input	0		Without register	
	P76	INT3 input	0		Without register	
	P75	INT2 input	0		Without register	
	P74	$\overline{\text{DVO}}$ output	1		1	
	P73	TCA1 input	0		*	
		$\overline{\text{PPGA1}}$ output	1		1	
	P72	TCA0 input	0		*	SERSEL<TCA0SEL>="00"
		$\overline{\text{PPGA0}}$ output	1		1	
	P71	TC01 input	0		*	
		$\overline{\text{PPG01}}$ / $\overline{\text{PWM01}}$ output	1		1	
	P70	TC00 input	0		*	
		$\overline{\text{PPG00}}$ / $\overline{\text{PWM00}}$ output	1		1	
Port P8	P81 to P80	Port input	0	Without register	*	
		Port output	1		0	
	P81	TC03 input	0		*	
		$\overline{\text{PPG03}}$ / $\overline{\text{PWM03}}$ output	1		1	
	P80	TC02 input	0		*	
		$\overline{\text{PPG02}}$ / $\overline{\text{PWM02}}$ output	1		1	
Port P9	P92 to P90	Port input	0	*	*	
		Port output	1	**	0	
	P91	RXD1 input	0	*	0	UATCNG<UAT1IO>="0"
		TXD1 output	1	**	1	UATCNG<UAT1IO>="1"
	P90	TXD1 output	1	**	1	UATCNG<UAT1IO>="0"
		RXD1 input	0	*	0	UATCNG<UAT1IO>="1"
Port PB	PB7 to PB4	Port input	0	*	*	
		Port output	1	**	0	
	PB6	SCLK0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"
		SCLK0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"
	PB5	RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="0"
		TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="1"
		SI0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
	PB4	TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="0"
		RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="1"
		SO0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"

Note 1: After the power is turned on, pin P10 serves as an external reset input. To use pin P10 as a port, refer to "How to use the external reset input pin as a port".

Note 2: About SERSEL, please refer to "8.4 Serial Interface Selecting Function".

Note 3: The symbol and numeric characters in the table have the following meanings:

Symbol and numeric characters	Meaning
0	Set "0".
1	Set "1".
*	Don't care (Operation is the same whether "1" or "0" is selected.)
**	The sink open drain output or the C-MOS output can be selected.
Without register	There is no register that corresponds to the bit.

8.3 I/O Port Registers

8.3.1 Port P0 (P03 to P00)

Port P0 is a 4-bit input/output port that can be set to input or output for each bit individually, and it is also used as the high-frequency oscillation connection pin and the low-frequency oscillation connection pin.

Port P0 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

Table 8-3 Port P0

	-	-	-	-	P03	P02	P01	P00
Secondary function	-	-	-	-	XTOUT	XTIN	XOUT	XIN

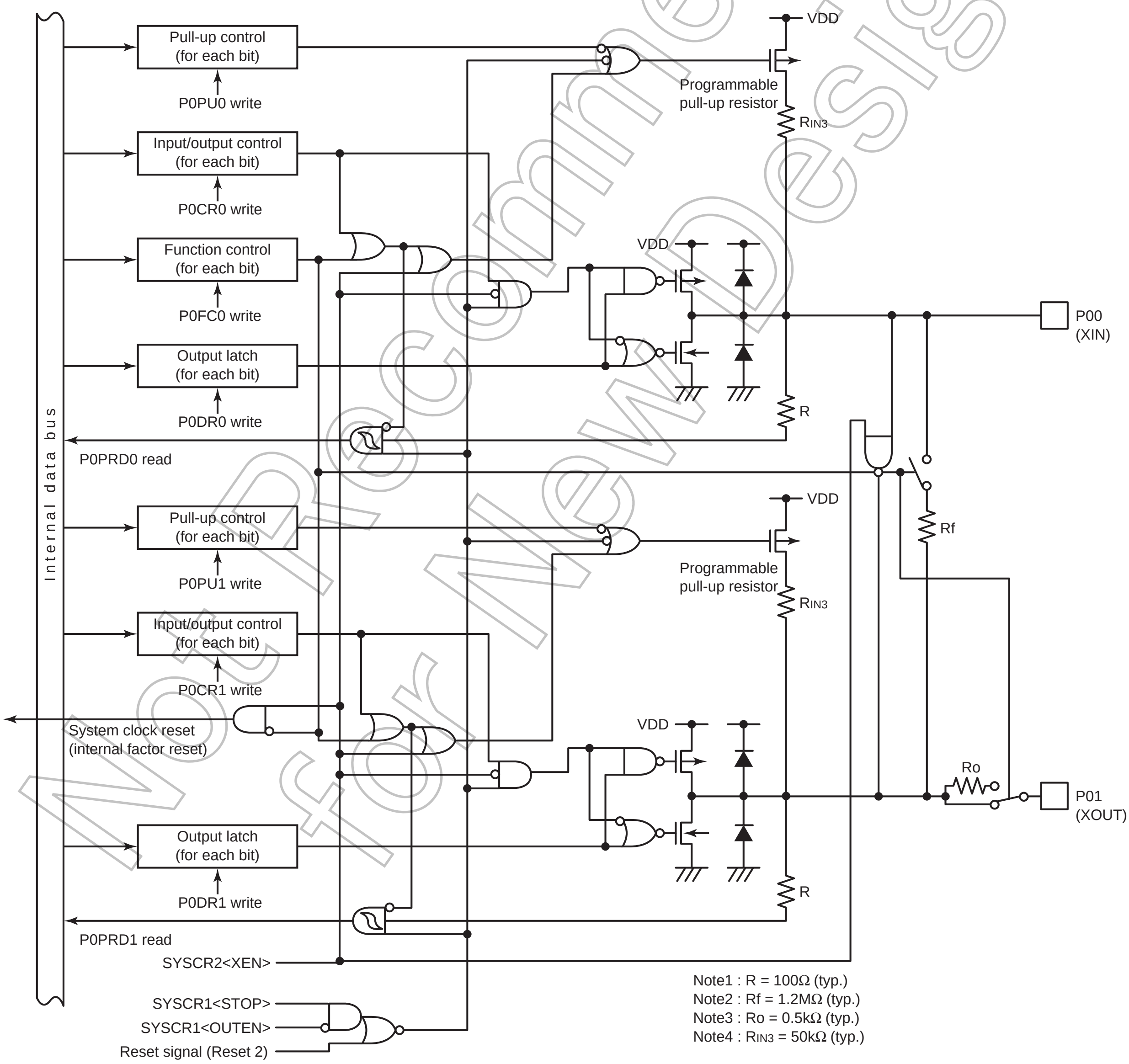


Figure 8-2 Port P0 (P00, P01)

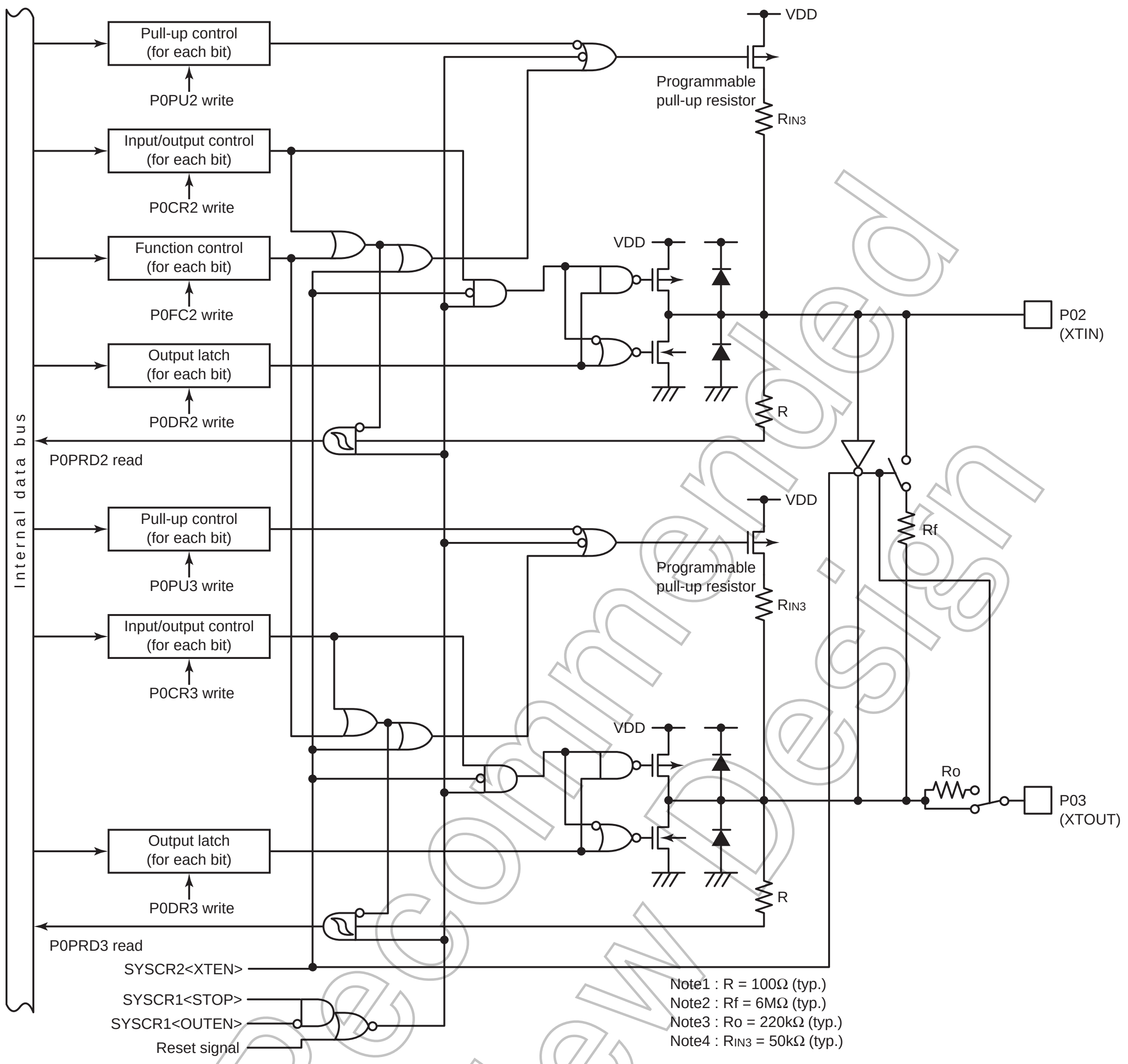


Figure 8-3 Port P0 (P02, P03)

Port P0 output latch

P0DR (0x0000)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P03	P02	P01	P00
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected.			
	1:					Outputs H level when the output mode is selected.			

Port P0 input/output control

P0CR (0x0F1A)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P0CR3	P0CR2	P0CR1	P0CR0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			
	1:					Output mode (port output)			

Note: P0CR1 and P0CR0 must be clear to "0".

Port P0 function control

P0FC (0x0F34)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	P0FC2	-	P0FC0
Read/Write		R	R	R	R	R	R/W	R	R/W
After reset		0	0	0	0	0	0	0	1
Function	0:						Port function		Port function
	1:						XTIN (I)		XIN (I)

Note 1: When SYSCR2<XEN> is "1", setting P0FC0 to "0" generates a system clock (internal factor) reset. Normally, ports P00 or P01 are not used as ports, so P0FC0 must be set to "1".

Note 2: Symbol "I" means secondary function input

Port P0 built-in pull-up resistor control

P0PU (0x0F27)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P0PU2	P0PU2	P0PU1	P0PU0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					The built-in pull-up resistor is not connected.			
	1:					The built-in pull-up resistor is connected. (The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected.)			

Port P0 input data

P0PRD (0x000D)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	P0PRD3	P0PRD2	P0PRD1	P0PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	*	*	*	*
Function					If the port is in the input mode, the contents of the port are read. If not, "0" is read.			

Table 8-4 P0PRD Read Value (P00 to P01)

Set condition		P0PRDi read value
P0FC0	P0CRi	
*	1	"0"
1	*	"0"
0	0	Contents of port

Note 1: * : Don't care
Note 2: i = 0, 1

Table 8-5 P0PRD Read Value (P02 to P03)

Set condition		P0PRDj read value
P0FC2	P0CRj	
*	1	"0"
1	*	"0"
0	0	Contents of port

Note 1: * : Don't care
Note 2: j = 2, 3

8.3.2 Port P1 (P13 to P10)

Port P1 is a 4-bit input/output port that can be set to input or output for each bit individually, and is also used as the external interrupt input, the STOP mode release signal input and the external reset input.

Port P1 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

After reset, pin P10 serves as the external reset input. To use pin P10 as a port, refer to "How to use external reset input pin as a port".

Table 8-6 Port P1

	-	-	-	-	P13	P12	P11	P10
Secondary function	-	-	-	-	INT1	$\overline{\text{INT0}}$	$\overline{\text{INT5}}$ $\overline{\text{STOP}}$	$\overline{\text{RESET}}$

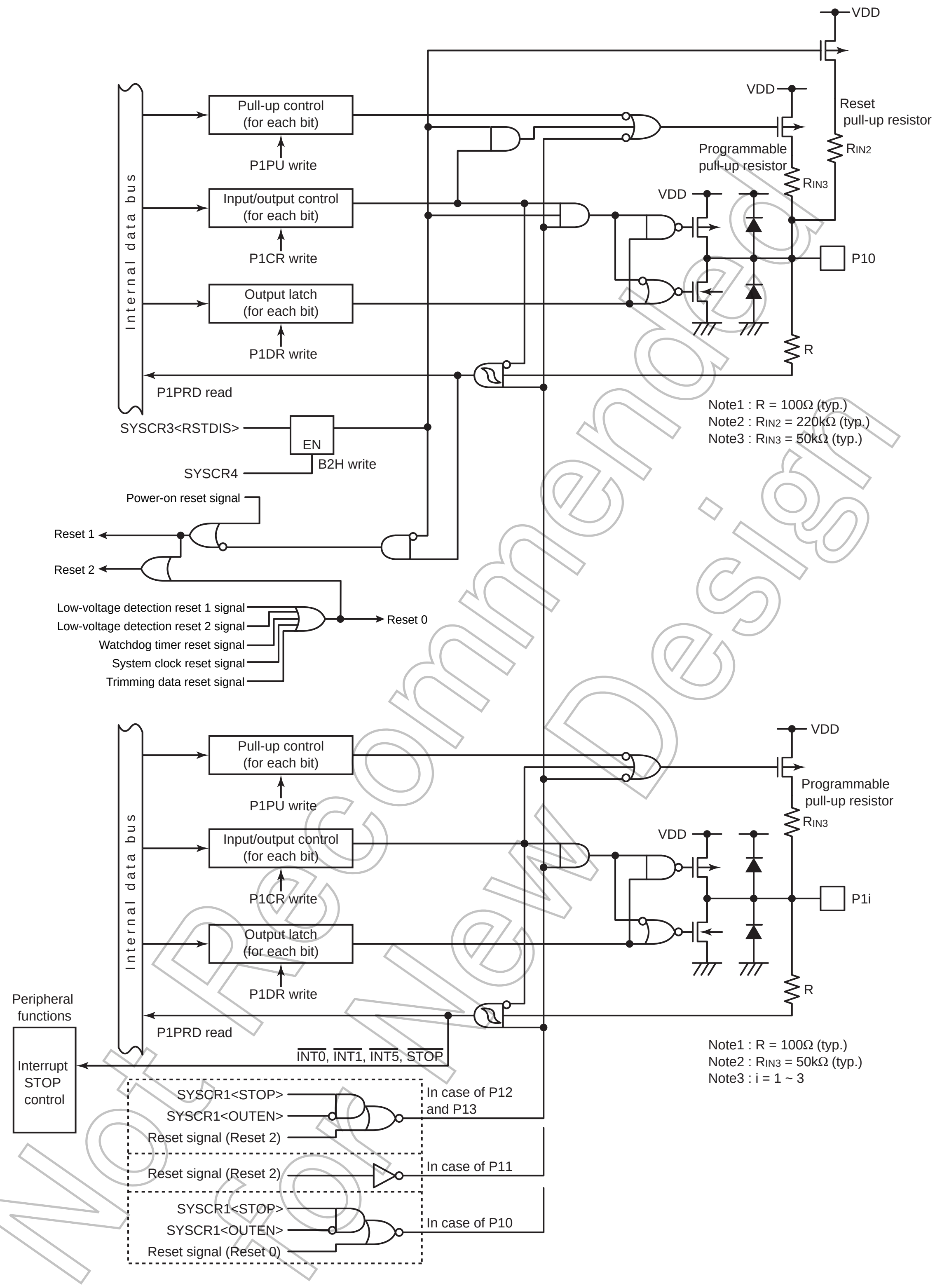


Figure 8-4 Port P1

Port P1 output latch

P1DR (0x0001)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P13	P12	P11	P10
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected.			
	1:					Outputs H level when the output mode is selected.			

Port P1 input/output control

P1CR (0x0F1B)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P1CR3	P1CR2	P1CR1	P1CR0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			
						INT1 (I)	$\overline{\text{INT0}}$ (I)	$\overline{\text{INT5}}$ (I) $\overline{\text{STOP}}$ (I)	-
	1:					Output mode (port output)			

Note: Symbol "I" means secondary function input

Port P1 built-in pull-up resistor control

P1PU (0x0F28)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P1PU4	P1PU2	P1PU1	P1PU0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					The built-in pull-up resistor is not connected.			
	1:					The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.)			

Port P1 input data

P1PRD (0x000E)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	P1PRD3	P1PRD2	P1PRD1	P1PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	*	*	*	*
Function					If the port is in the input mode, the contents of the port are read. If not, "0" is read.			

Table 8-7 P1PRD Read Value

Set condition P1CRi	P1PRDi read value
0	Contents of port
1	"0"

Note 1: * : Don’t care

Note 2: i = 0 to 3

8.3.3 Port P2 (P27 to P20)

Port P2 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial bus interface input/output, the serial interface input/output, the UART input/output and the on-chip debug function.

The output circuit has the P-channel output control function and either the sink open drain output or the C-MOS output can be selected. Port P2 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode or as a sink open drain output.

When this port is used as the serial bus interface, the serial interface or the UART, setting for serial interface selecting function is also needed. For details, refer to "8.4 Serial Interface Selecting Function".

Table 8-8 Port P2

	P27	P26	P25	P24	P23	P22	P21	P20
Secondary function	-	-	SCLK0	SI0 SCL0	SO0 SDA0	SCLK0	SI0 RXD0 TXD0	SO0 TXD0 RXD0

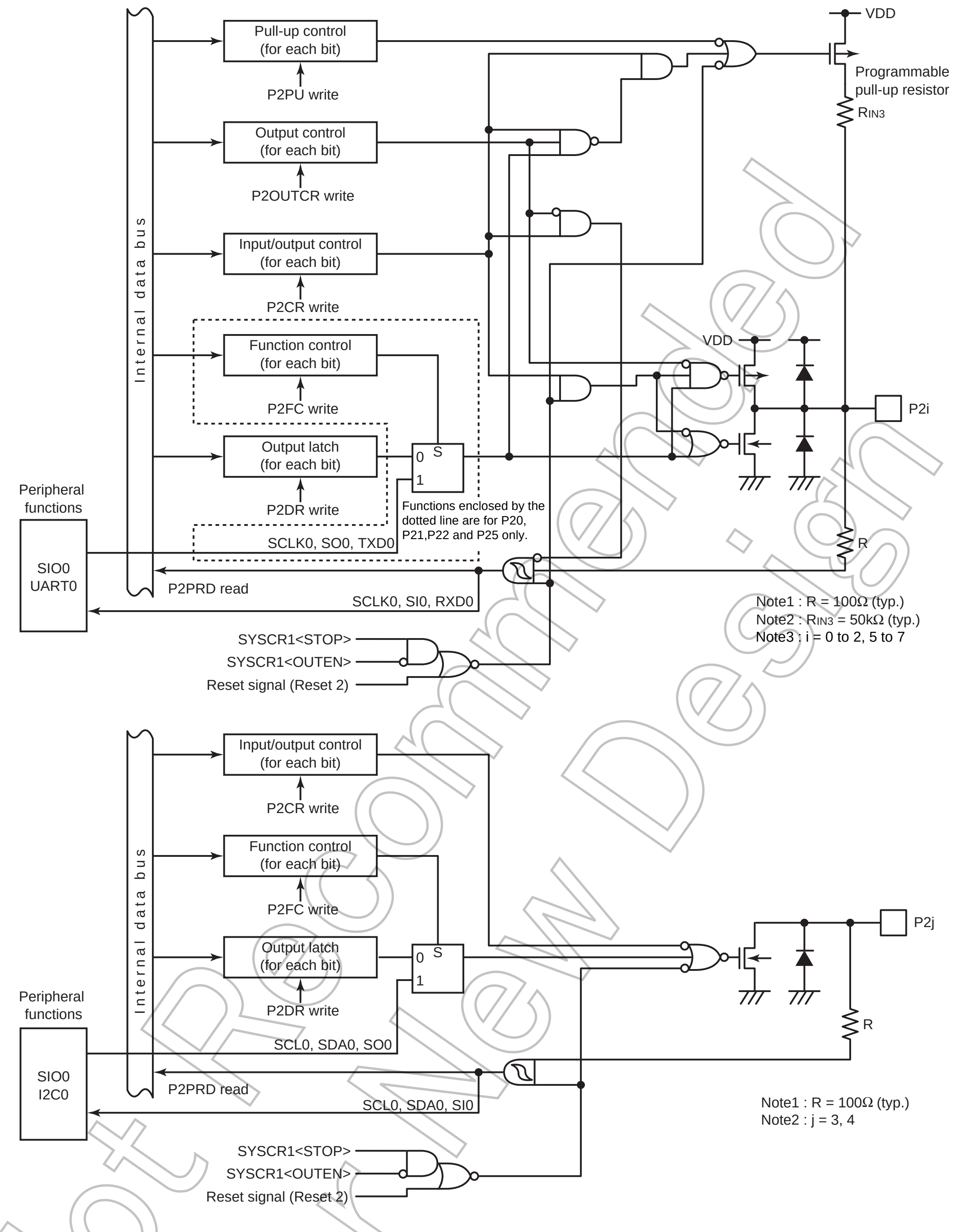


Figure 8-5 Port P2

Port P2 output latch

P2DR (0x0002)

		7	6	5	4	3	2	1	0
Bit Symbol		P27	P26	P25	P24	P23	P22	P21	P20
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected. (Serves as Hi-Z or pull-up depending on settings of P2OUTCR and P2PU.)							

Port P2 input/output control

P2CR (0x0F1C)

		7	6	5	4	3	2	1	0
Bit Symbol		P2CR7	P2CR6	P2CR5	P2CR4	P2CR3	P2CR2	P2CR1	P2CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
		-	-	SCLK0 (I)	SI0 (I)	-	SCLK0 (I)	RXD0 (I) SI0 (I)	RXD0 (I)
	1:	Output mode (port output)							
		-	-	SCLK0 (O)	SCL0 (I/O)	SDA0 (I/O) SO (O)	SCLK0 (O)	TXD0(O)	TXD0 (O) SO0 (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output. Symbol "I/O" means secondary function input/output

Port P2 function control

P2FC (0x0F36)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2FC5	P2FC4	P2FC3	P2FC2	P2FC1	P2FC0
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:			Port function					
	1:			SCLK0 (O)	SCL0 (I/O)	SDA0 (I/O) SO0 (O)	SCLK0 (O)	TXD0 (O)	TXD0 (O) SO0 (O)

Port P2 output control

P2OUTCR (0x0F43)

		7	6	5	4	3	2	1	0
Bit Symbol		P2OUT7	P2OUT6	P2OUT5	-	-	P2OUT2	P2OUT1	P2OUT0
Read/Write		R/W	R/W	R/W	R	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	C-MOS output					C-MOS output		
	1:	Open drain output					Open drain output		

Port P2 built-in pull-up resistor control

P2PU (0x0F29)

		7	6	5	4	3	2	1	0
Bit Symbol		P2PU7	P2PU6	P2PU5	-	-	P2PU2	P2PU1	P2PU0
Read/Write		R/W	R/W	R/W	R	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.					The built-in pull-up resistor is not connected.		
	1:	The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.)					The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.)		

Port P2 input data

P2PRD (0x000F)

	7	6	5	4	3	2	1	0
Bit Symbol	P2PRD7	P2PRD6	P2PRD5	P2PRD4	P2PRD3	P2PRD2	P2PRD1	P2PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.			The contents of the port are read without condition.		If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.		

Table 8-9 P2PRD Read Value (P20 to P22, P25 to P27)

Set condition		P2PRDi read value
P2CRi	P2OUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

Note 1: * : Don’t care

Note 2: i = 0 to 2, 5 to 7

8.3.4 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the analog input and the key-on wakeup input.

Port P4 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

Table 8-10 Port P4

	P47	P46	P45	P44	P43	P42	P41	P40
Secondary function	AIN7 KWI7	AIN6 KWI6	AIN5 KWI5	AIN4 KWI4	AIN3 KWI3	AIN2 KWI2	AIN1 KWI1	AIN0 KWI0

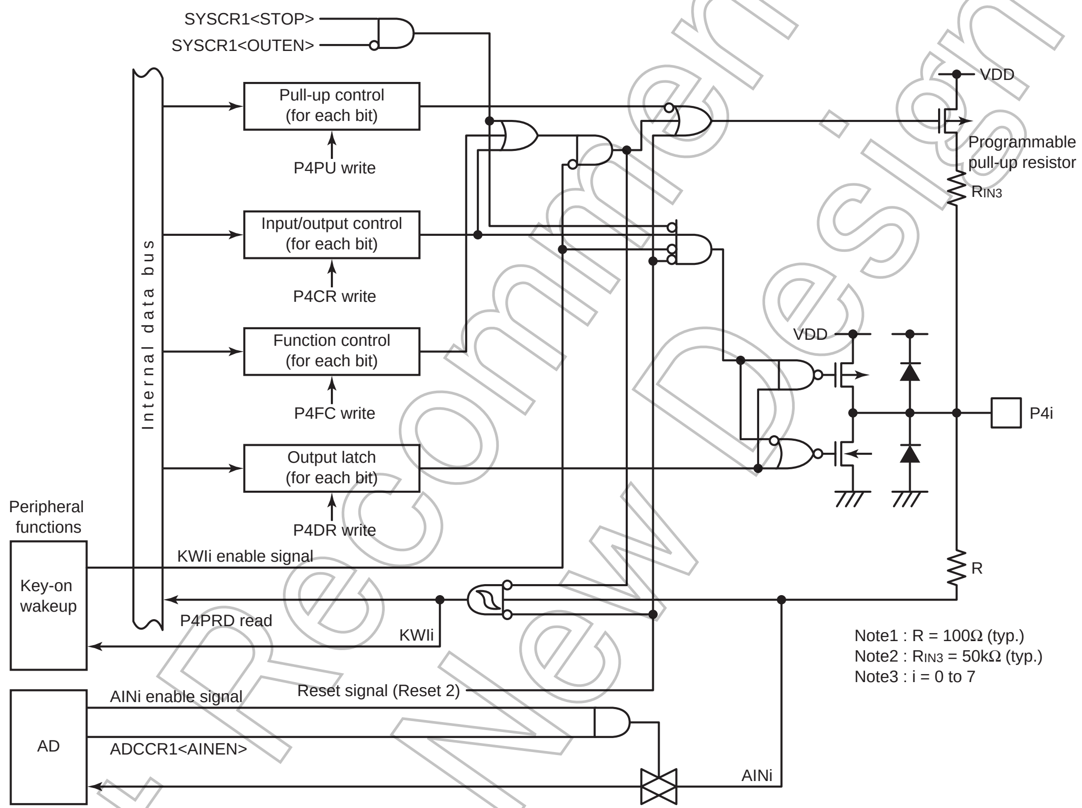


Figure 8-6 Port P4

Port P4 output latch

P4DR (0x0004)

		7	6	5	4	3	2	1	0
Bit Symbol		P47	P46	P45	P44	P43	P42	P41	P40
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

Port P4 input/output control

P4CR (0x0F1E)

		7	6	5	4	3	2	1	0
Bit Symbol		P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
		AIN7 (I)	AIN6 (I)	AIN5 (I)	AIN4 (I)	AIN3 (I)	AIN2 (I)	AIN1 (I)	AIN0 (I)
	1:	Output mode (port output)							

Note 1: Symbol "I" means secondary function input.

Note 2: When the key-on wakeup input (KWIi) is enabled (KWUCRm<KWnEN>="1"), there is no need to set P4CRi. (i=7 to 0, m=1 to 0, n=3 to 0)

Port P4 function control

P4FC (0x0F38)

		7	6	5	4	3	2	1	0
Bit Symbol		P4FC7	P4FC6	P4FC5	P4FC4	P4FC3	P4FC2	P4FC1	P4FC0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Port function							
	1:	AIN7 (I)	AIN6 (I)	AIN5 (I)	AIN4 (I)	AIN3 (I)	AIN2 (I)	AIN1 (I)	AIN0 (I)

Note 1: When the key-on wakeup input (KWIi) is enabled, there is no need to set P4FCi.

Port P4 built-in pull-up resistor control

P4PU (0x0F2B)

		7	6	5	4	3	2	1	0
Bit Symbol		P4PU7	P4PU6	P4PU5	P4PU4	P4PU3	P4PU2	P4PU1	P4PU0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.							
	1:	The built-in pull-up resistor is connected. (The resistor is connected only when the key-on wakeup input (KWIi) is enabled or the port is used in the input mode (P4FCi="0" and P4CRi="0"). Under any other conditions, setting to "1" does not make the resistor connected.)							

Port P4 input data

P4PRD (0x0011)

	7	6	5	4	3	2	1	0
Bit Symbol	P4PRD7	P4PRD6	P4PRD5	P4PRD4	P4PRD3	P4PRD2	P4PRD1	P4PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-11 P4PRD Read Value

Set condition		P4PRDi read value
P4CRi	P4FCi	
0	0	Contents of port
*	1	"0"
1	*	"0"

Note 1: * : Don't care

Note 2: i = 0 to 7

8.3.5 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the external interrupt input, the divider output and the timer counter input/output.

Table 8-12 Port P7

	P77	P76	P75	P74	P73	P72	P71	P70
Secondary function	INT4	INT3	INT2	$\overline{\text{DVO}}$	$\overline{\text{PPGA1}}$ TCA1	$\overline{\text{PPGA0}}$ TCA0	$\overline{\text{PPG01}}$ $\overline{\text{PWM01}}$ TC01	$\overline{\text{PPG00}}$ $\overline{\text{PWM00}}$ TC00

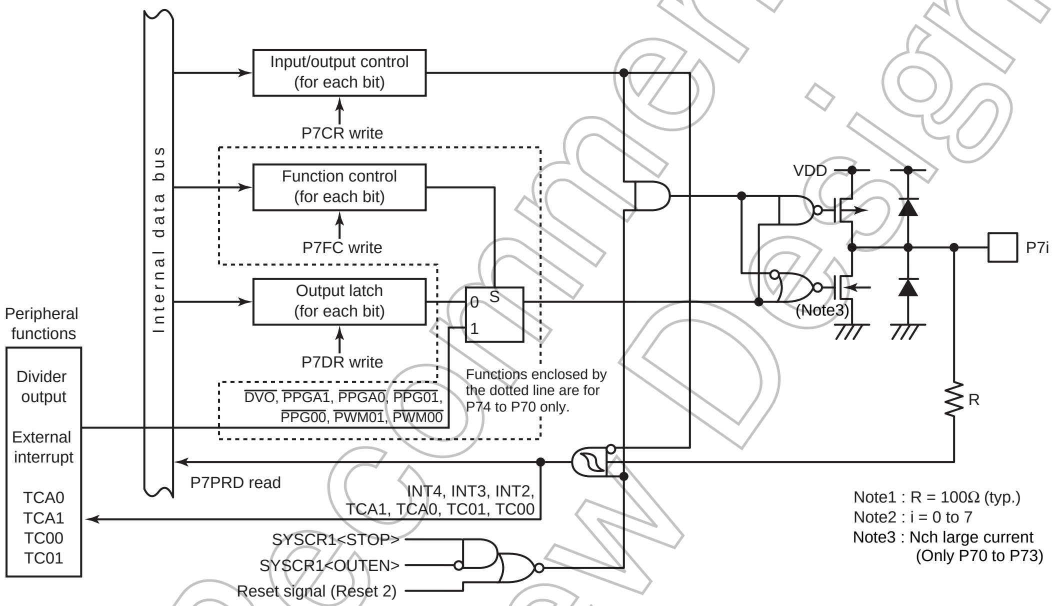


Figure 8-7 Port P7

Port P7 output latch

P7DR (0x0007)

		7	6	5	4	3	2	1	0
Bit Symbol		P77	P76	P75	P74	P73	P72	P71	P70
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected							
	1:	Outputs H level when the output mode is selected							

Port P7 input/output control

P7CR (0x0F21)

		7	6	5	4	3	2	1	0
Bit Symbol		P7CR7	P7CR6	P7CR5	P7CR4	P7CR3	P7CR2	P7CR1	P7CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
		INT4 (I)	INT3 (I)	INT2 (I)	-	TCA1 (I)	TCA0 (I)	TC01 (I)	TC00 (I)
	1:	Output mode (port output)							
		-	-	-	$\overline{\text{DVO}}$ (O)	$\overline{\text{PPGA1}}$ (O)	$\overline{\text{PPGA0}}$ (O)	$\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)	$\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P7 function control

P7FC (0x0F3B)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	P7FC3	P7FC3	P7FC2	P7FC1	P7FC0
Read/Write		R	R	R	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:				Port function				
	1:				$\overline{\text{DVO}}$ (O)	$\overline{\text{PPGA1}}$ (O)	$\overline{\text{PPGA0}}$ (O)	$\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)	$\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)

Port P7 input data

P7PRD (0x0014)

	7	6	5	4	3	2	1	0
Bit Symbol	P7PRD7	P7PRD6	P7PRD5	P7PRD4	P7PRD3	P7PRD2	P7PRD1	P7PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-13 P7PRD Read Value

Set condition P7CRi	P7PRDi read value
0	Contents of port
1	"0"

Note 1: * : Don't care
Note 2: i = 0 to 7

8.3.6 Port P8 (P81 to P80)

Port P8 is a 2-bit input/output port that can be set to input or output for each bit individually, and it is also used as the timer counter input/output.

Table 8-14 Port P8

							P81	P80
Secondary function	-	-	-	-	-	-	$\overline{\text{PPG03}}$ $\overline{\text{PWM03}}$ TC03	$\overline{\text{PPG02}}$ $\overline{\text{PWM02}}$ TC02

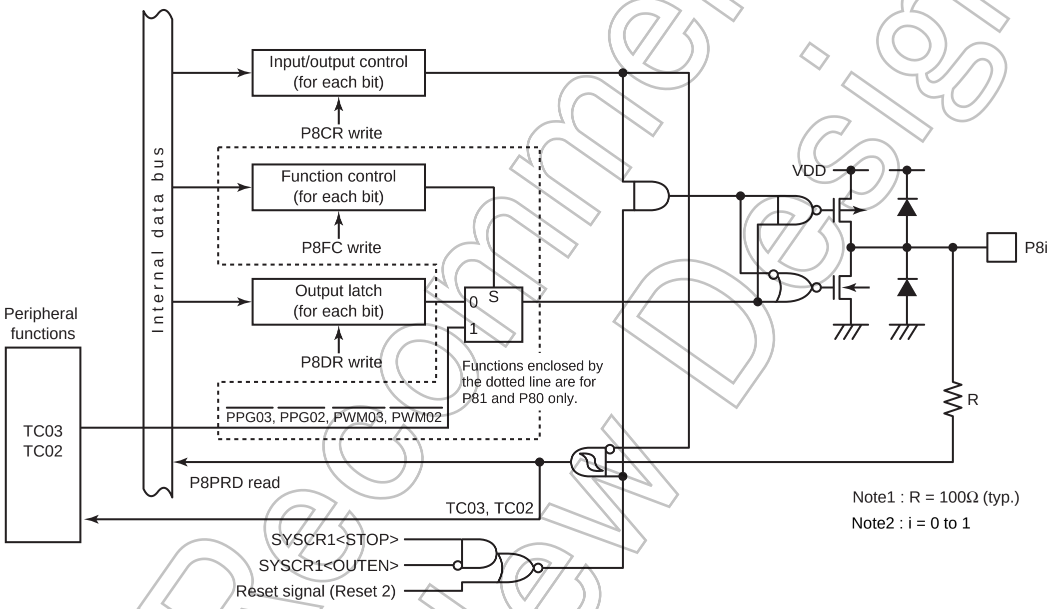


Figure 8-8 Port P8

Port P8 output latch

P8DR (0x0008)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P81	P80
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Outputs L level when the output mode is selected.	
	1:							Outputs H level when the output mode is selected.	

Port P8 input/output control

P8CR (0x0F22)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P8CR1	P8CR0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Input mode (port input)	
								TC03 (I)	TC02 (I)
	1:							Output mode (port output)	
								$\overline{\text{PPG03}}$ (O) $\overline{\text{PWM03}}$ (O)	$\overline{\text{PPG02}}$ (O) $\overline{\text{PWM02}}$ (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P8 function control

P8FC (0x0F3C)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P8FC1	P8FC0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Port function	
	1:							$\overline{\text{PPG03}}$ (O) $\overline{\text{PWM03}}$ (O)	$\overline{\text{PPG02}}$ (O) $\overline{\text{PWM02}}$ (O)

Port P8 input data

P8PRD (0x0015)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	P8PRD1	P8PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*
Function							If the port is used in the input mode, the contents of the port are read. If not, "0" is read.	

Table 8-15 P8PRD Read Value

Set condition	P8PRDi read value
P8CRi	
0	Contents of port
1	"0"

Note 1: * : Don’t care
Note 2: i = 0 to 1

8.3.7 Port P9 (P91 to P90)

Port P9 is a 2-bit input/output port that can be set to input or output for each bit individually, and it is also used as the UART.

The output circuit has the P-channel output control function and either the sink open drain output or the C-MOS output can be selected. Port P9 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode or as a sink open drain output.

When this port is used as the UART, setting for the serial interface selecting function is also needed. For details, refer to "8.4 Serial Interface Selecting Function".

Table 8-16 Port P9

							P91	P90
Secondary function	-	-	-	-	-	-	RXD1 TXD1	TXD1 RXD1

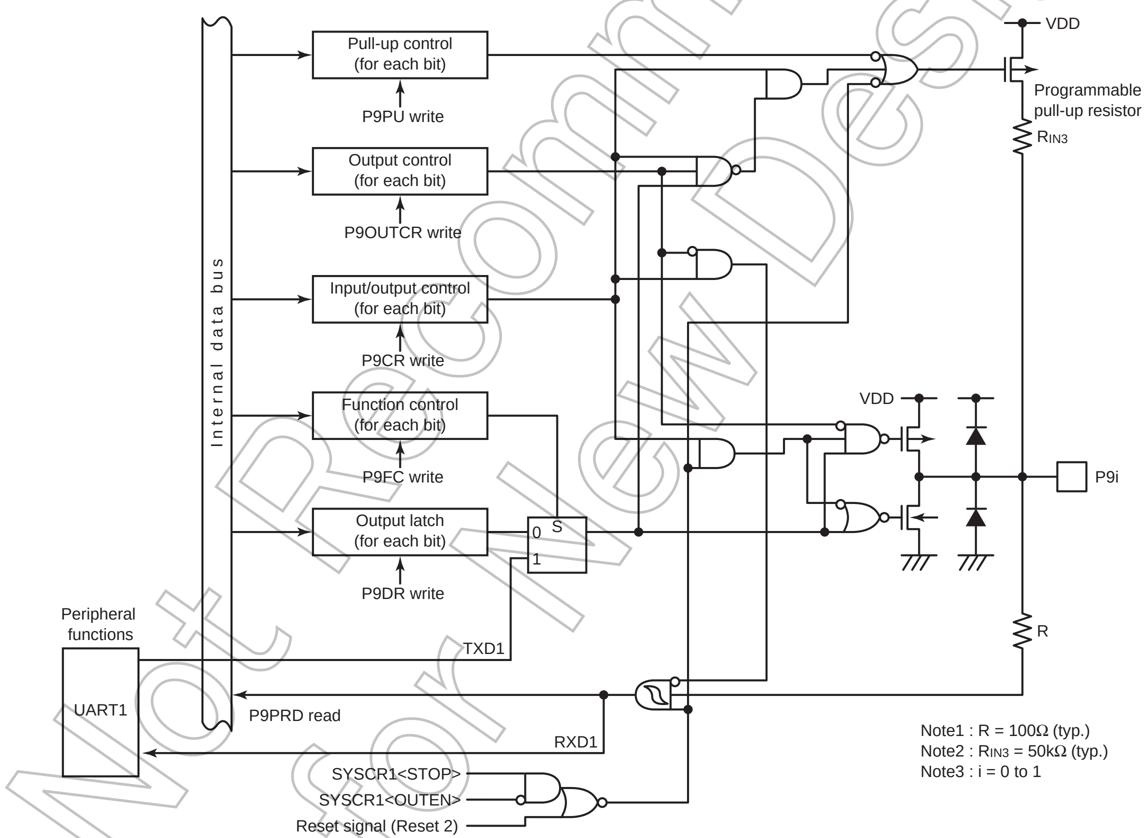


Figure 8-9 Port P9

Port P9 output latch

P9DR (0x0009)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P91	P90
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Outputs L level when the output mode is selected.	
	1:							Outputs H level when the output mode is selected. (Serves as Hi-Z or pull-up depending on settings of P9OUTCR and P9PU.)	

Port P9 input/output control

P9CR (0x0F23)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9CR1	P9CR0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Input mode (port input)	
								RXD1 (I)	RXD1 (I)
	1:							Output mode (port output)	
								TXD1 (O)	TXD1 (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P9 function control

P9FC (0x0F3D)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9FC1	P9FC0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Port function	
	1:							TXD1 (O)	TXD1 (O)

Port P9 output control

P9OUTCR (0x0F4A)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9OUT1	P9OUT0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							C-MOS output	
	1:							Open drain output	

Port P9 built-in pull-up resistor control

P9PU (0x0F30)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9PU1	P9PU0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							The built-in pull-up resistor is not connected.	
	1:							Note 1	

Note 1: The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.)

Port P9 input data

P9PRD (0x0016)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	P9PRD1	P9PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*
Function							If the port is used in the input mode or as the sink open drain output, the contents of the port are read. If not, "0" is read.	

Table 8-17 P9PRD Read Value

Set condition		P9PRDi read value
P9CRi	P9OUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

Note 1: * : Don’t care
Note 2: i = 0 to 1

8.3.8 Port PB (PB7 to PB4)

Port PB is an 4-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial interface input/output and the UART input/output.

The output circuit has the P-channel output control function and either the sink open drain output or the C-MOS output can be selected.

When this port is used as the serial interface or the UART, setting for serial interface selecting function is also needed. For details, refer to "8.4 Serial Interface Selecting Function".

Table 8-18 Port PB

	PB7	PB6	PB5	PB4	-	-	-	-
Secondary function	-	SCLK0	SI0 RXD0 TXD0	SO0 TXD0 RXD0	-	-	-	-

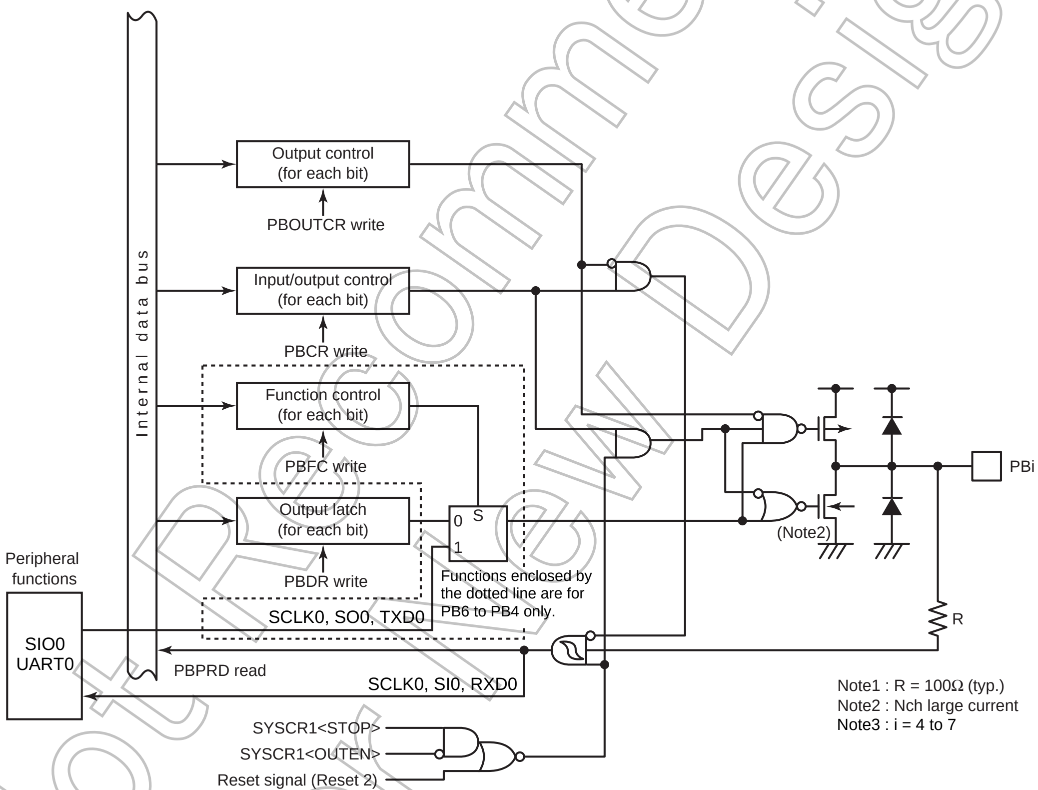


Figure 8-10 Port PB

Port PB output latch

PBDR (0x000B)

		7	6	5	4	3	2	1	0
Bit Symbol		PB7	PB6	PB5	PB4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

Port PB input/output control

PBCR (0x0F25)

		7	6	5	4	3	2	1	0
Bit Symbol		PBCR7	PBCR6	PBCR5	PBCR4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
	1:	Output mode (port output)							

Port PB function control

PBFC (0x0F3F)

		7	6	5	4	3	2	1	0
Bit Symbol		-	PBFC6	PBFC5	PBFC4	-	-	-	-
Read/Write		R	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:		Port function						
	1:		SCLK0 (O)	TXD0 (O)	TXD0 (O) SO0 (O)				

Port PB output control

PBOUTCR (0x0F4C)

		7	6	5	4	3	2	1	0
Bit Symbol		PBOUT7	PBOUT6	PBOUT5	PBOUT4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	C-MOS output							
	1:	Open drain output							

Port PB input data

PBPRD (0x0018)

	7	6	5	4	3	2	1	0
Bit Symbol	PBPRD7	PBPRD6	PBPRD5	PBPRD4				
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.							

Table 8-19 PBPRD Read Value

Set condition		PBPRDi read value
PBCRi	PBOUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

Note 1: * : Don’t care
Note 2: i = 4 to 7

8.4 Serial Interface Selecting Function

On the TMP89CM42, the built-in serial interface (SIO, UART and I[2]C) communication pins and interrupt source assignment can be changed. Two out of three functions, SIO0, UART0 and I2C0, can be used at the same time by using this selecting function.

The input pins of the 16-bit timer counter A0 input (TCA0 input) can be changed by using this selecting function.

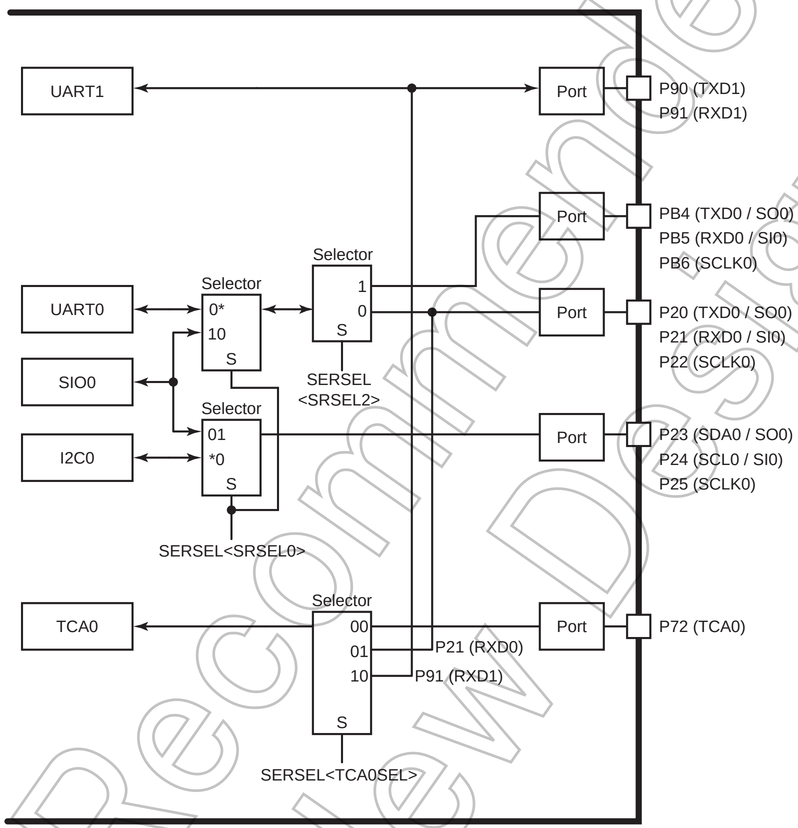


Figure 8-11 Serial Interface Selecting Function

Serial interface selection control register

SERSEL (0x0FCB)

	7	6	5	4	3	2	1	0
Bit Symbol	TCA0SEL			SRSEL2			SRSEL0	
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TCA0SEL	16-bit timer counter A0 input switching	00: P72 input (TCA0) 01: P21 input (also used as RXD0) 10: P91 input (also used as RXD1) 11: Reserved
SRSEL2	Select UART0/SIO0 input/output port	0: Select P20, P21, P22 1: Select PB4, PB5, PB6
SRSEL0	Serial interface selection 0	00: Select UART0, I2C0 01: Select UART0, SIO0 10: Select SIO0, I2C0 11: Reserved

Note 1: The operation for changing SERSEL must be executed while the applicable serial interface and timer counter operations are stopped. If SERSEL is switched during operation of these peripheral functions, each peripheral function may receive (transmit) unexpected data and operate improperly.

Note 2: It is recommended to clear the interrupt latch for the applicable serial interface immediately after changing SERSEL. Interrupt latches are common to INTRXD and INTSIO and to INTSBI and INTSIO. Therefore, if an interrupt occurs before or after SERSEL is switched, it is difficult to tell which function has caused the interrupt.

UART input/output change control register

UATCNG (0x0F57)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	UAT1IO	UAT0IO
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

			RXD pin		TXD pin	
UAT1IO	Select UART1 input/ output port	0: 1:	P91 P90		P90 P91	
UAT0IO	Select UART0 input/ output port		SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"	SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"
		0: 1:	P21 P20	PB5 PB4	P20 P21	PB4 PB5

Note 1: The operation for changing UATCNG must be executed while the applicable serial interface operations are stopped.

Table 8-20 Select input/output port and interrupt

<table>
<tr><th rowspan="3">SERSEL
<SRSEL0></th><th rowspan="3">SERSEL
<SRSEL2></th><th rowspan="3">UATCNG
<UAT0IO></th><th colspan="9">Port</th><th colspan="3" rowspan="2">Interrupt</th></tr>
<tr><th colspan="6">UART0/SIO0</th><th colspan="3">I2C0/SIO0</th></tr>
<tr><th>PB4</th><th>PB5</th><th>PB6</th><th>P20</th><th>P21</th><th>P22</th><th>P23</th><th>P24</th><th>P25</th><th>IL7</th><th>IL6</th><th>IL15</th></tr>
<tr><td rowspan="4">00:</td><td rowspan="2">0:</td><td>0:</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td>TXD0</td><td>RXD0</td><td rowspan="2">Note 1</td><td rowspan="4">SDA0</td><td rowspan="4">SCL0</td><td rowspan="4">Note 1</td><td rowspan="4">INTTXD0</td><td rowspan="4">INTRXD0</td><td rowspan="4">INTSBI0</td></tr>
<tr><td>1:</td><td>RXD0</td><td>TXD0</td></tr>
<tr><td rowspan="2">1:</td><td>0:</td><td>TXD0</td><td>RXD0</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td></tr>
<tr><td>1:</td><td>RXD0</td><td>TXD0</td></tr>
<tr><td rowspan="4">01:</td><td rowspan="2">0:</td><td>0:</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td>TXD0</td><td>RXD0</td><td rowspan="2">Note 1</td><td rowspan="4">SO0</td><td rowspan="4">SI0</td><td rowspan="4">SCLK0</td><td rowspan="4">INTTXD0</td><td rowspan="4">INTRXD0</td><td rowspan="4">INTSIO0</td></tr>
<tr><td>1:</td><td>RXD0</td><td>TXD0</td></tr>
<tr><td rowspan="2">1:</td><td>0:</td><td>TXD0</td><td>RXD0</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td><td rowspan="2">Note 1</td></tr>
<tr><td>1:</td><td>RXD0</td><td>TXD0</td></tr>
<tr><td rowspan="2">10:</td><td>0:</td><td>0 or 1:</td><td>Note 1</td><td>Note 1</td><td>Note 1</td><td>SO0</td><td>SI0</td><td>SCLK0</td><td rowspan="2">SDA0</td><td rowspan="2">SCL0</td><td rowspan="2">Note 1</td><td rowspan="2">-</td><td rowspan="2">INTSIO0</td><td rowspan="2">INTSBI0</td></tr>
<tr><td>1:</td><td>0 or 1:</td><td>SO0</td><td>SI0</td><td>SCLK0</td><td>Note 1</td><td>Note 1</td><td>Note 1</td></tr>
<tr><td>11:</td><td>0 or 1:</td><td>0 or 1:</td><td colspan="12">Reserved</td></tr>
</table>

Note 1: Can be used as a port. (Set the function register (PxFC) to "0".)

8.5 Revision History

Rev	Description
RA002	"Table 8-1 List of I/O Ports" Added description to P9 Port. "Table 8-2 List of I/O Port Settings" Added description of UART setting to P90 and P91. " Port P2 input/output control" Added RXD0(I) to P20. Added TXD0(O) to P21. "8.3.10 Port P8 (P81 to P80)" Added description about P8 Port. "8.3.11 Port P9 (P9741 to P90)" Deleted description about serial interface (SIO). "8.3.13 Port PB (PB75 to PB04)" Added detail description about PB port. "8.4 Serial Interface Selecting Function" Deleted description about SIO1 and UART1. "Figure 8-17 Serial Interface Selecting Function" Added PB Port. Deleted P92 and P94 Ports.Deleted SIO1. " Serial interface selection control register" Deleted SRSEL1. Revised SRSEL2 description from "output" to "input/output". Deleted Table 8-20.
RA003	"Figure 8-2 Port P0 (P00, P01)", "Figure 8-3 Port P0 (P02, P03)" Added damping resistor (Ro). "Figure 8-4 Port P1" Deleted STOP control from P11 pin input.
RA004	Defined symbol of programmable pull-up resistor to R_{IN3}. Defined symbol of reset pull-up resister to R_{IN2}.
	"8.3.2 Port P1 (P13 to P10)" Deleted description of "or as a sink open drain output" "8.3.6 Port P4 (P47 to P40)" Deleted description of "or as a sink open drain output"
RA005	"Figure 8-4 Port P1" Revised reset control signal.
RA006	"Table 8-2 List of I/O Port Settings" Added new character for PxOUTCR.
RA007	"Figure 8-4 Port P1" Revised block diagram. Added "Reset 0 or 2" to reset signal of each port block diagram.

9. Special Function Registers

The TMP89CM42 adopts the memory mapped I/O system, and all peripheral hardware data control and transfer operations are performed through the special function registers (SFR). SFR1 is mapped on addresses 0x0000 to 0x003F, SFR2 is mapped on addresses 0x0F00 to 0x0FFF, and SFR3 is mapped on addresses 0x0E40 to 0x0EBF.

9.1 SFR1 (0x0000 to 0x003F)

Table 9-1 SFR1 (0x0000 to 0x003F)

Address	Register Name	Address	Register Name
0x0000	P0DR	0x0020	SIO0SR
0x0001	P1DR	0x0021	SIO0BUF
0x0002	P2DR	0x0022	SBI0CR1
0x0003	Reserved	0x0023	SBI0CR2/SBI0SR2
0x0004	P4DR	0x0024	I2C0AR
0x0005	Reserved	0x0025	SBI0DBR
0x0006	Reserved	0x0026	T00REG
0x0007	P7DR	0x0027	T01REG
0x0008	P8DR	0x0028	T00PWM
0x0009	P9DR	0x0029	T01PWM
0x000A	Reserved	0x002A	T00MOD
0x000B	PBDR	0x002B	T01MOD
0x000C	Reserved	0x002C	T001CR
0x000D	P0PRD	0x002D	TA0DRAL
0x000E	P1PRD	0x002E	TA0DRAH
0x000F	P2PRD	0x002F	TA0DRBL
0x0010	Reserved	0x0030	TA0DRBH
0x0011	P4PRD	0x0031	TA0MOD
0x0012	Reserved	0x0032	TA0CR
0x0013	Reserved	0x0033	TA0SR
0x0014	P7PRD	0x0034	ADCCR1
0x0015	P8PRD	0x0035	ADCCR2
0x0016	P9PRD	0x0036	ADCDRL
0x0017	Reserved	0x0037	ADCDRH
0x0018	PBPRD	0x0038	DVOCR
0x0019	Reserved	0x0039	TBTCR
0x001A	UART0CR1	0x003A	EIRL
0x001B	UART0CR2	0x003B	EIRH
0x001C	UART0DR	0x003C	EIRE
0x001D	UART0SR	0x003D	EIRD
0x001E	TD0BUF/RD0BUF	0x003E	Reserved
0x001F	SIO0CR	0x003F	PSW

Note 1: Do not access reserved addresses by the program.

9.2 SFR2 (0x0F00 to 0x0FFF)

Table 9-2 SFR2 (0x0F00 to 0x0F7F)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0F00	Reserved	0x0F20	Reserved	0x0F40	Reserved	0x0F60	Reserved
0x0F01	Reserved	0x0F21	P7CR	0x0F41	Reserved	0x0F61	Reserved
0x0F02	Reserved	0x0F22	P8CR	0x0F42	Reserved	0x0F62	Reserved
0x0F03	Reserved	0x0F23	P9CR	0x0F43	P2OUTCR	0x0F63	Reserved
0x0F04	Reserved	0x0F24	Reserved	0x0F44	Reserved	0x0F64	Reserved
0x0F05	Reserved	0x0F25	PBCR	0x0F45	Reserved	0x0F65	Reserved
0x0F06	Reserved	0x0F26	Reserved	0x0F46	Reserved	0x0F66	Reserved
0x0F07	Reserved	0x0F27	P0PU	0x0F47	Reserved	0x0F67	Reserved
0x0F08	Reserved	0x0F28	P1PU	0x0F48	Reserved	0x0F68	Reserved
0x0F09	Reserved	0x0F29	P2PU	0x0F49	Reserved	0x0F69	Reserved
0x0F0A	Reserved	0x0F2A	Reserved	0x0F4A	P9OUTCR	0x0F6A	Reserved
0x0F0B	Reserved	0x0F2B	P4PU	0x0F4B	Reserved	0x0F6B	Reserved
0x0F0C	Reserved	0x0F2C	Reserved	0x0F4C	PBOUTCR	0x0F6C	Reserved
0x0F0D	Reserved	0x0F2D	Reserved	0x0F4D	Reserved	0x0F6D	Reserved
0x0F0E	Reserved	0x0F2E	Reserved	0x0F4E	Reserved	0x0F6E	Reserved
0x0F0F	Reserved	0x0F2F	Reserved	0x0F4F	Reserved	0x0F6F	Reserved
0x0F10	Reserved	0x0F30	P9PU	0x0F50	Reserved	0x0F70	Reserved
0x0F11	Reserved	0x0F31	Reserved	0x0F51	Reserved	0x0F71	Reserved
0x0F12	Reserved	0x0F32	Reserved	0x0F52	Reserved	0x0F72	Reserved
0x0F13	Reserved	0x0F33	Reserved	0x0F53	Reserved	0x0F73	Reserved
0x0F14	Reserved	0x0F34	P0FC	0x0F54	UART1CR1	0x0F74	POFFCR0
0x0F15	Reserved	0x0F35	Reserved	0x0F55	UART1CR2	0x0F75	POFFCR1
0x0F16	Reserved	0x0F36	P2FC	0x0F56	UART1DR	0x0F76	POFFCR2
0x0F17	Reserved	0x0F37	Reserved	0x0F57	UART1SR	0x0F77	POFFCR3
0x0F18	Reserved	0x0F38	P4FC	0x0F58	TD1BUF/RD1BUF	0x0F78	Reserved
0x0F19	Reserved	0x0F39	Reserved	0x0F59	Reserved	0x0F79	Reserved
0x0F1A	P0CR	0x0F3A	Reserved	0x0F5A	Reserved	0x0F7A	Reserved
0x0F1B	P1CR	0x0F3B	P7FC	0x0F5B	Reserved	0x0F7B	Reserved
0x0F1C	P2CR	0x0F3C	P8FC	0x0F5C	Reserved	0x0F7C	Reserved
0x0F1D	Reserved	0x0F3D	P9FC	0x0F5D	Reserved	0x0F7D	Reserved
0x0F1E	P4CR	0x0F3E	Reserved	0x0F5E	Reserved	0x0F7E	Reserved
0x0F1F	Reserved	0x0F3F	PBFC	0x0F5F	Reserved	0x0F7F	Reserved

Note 1: Do not access reserved addresses by the program.

Table 9-3 SFR2 (0x0F80 to 0x0FFF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0F80	Reserved	0x0FA0	Reserved	0x0FC0	Reserved	0x0FE0	ILL
0x0F81	Reserved	0x0FA1	Reserved	0x0FC1	Reserved	0x0FE1	ILH
0x0F82	Reserved	0x0FA2	Reserved	0x0FC2	Reserved	0x0FE2	ILE
0x0F83	Reserved	0x0FA3	Reserved	0x0FC3	Reserved	0x0FE3	ILD
0x0F84	Reserved	0x0FA4	Reserved	0x0FC4	KWUCR0	0x0FE4	Reserved
0x0F85	Reserved	0x0FA5	Reserved	0x0FC5	KWUCR1	0x0FE5	Reserved
0x0F86	Reserved	0x0FA6	Reserved	0x0FC6	VDCR1	0x0FE6	Reserved
0x0F87	Reserved	0x0FA7	Reserved	0x0FC7	VDCR2	0x0FE7	Reserved
0x0F88	T02REG	0x0FA8	TA1DRAL	0x0FC8	RTCCR	0x0FE8	Reserved
0x0F89	T03REG	0x0FA9	TA1DRAH	0x0FC9	Reserved	0x0FE9	Reserved
0x0F8A	T02PWM	0x0FAA	TA1DRBL	0x0FCA	Reserved	0x0FEA	Reserved
0x0F8B	T03PWM	0x0FAB	TA1DRBH	0x0FCB	SERSEL	0x0FEB	Reserved
0x0F8C	T02MOD	0x0FAC	TA1MOD	0x0FCC	IRSTSR	0x0FEC	Reserved
0x0F8D	T03MOD	0x0FAD	TA1CR	0x0FCD	WUCCR	0x0FED	Reserved
0x0F8E	T023CR	0x0FAE	TA1SR	0x0FCE	WUCDR	0x0FEE	Reserved
0x0F8F	Reserved	0x0FAF	Reserved	0x0FCF	CGCR	0x0FEF	Reserved
0x0F90	Reserved	0x0FB0	Reserved	0x0FD0	Reserved	0x0FF0	ILPRS1
0x0F91	Reserved	0x0FB1	Reserved	0x0FD1	Reserved	0x0FF1	ILPRS2
0x0F92	Reserved	0x0FB2	Reserved	0x0FD2	Reserved	0x0FF2	ILPRS3
0x0F93	Reserved	0x0FB3	Reserved	0x0FD3	Reserved	0x0FF3	ILPRS4
0x0F94	Reserved	0x0FB4	Reserved	0x0FD4	WDCTR	0x0FF4	ILPRS5
0x0F95	Reserved	0x0FB5	Reserved	0x0FD5	WDCDR	0x0FF5	ILPRS6
0x0F96	Reserved	0x0FB6	Reserved	0x0FD6	WDCNT	0x0FF6	Reserved
0x0F97	Reserved	0x0FB7	Reserved	0x0FD7	WDST	0x0FF7	Reserved
0x0F98	Reserved	0x0FB8	Reserved	0x0FD8	EINTCR1	0x0FF8	Reserved
0x0F99	Reserved	0x0FB9	Reserved	0x0FD9	EINTCR2	0x0FF9	Reserved
0x0F9A	Reserved	0x0FBA	Reserved	0x0FDA	EINTCR3	0x0FFA	Reserved
0x0F9B	Reserved	0x0FBB	Reserved	0x0FDB	EINTCR4	0x0FFB	Reserved
0x0F9C	Reserved	0x0FBC	Reserved	0x0FDC	SYSCR1	0x0FFC	Reserved
0x0F9D	Reserved	0x0FBD	Reserved	0x0FDD	SYSCR2	0x0FFD	Reserved
0x0F9E	Reserved	0x0FBE	Reserved	0x0FDE	SYSCR3	0x0FFE	Reserved
0x0F9F	Reserved	0x0FBF	Reserved	0x0FDF	SYSCR4/SYSSR4	0x0FFF	Reserved

Note 1: Do not access reserved addresses by the program.

9.3 SFR3 (0x0E40 to 0x0EFF)

Table 9-4 SFR3 (0x0E40 to 0x0EBF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0E40	Reserved	0x0E60	Reserved	0x0E80	Reserved	0x0EA0	Reserved
0x0E41	Reserved	0x0E61	Reserved	0x0E81	Reserved	0x0EA1	Reserved
0x0E42	Reserved	0x0E62	Reserved	0x0E82	Reserved	0x0EA2	Reserved
0x0E43	Reserved	0x0E63	Reserved	0x0E83	Reserved	0x0EA3	Reserved
0x0E44	Reserved	0x0E64	Reserved	0x0E84	Reserved	0x0EA4	Reserved
0x0E45	Reserved	0x0E65	Reserved	0x0E85	Reserved	0x0EA5	Reserved
0x0E46	Reserved	0x0E66	Reserved	0x0E86	Reserved	0x0EA6	Reserved
0x0E47	Reserved	0x0E67	Reserved	0x0E87	Reserved	0x0EA7	Reserved
0x0E48	Reserved	0x0E68	Reserved	0x0E88	Reserved	0x0EA8	Reserved
0x0E49	Reserved	0x0E69	Reserved	0x0E89	Reserved	0x0EA9	Reserved
0x0E4A	Reserved	0x0E6A	Reserved	0x0E8A	Reserved	0x0EAA	Reserved
0x0E4B	Reserved	0x0E6B	Reserved	0x0E8B	Reserved	0x0EAB	Reserved
0x0E4C	Reserved	0x0E6C	Reserved	0x0E8C	Reserved	0x0EAC	Reserved
0x0E4D	Reserved	0x0E6D	Reserved	0x0E8D	Reserved	0x0EAD	Reserved
0x0E4E	Reserved	0x0E6E	Reserved	0x0E8E	Reserved	0x0EAE	Reserved
0x0E4F	Reserved	0x0E6F	Reserved	0x0E8F	Reserved	0x0EAF	Reserved
0x0E50	Reserved	0x0E70	Reserved	0x0E90	Reserved	0x0EB0	Reserved
0x0E51	Reserved	0x0E71	Reserved	0x0E91	Reserved	0x0EB1	Reserved
0x0E52	Reserved	0x0E72	Reserved	0x0E92	Reserved	0x0EB2	Reserved
0x0E53	Reserved	0x0E73	Reserved	0x0E93	Reserved	0x0EB3	Reserved
0x0E54	Reserved	0x0E74	Reserved	0x0E94	Reserved	0x0EB4	Reserved
0x0E55	Reserved	0x0E75	Reserved	0x0E95	Reserved	0x0EB5	Reserved
0x0E56	Reserved	0x0E76	Reserved	0x0E96	Reserved	0x0EB6	Reserved
0x0E57	UATCNG	0x0E77	Reserved	0x0E97	Reserved	0x0EB7	Reserved
0x0E58	Reserved	0x0E78	Reserved	0x0E98	Reserved	0x0EB8	Reserved
0x0E59	Reserved	0x0E79	Reserved	0x0E99	Reserved	0x0EB9	Reserved
0x0E5A	Reserved	0x0E7A	Reserved	0x0E9A	Reserved	0x0EBA	Reserved
0x0E5B	Reserved	0x0E7B	Reserved	0x0E9B	Reserved	0x0EBB	Reserved
0x0E5C	Reserved	0x0E7C	Reserved	0x0E9C	Reserved	0x0EBC	Reserved
0x0E5D	Reserved	0x0E7D	Reserved	0x0E9D	Reserved	0x0EBD	Reserved
0x0E5E	Reserved	0x0E7E	Reserved	0x0E9E	Reserved	0x0EBE	Reserved
0x0E5F	Reserved	0x0E7F	Reserved	0x0E9F	Reserved	0x0EBF	Reserved

Note 1: Do not access reserved addresses by the program.

Table 9-5 SFR3 (0x0EC0 to 0x0EFF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0EC0	Reserved	0x0ED0	Reserved	0x0EE0	Reserved	0x0EF0	Reserved
0x0EC1	Reserved	0x0ED1	Reserved	0x0EE1	Reserved	0x0EF1	Reserved
0x0EC2	Reserved	0x0ED2	Reserved	0x0EE2	Reserved	0x0EF2	Reserved
0x0EC3	Reserved	0x0ED3	Reserved	0x0EE3	Reserved	0x0EF3	Reserved
0x0EC4	Reserved	0x0ED4	Reserved	0x0EE4	Reserved	0x0EF4	Reserved
0x0EC5	Reserved	0x0ED5	Reserved	0x0EE5	Reserved	0x0EF5	Reserved
0x0EC6	Reserved	0x0ED6	Reserved	0x0EE6	Reserved	0x0EF6	Reserved
0x0EC7	Reserved	0x0ED7	Reserved	0x0EE7	Reserved	0x0EF7	Reserved
0x0EC8	Reserved	0x0ED8	Reserved	0x0EE8	Reserved	0x0EF8	Reserved
0x0EC9	Reserved	0x0ED9	Reserved	0x0EE9	Reserved	0x0EF9	Reserved
0x0ECA	Reserved	0x0EDA	Reserved	0x0EEA	Reserved	0x0EFA	Reserved
0x0ECB	Reserved	0x0EDB	Reserved	0x0EEB	Reserved	0x0EFB	Reserved
0x0ECC	Reserved	0x0EDC	Reserved	0x0EEC	Reserved	0x0EFC	Reserved
0x0ECD	Reserved	0x0EDD	Reserved	0x0EED	Reserved	0x0EFD	Reserved
0x0ECE	Reserved	0x0EDE	Reserved	0x0EEE	Reserved	0x0EFE	Reserved
0x0ECF	Reserved	0x0EDF	Reserved	0x0EEF	Reserved	0x0EFF	Reserved

Note 1: Do not access reserved addresses by the program.

10. Low Power Consumption Function for Peripherals

The TMP89CM42 has low power consumption registers (POFFCRn) that save power when specific peripheral functions are unused. Each bit of the low power consumption registers can be set to enable or disable each peripheral function. (n = 0, 1, 2, 3)

The basic clock supply to each peripheral function is disabled for power saving, by setting the corresponding bit of the low power consumption registers (POFFCRn) to "0". (The disabled peripheral functions become unavailable.) The basic clock supply to each peripheral function is enabled and the function becomes available by setting the corresponding bit of the low power consumption registers (POFFCRn) to "1".

After reset, the low power consumption registers (POFFCRn) are initialized to "0", and thus the peripheral functions are unavailable. When each peripheral function is used for the first time, be sure to set the corresponding bit of the low power consumption registers (POFFCRn) to "1" in the initial settings of the program (before operating the control register for the peripheral function).

When a peripheral function is operating, the corresponding bit of the low power consumption registers (POFFCRn) must not be changed to "0". If it is changed, the peripheral function may operate unexpectedly.

10.1 Control

The low power consumption function is controlled by the low power consumption registers (POFFCRn). (n = 0, 1, 2, 3)

Low power consumption register 0

POFFCR0 (0x0F74)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	TC023EN	TC001EN	-	-	TCA1EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 control	0 1	Disable Enable
TC001EN	TC00,01 control	0 1	Disable Enable
TCA1EN	TCA1 control	0 1	Disable Enable
TCA0EN	TCA0 control	0 1	Disable Enable

Low power consumption register 1

POFFCR1 (0x0F75)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	SBI0EN	-	-	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0 1	Disable Enable
UART1EN	UART1 control	0 1	Disable Enable
UART0EN	UART0 control	0 1	Disable Enable

Low power consumption register 2

POFFCR2 (0x0F76)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0 1	Disable Enable
SIO0EN	SIO0 control	0 1	Disable Enable

Low power consumption register 3

POFFCR3		7	6	5	4	3	2	1	0
(0x0F77)	Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 control	0 1	Disable Enable
INT4EN	INT4 control	0 1	Disable Enable
INT3EN	INT3 control	0 1	Disable Enable
INT2EN	INT2 control	0 1	Disable Enable
INT1EN	INT1 control	0 1	Disable Enable
INT0EN	INT0 control	0 1	Disable Enable

11. Divider Output ($\overline{\text{DVO}}$)

This function outputs approximately 50% duty pulses that can be used to drive the piezoelectric buzzer or other device.

11.1 Configuration

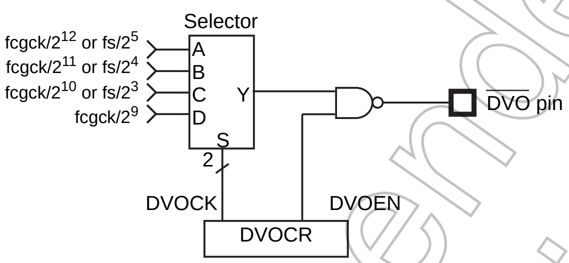


Figure 11-1 Divider Output

11.2 Control

The divider output is controlled by the divider output control register (DVOCR).

Divider output control register

DVOCR (0x0038)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	DV0EN	DVOCK	
Read/Write	R	R	R	R	R	R/W	R/W	
After reset	0	0	0	0	0	0	0	0

DVOEN	Enables/disables the divider output	0: Disable the divider output 1: Enable the divider output			
DVOCK	Selects the divider output frequency Unit: [Hz]		NORMAL 1/2, IDLE 1/2 mode		SLOW1/2 SLEEP1 mode
			DV9CK=0	DV9CK=1	
		00	fcgck/2^{12}	fs/2^5	fs/2^5
		01	fcgck/2^{11}	fs/2^4	fs/2^4
		10	fcgck/2^{10}	fs/2^3	fs/2^3
		11	fcgck/2^9	Reserved	Reserved

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: DVOCR<DVOEN> is cleared to "0" when the operation is switched to STOP or IDLE0/SLEEP0 mode. DVOCR<DVOCK> holds the value.

Note 3: When SYSCR1<DV9CK> is "1" in the NORMAL 1/2 or IDLE 1/2 mode, the DVO frequency is subject to some fluctuations to synchronize fs and fcgck.

Note 4: Bits 7 to 3 of DVOCR are read as "0".

11.3 Function

Select the divider output frequency at DVOCR<DVOCK>.

The divider output is enabled by setting DVOCR<DVOEN> to "1". Then, The rectangular waves selected by DVOCR<DVOCK> is output from $\overline{\text{DVO}}$ pin.

It is disabled by clearing DVOVR<DVOEN> to "0". And $\overline{\text{DVO}}$ pin keeps "H" level.

When the operation is changed to STOP or IDLE0/SLEEP0 mode, DVOCR<DVOEN> is cleared to "0" and the $\overline{\text{DVO}}$ pin outputs the "H" level.

The divider output source clock operates, regardless of the value of DVOCR<DVOEN>.

Therefore, the frequency of the first divider output after DVOCR<DVOEN> is set to "1" is not the frequency set at DVOCR<DVOCK>.

When the operation is changed to the software, STOP or IDLE0/SLEEP0 mode is activated and DVOCR<DVOEN> is cleared to "0", the frequency of the divider output is not the frequency set at DVOCR<DVOCK>.

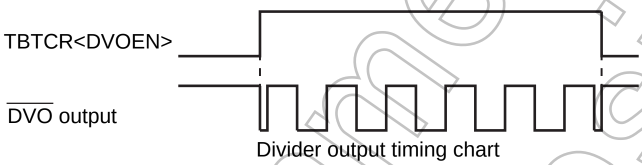


Figure 11-2 Divider Output Timing

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the divider output frequency does not reach the expected value due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs).

Example:2.441 kHz pulse output (fcgck = 10.0 MHz)

```
LD      (DVOCR), 0y00000100      ;DVOCK ← "00", DVOEN ← "1"
```

Table 11-1 Divider Output Frequency (Example: fcgck = 10.0 MHz, fs = 32.768 kHz)

DVOCK	Divider output frequency [Hz]		
	NORMAL 1/2, IDLE 1/2 mode		SLOW1/2, SLEEP1 mode
	DV9CK = 0	DV9CK = 1	
00	2.441 k	1.024 k	1.024 k
01	4.883 k	2.048 k	2.048 k
10	9.766 k	4.096 k	4.096 k
11	19.531 k	Reserved	Reserved

11.4 Revision History

Rev	Description
RA001	Deleted SLEEP2 description.

12. Time Base Timer (TBT)

The time base timer generates the time base for key scanning, dynamic display and other processes. It also provides a time base timer interrupt (INTTBT) in a certain cycle.

12.1 Time Base Timer

12.1.1 Configuration

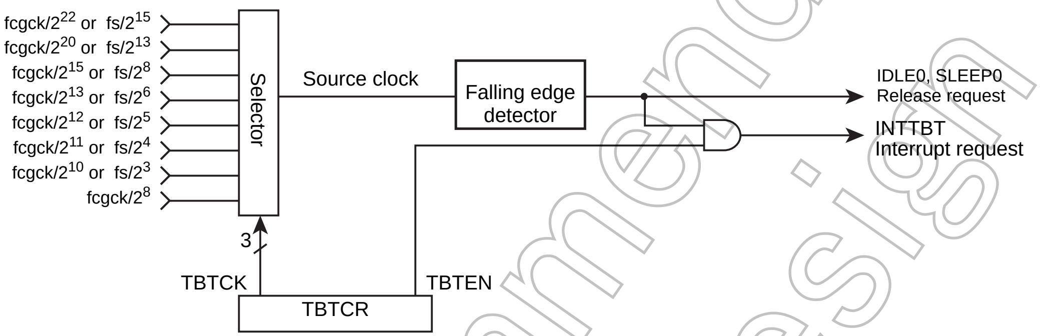


Figure 12-1 Time Base Timer Configuration

12.1.2 Control

The time base timer is controlled by the time base timer control register (TBTCR).

Time base timer control register

TBTCR (0x0039)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	TBTEN	TBTCK		
Read/Write	R	R	R	R	R/W	R/W		
After reset	0	0	0	0	0	0	0	0

TBTEN	Enables/disables the time base timer interrupt requests	0: Disables generation of interrupt request signals 1: Enables generation of interrupt request signals			
TBTCK	Selects the time base timer interrupt frequency Unit: [Hz]	TBTCK	NORMAL 1/2, IDLE 1/2 mode		SLOW1/2, SLEEP1 mode
			DV9CK = 0	DV9CK = 1	
		000	$fcgck/2^{22}$	$fs/2^{15}$	$fs/2^{15}$
		001	$fcgck/2^{20}$	$fs/2^{13}$	$fs/2^{13}$
		010	$fcgck/2^{15}$	$fs/2^{8}$	Reserved
		011	$fcgck/2^{13}$	$fs/2^{6}$	Reserved
		100	$fcgck/2^{12}$	$fs/2^{5}$	Reserved
		101	$fcgck/2^{11}$	$fs/2^{4}$	Reserved
		110	$fcgck/2^{10}$	$fs/2^{3}$	Reserved
		111	$fcgck/2^{8}$	Reserved	Reserved

Note 1: fcgck : Gear clock [Hz], fs : Low-frequency clock [Hz]

Note 2: When the operation is changed to the STOP mode, TBTCR<TBTEN> is cleared to "0" and TBTCR<TBTCK> maintains the value.

Note 3: TBTCR<TBTCK> should be set when TBTCR<TBTEN> is "0".

Note 4: When SYSCR1<DV9CK> is "1" in the NORMAL 1/2 or IDLE1/2 mode, the interrupt request is subject to some fluctuations to synchronize fs and fcgck.

Note 5: Bits 7 to 4 of TBTCR are read as "0".

12.1.3 Functions

Select the source clock frequency for the time base timer by TBTCR<TBTCK>. TBTCR<TBTCK> should be changed when TBTCR<TBTEN> is "0". Otherwise, the INTTBT interrupt request is generated at unexpected timing.

Setting TBTCR<TBTEN> to "1" causes interrupt request signals to occur at the falling edge of the source clock. When TBTCR<TBTEN> is cleared to "0", no interrupt request signal will occur.

When the operation is changed to the STOP mode, TBTCR<TBTEN> is cleared to "0".

The source clock of the time base timer operates regardless of the TBTCR<TBTEN> value.

A time base timer interrupt is generated at the first falling edge of the source clock after a time base timer interrupt request is enabled. Therefore, the period from when the time TBTCR<TBTEN> is set to "1" to the time when the first interrupt request occurs is shorter than the frequency period set at TBTCR<TBTCK>.

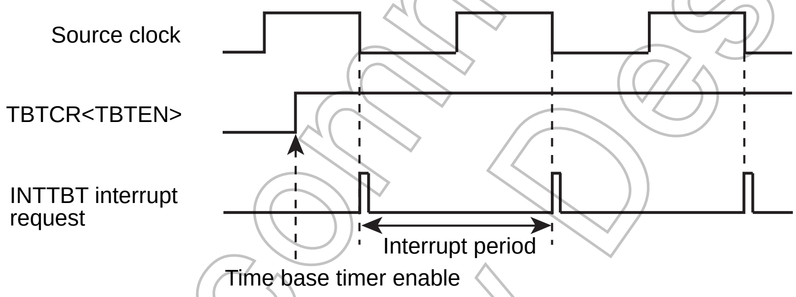


Figure 12-2 Time Base Timer Interrupt

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, The interrupt request will not occur at the expected timing due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs). It is recommened that the operation mode is changed when TBTCR<TBTEN> is "0".

Table 12-1 Time Base Timer Interrupt Frequency (Example: when fcgck = 10.0 MHz and fs = 32.768 kHz)

TBTCK	Time base timer interrupt frequency [Hz]		
	NORMAL1/2, IDLE1/2 mode	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1 mode
	DV9CK = 0	DV9CK = 1	
000	2.38	1	1
001	9.54	4	4
010	305.18	128	Reserved
011	1220.70	512	Reserved
100	2441.41	1024	Reserved
101	4882.81	2048	Reserved
110	9765.63	4096	Reserved
111	39062.5	Reserved	Reserved

Example:Set the time base timer interrupt frequency to fcgck/2^{15} [Hz] and enable interrupts.

```
DI                                   ;IMF ← 0
SET     (EIRL). 5                    ;Set the interrupt enable register
EI                                   ;IMF ← 1
LD      (TBTCR), 0y00000010          ;Set the interrupt frequency
LD      (TBTCR), 0y00001010          ;Enable generation of interrupt request signals
```

12.2 Revision History

Rev	Description
RA001	Deleted SLEEP2 description

13. 16-bit Timer Counter (TCA)

The TMP89CM42 contains 2 channels of high-performance 16-bit timer counters (TCA).

This chapter describes the 16-bit timer counter A0. For the 16-bit timer counter A1, replace the SFR addresses and pin names, as shown in Table 13-1 and Table 13-2.

Table 13-1 SFR Address Assignment

	TAxDRAL (Address)	TAxDRAH (Address)	TAxDRBL (Address)	TAxDRBH (Address)	TAxMOD (Address)	TAxCR (Address)	TAxSR (Address)	Low power consump-tion register
Timer counter A0	TA0DRAL (0x002D)	TA0DRAH (0x002E)	TA0DRBL (0x002F)	TA0DRBH (0x0030)	TA0MOD (0x0031)	TA0CR (0x0032)	TA0SR (0x0033)	POFFCR0 <TCA0EN>
Timer counter A1	TA1DRAL (0x0FA8)	TA1DRAH (0x0FA9)	TA1DRBL (0x0FAA)	TA1DRBH (0x0FAB)	TA1MOD (0x0FAC)	TA1CR (0x0FAD)	TA1SR (0x0FAE)	POFFCR0 <TCA1EN>

Table 13-2 Pin Names

	Timer input pin	PPG output pin
Timer counter A0	TCA0 pin	$\overline{\text{PPGA0}}$ pin
Timer counter A1	TCA1 pin	$\overline{\text{PPGA1}}$ pin

13.1 Configuration

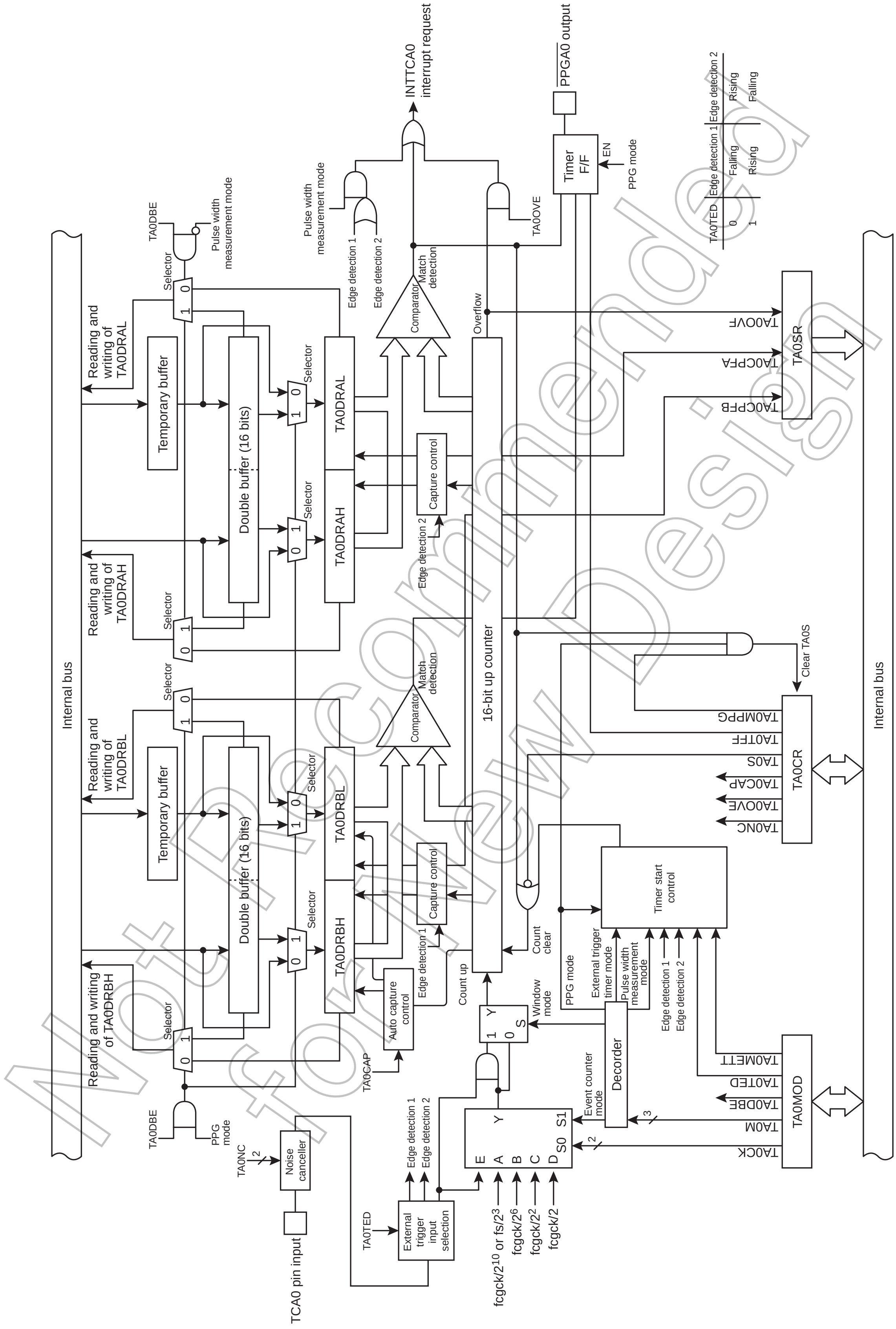


Figure 13-1 Timer Counter A0

13.2 Control

Timer Counter A0 is controlled by the low power consumption register (POFFCR0), the timer counter A0 mode register (TA0MOD), the timer counter A0 control register (TA0CR) and two 16-bit timer A0 registers (TA0DRA and TA0DRB).

Low power consumption register 0

POFFCR0 (0x0F74)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	TC023EN	TC001EN	-	-	TCA1EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 control	0 1	Disable Enable
TC001EN	TC00,01 control	0 1	Disable Enable
TCA1EN	TCA1 control	0 1	Disable Enable
TCA0EN	TCA0 control	0 1	Disable Enable

Timer counter A0 mode register

TA0MOD		7	6	5	4	3	2	1	0
(0x0031)	Bit Symbol	TA0DBE	TA0TED	TA0MCAP TA0METT	TA0CK		TA0M		
	Read/Write	R/W	R/W	R/W	R/W		R/W		
	After reset	1	0	0	0	0	0	0	0

TA0DBE	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TA0TED	External trigger input selection	0	Rising edge/H level		
		1	Falling edge/L level		
TA0MCAP	Pulse width measurement mode control	0	Double edge capture		
		1	Single edge capture		
TA0METT	External trigger timer mode control	0	Trigger start		
		1	Trigger start & stop		
TA0CK	Timer counter 1 source clock selection		NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> ="0"	SYSCR1<DV9CK> ="1"	
		00	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$
		01	$fcgck/2^6$	$fcgck/2^6$	-
		10	$fcgck/2^2$	$fcgck/2^2$	-
		11	fcgck/2	fcgck/2	-
TA0M	Timer counter 1 operation mode selection	000	Timer mode		
		001	Timer mode		
		010	Event counter mode		
		011	PPG output mode (Software start)		
		100	External trigger timer mode		
		101	Window mode		
		110	Pulse width measurement mode		
		111	Reserved		

Note 1: fcgck, Gear clock [Hz]; fs, Low-frequency clock [Hz]

Note 2: Set TA0MOD in the stopped state (TA0CR<TA0S>="0"). Writing to TA0MOD is invalid during the operation (TA0CR<TA0S>="1").

Timer counter A0 control register

TA0CR		7	6	5	4	3	2	1	0
(0x0032)	Bit Symbol	TA0OVE	TA0TFF	TA0NC		-	-	TA0ACAP TA0MPPG	TA0S
	Read/Write	R/W	R/W	R/W		R	R	R/W	R/W
	After reset	0	1	0	0	0	0	0	0

TA0OVE	Overflow interrupt control	0	Generate no INTTCA0 interrupt request when the counter overflow occurs.	
		1	Generate an INTTCA0 interrupt request when the counter overflow occurs.	
TA0TFF	Timer F/F control	0	Clear	
		1	Set	
TA0NC	Noise canceller sampling interval setting		NORMAL 1/2 or IDLE 1/2 mode	SLOW1/2 or SLEEP1 mode
		00	No noise canceller	No noise canceller
		01	fcgck/2	-
		10	$fcgck/2^2$	-
		11	$fcgck/2^8$	fs/2
TA0ACAP	Auto capture function	0	Disable the auto capture	
		1	Enable the auto capture	
TA0MPPG	PPG output control	0	Continuous	
		1	One-shot	
TA0S	Timer counter A start control	0	Stop & counter clear	
		1	Start	

Note 1: The auto capture can be used only in the timer, event counter, external trigger timer and window modes.

Note 2: Set TA0TFF, TA0OVE and TA0NC in the stopped state (TA0S="0"). Writing is invalid during the operation (TA0S="1").

Note 3: When the STOP mode is started, the start control (TA0S) is automatically cleared to "0" and the timer stops. Set TA0S again to use the timer counter after the release of the STOP mode.

Note 4: When a read instruction is executed on TA0CR, bits 3 and 2 are read as "0".

Note 5: Do not set TA0NC to "01" or "10" when the SLOW 1/2 or SLEEP 1 mode is used. Setting TA0NC to "01" or "10" stops the noise canceller and no signal is input to the timer.

Timer counter A0 status register

TA0SR (0x0033)		7	6	5	4	3	2	1	0
	Bit Symbol	TA0OVF	-	-	-	-	-	TA0CPFA	TA0CPFB
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

TA0OVF	Overflow flag	0	No overflow has occurred.
		1	At least an overflow has occurred.
TA0CPFA	Capture completion flag A	0	No capture operation has been executed.
		1	At least a pulse width capture has been executed in the double-edge capture.
TA0CPFB	Capture completion flag B	0	No capture operation has been executed.
		1	At least a capture operation has been executed in the single-edge capture. At least a pulse duty width capture has been executed in the double-edge capture.

Note 1: TA0OVF, TA0CPFA and TA0CPFB are cleared to "0" automatically after TA0SR is read. Writing to TA0SR is invalid.

Note 2: When a read instruction is executed on TA0SR, bits 6 to 2 are read as "0".

Timer counter A0 register AH

TA0DRAH (0x002E)		15	14	13	12	11	10	9	8
	Bit Symbol	TA0DRAH							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register AL

TA0DRAL (0x002D)		7	6	5	4	3	2	1	0
	Bit Symbol	TA0DRAL							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BH

TA0DRBH (0x0030)		15	14	13	12	11	10	9	8
	Bit Symbol	TA0DRBH							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BL

TA0DRBL (0x002F)		7	6	5	4	3	2	1	0
	Bit Symbol	TA0DRBL							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note 1: When a write instruction is executed on TA0DRAL (TA0DRBL), the set value does not become effective immediately, but is temporarily stored in the temporary buffer. Subsequently, when a write instruction is executed on the higher-level register, TA0DRAH (TA0DRBH), the 16-bit set values are collectively stored in the double buffer or TA0DRAL/H. When setting data to the timer counter A0 register, be sure to write the data into the lower level register and the higher level in this order.

Note 2: The timer counter A0 register is not writable in the pulse width measurement mode.

13.3 Low Power Consumption Function

Timer counter A0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCA0EN> to "0" disables the basic clock supply to timer counter A0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCA0EN> to "1" enables the basic clock supply to timer counter A0 and allows the timer to operate.

After reset, POFFCR0<TCA0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCA0EN> to "1" in the initial setting of the program (before the timer control register is operated).

Do not change POFFCR0<TCA0EN> to "0" during the timer operation. Otherwise timer counter A0 may operate unexpectedly.

13.4 Timer Function

Timer counter A0 has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

13.4.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times.

13.4.1.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "000" or "001" activates the timer mode. Select the source clock at TA0MOD<TA0CK>.

Setting TA0CR<TA0S> to "1" starts the timer operation. After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> becomes invalid. Be sure to complete the required mode settings before starting the timer.

Table 13-3 Timer Mode Resolution and Maximum Time Setting

TA0MOD <TA0CK>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
00	fcgck/2^{10}	fs/2^3	fs/2^3	102.4μs	244.1μs	6.7s	16s
01	fcgck/2^6	fcgck/2^6	-	6.4μs	-	419.4ms	-
10	fcgck/2^2	fcgck/2^2	-	400ns	-	26.2ms	-
11	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-

13.4.1.2 Operation

Setting TA0CR<TA0S> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up-counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting. Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

13.4.1.3 Auto capture

The latest contents of the up counter can be taken into timer register B (TA0DRB) by setting TA0CR<TA0ACAP> to "1" (auto capture function). When TA0CR<TA0ACAP> is "1", the current contents of the up counter can be read by reading TA0DRBL. TA0DRBH is loaded at the same time as TA0DRBL is read. Therefore, when reading the captured value, be sure to read TA0DRBL and TA0DRBH in this order. (The capture time is the timing when TA0DRBL is read.) The auto capture function can be used whether the timer is operating or stopped. When the timer is stopped, TA0DRBL is read as "0x00". TA0DRBH keeps the captured value after the timer stops, but it is cleared to "0x00" when TA0DRBL is read while the timer is stopped.

If the timer is started with TA0CR<TA0ACAP> written to "1", the auto capture is enabled immediately after the timer is started.

Note 1: The value set to TA0CR<TA0ACAP> cannot be changed at the same time as TA0CR<TA0S> is rewritten from "1" to "0". (This setting is invalid.)

13.4.1.4 Register buffer configuration

(1) Temporary buffer

The TMP89CM42 contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL, the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH, the set value is stored into the double buffer or TA0DRAH. At the same time, the set value in the temporary buffer is stored into the double buffer or TA0DRAL. (This structure is designed to enable the set values of the lower-level and higher-level registers simultaneously.) Therefore, when setting data to TA0DRA, be sure to write the data into TA0DRAL and TA0DRAH in this order.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89CM42, the double buffer can be used by setting TA0CR<TA0DBF>. Setting TA0CR<TA0DBF> to "0" disables the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TA0DRAH during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L compare the up counter value to the last set values. If the values are matched, an INTTCA0 interrupt request is generated and the double buffer set value is stored in TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TA0DRAH/L, the double buffer value (the last set value) is read, rather than the TA0DRAH/L values (the current effective values).

  When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L.

- When the double buffer is disabled

  When a write instruction is executed on TA0DRAH during the timer operation, the set value is immediately stored into TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

  If the values set to TA0DRAH/L are smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into TA0DRAH/L.

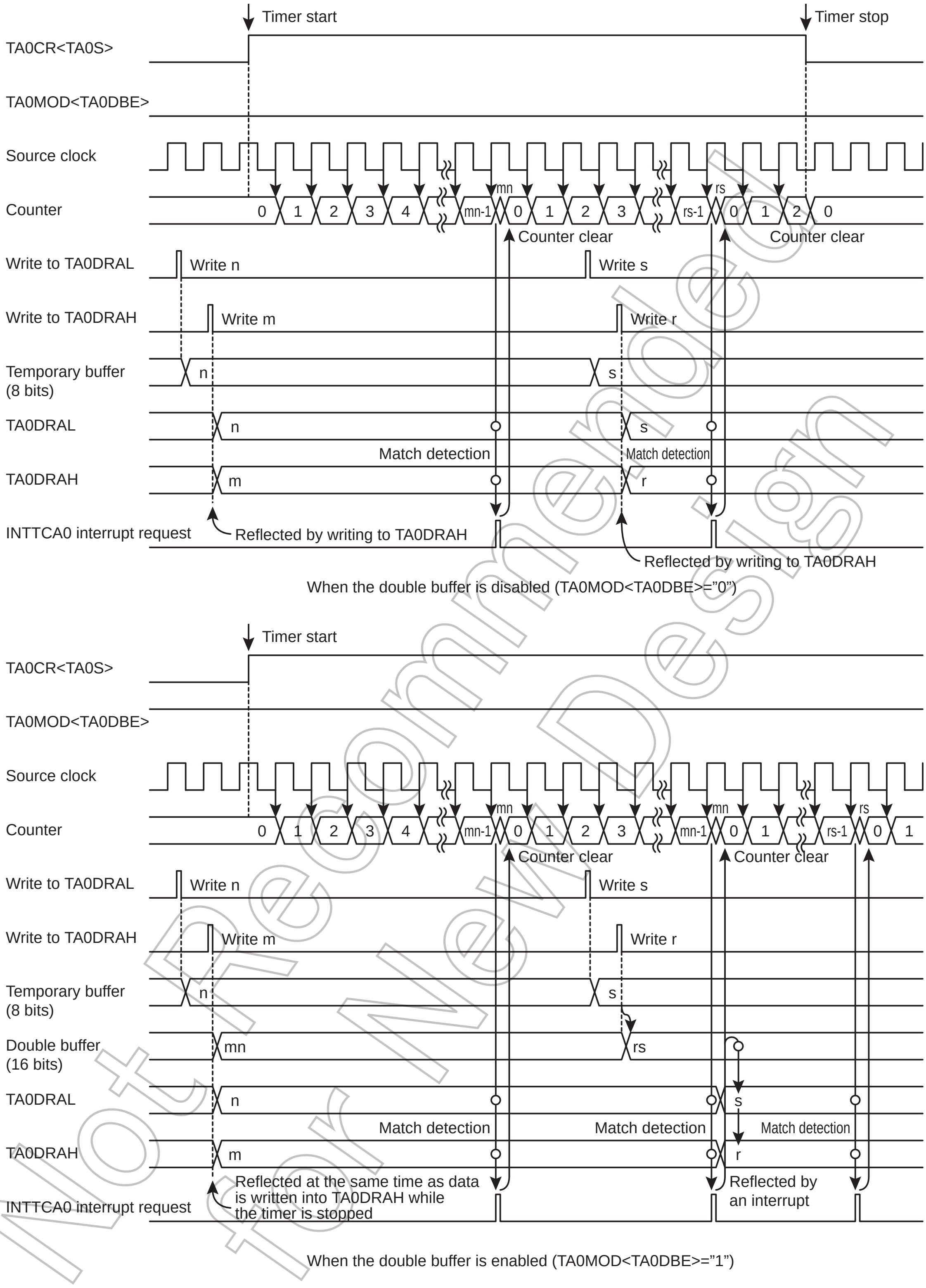


Figure 13-2 Timer Mode Timing Chart

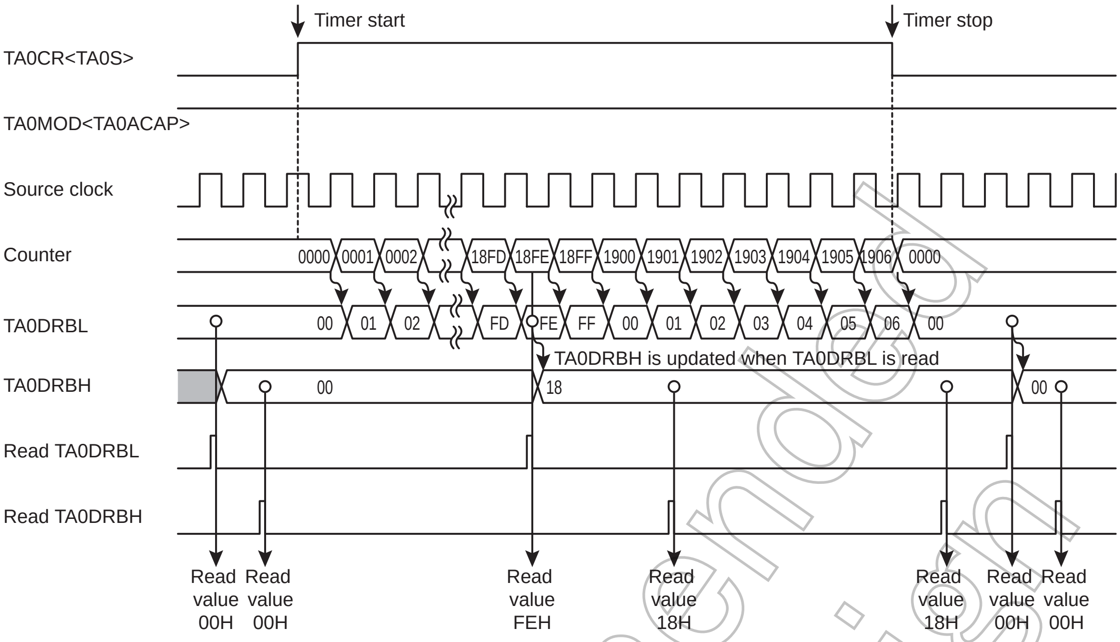


Figure 13-3 Timer Mode Timing Chart (Auto Capture)

13.4.2 External trigger timer mode

In the external trigger timer mode, the up counter starts counting when it is triggered by the input to the TCA0 pin.

13.4.2.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "100" activates the external trigger timer mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.2.2 Operation

After the timer is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments according to the selected source clock. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.

When TA0MOD<TA0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting. In this mode, an interrupt request can be generated by detecting that the input pulse exceeds a certain pulse width. If TA0MOD<TA0METT> is "0", the detection of the selected edge and the opposite edge is ignored during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

13.4.2.3 Auto capture

Refer to "13.4.1.3 Auto capture".

13.4.2.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

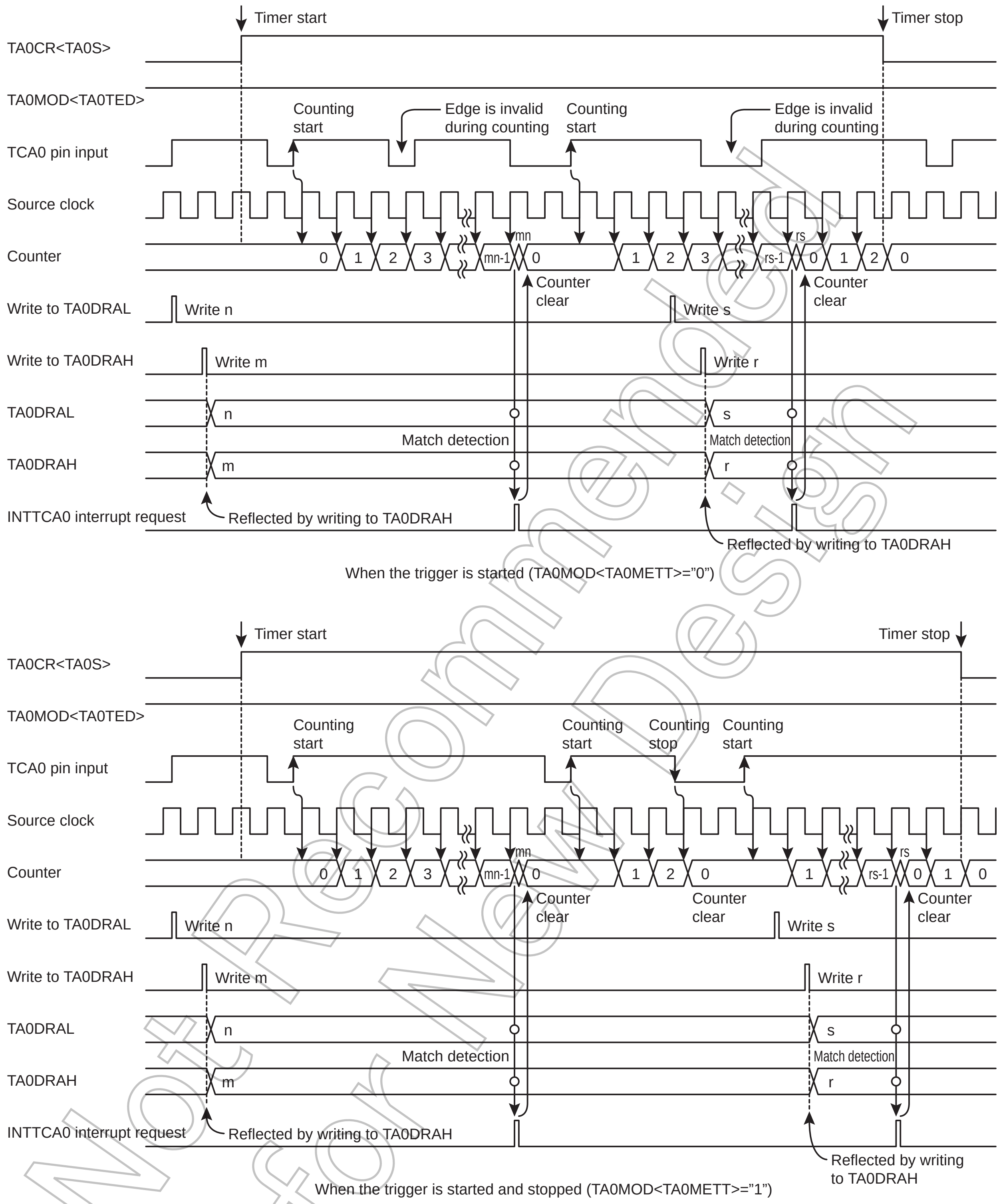


Figure 13-4 External Trigger Timer Timing Chart

13.4.3 Event counter mode

In the event counter mode, the up counter counts up at the edge of the input to the TCA0 pin.

13.4.3.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "010" activates the event counter mode.

Set the trigger edge at the external trigger input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge for counting up.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.3.2 Operation

After the event counter mode is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments.

When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting and counts up at each edge of the input to the TCA0 pin. Setting TA0CR<TA0S> to "0" during the operation causes the up counter to stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in the NORMAL 1/2 or IDLE 1/2 mode) or fs/2 [Hz] (in the SLOW 1/2 or SLEEP 1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

13.4.3.3 Auto capture

Refer to "13.4.1.3 Auto capture".

13.4.3.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

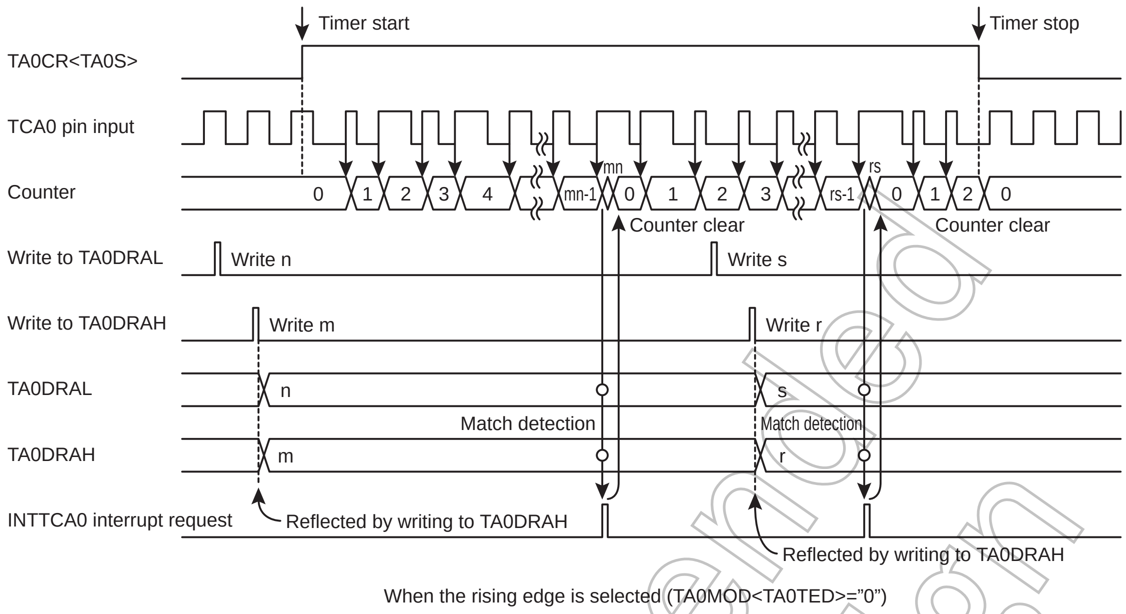


Figure 13-5 Event Count Mode Timing Chart

13.4.4 Window mode

In the window mode, the up counter counts up at the rising edge of the pulse that is logical anded product of the input pulse to the TCA0 pin (window pulse) and the internal clock.

13.4.4.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "101" activates the window mode. Select the source clock at TA0MOD<TA0CK>.

Select the window pulse level at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" enables counting up as long as the window pulse is at the "H" level. Setting TA0MOD<TA0TED> to "1" enables counting up as long as the window pulse is at the "L" level.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.4.2 Operation

After the operation is started, when the level selected at TA0MOD<TA0TED> is input to the TCA0 pin, the up counter increments according to the source clock selected at TA0MOD<TA0CK>. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency to be supplied must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the programmed internal source clock.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

13.4.4.3 Auto capture

Refer to "13.4.1.3 Auto capture".

13.4.4.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

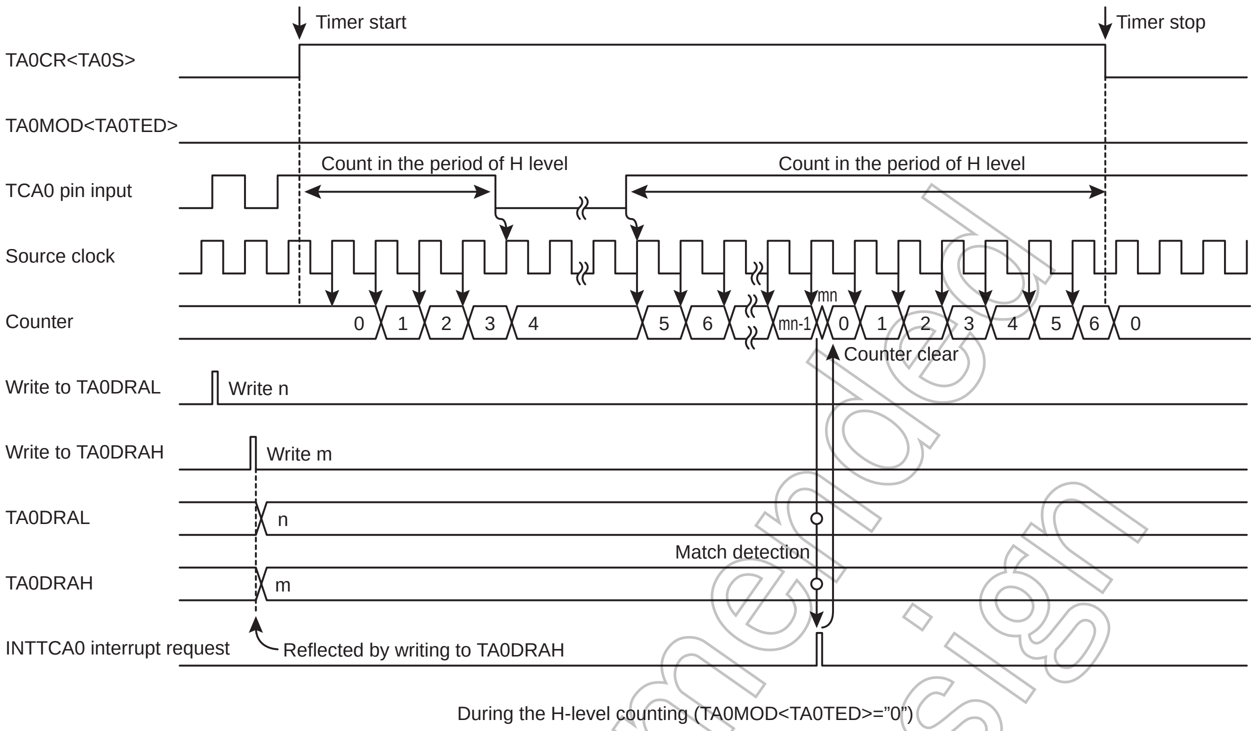


Figure 13-6 Window Mode Timing Chart

13.4.5 Pulse width measurement mode

In the pulse width measurement mode, the up counter starts counting at the rising/falling edge(s) of the input to the TCA0 pin and measures the input pulse width based on the internal clock.

13.4.5.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "110" activates the pulse width measurement mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge as a trigger to start the capture.

The operation after capturing is determined by the pulse width measurement mode control TA0MOD<TA0MCAP>. Setting TA0MOD<TA0MCAP> to "0" selects the double-edge capture. Setting TA0MOD<TA0MCAP> to "1" selects the single-edge capture.

The operation to be executed in case of an overflow of the up counter can be selected at the overflow interrupt control TA0CR<TA0OVE>. Setting TA0OVE to "1" makes an INTTCA0 interrupt request occur in case of an overflow. Setting TA0OVE to "0" makes no INTTCA0 interrupt request occur in case of an overflow.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". In this time, TA0DRA and TA0DRB register are initialized to "0x0000". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.5.2 Operation

After the timer is started, when the selected trigger edge (start edge) is input to the TCA0 pin, INTTCA0 interrupt request is generated, and then the up counter increments according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TA0DRB, an INTTCA0 interrupt request is generated, and TA0SR<TA0CPFB> is set to "1". Depending on the TA0MOD<TA0MCAP> setting, the operation differs as follows:

- Double-edge capture (When TA0MOD<TA0MCAP> is "0")

  The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TA0DRA, an INTTCA0 interrupt request is generated, and TA0SR<TA0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

- Single-edge capture (When TA0MOD<TA0MCAP> is "1")

  The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, INTTCA0 interrupt request is generated, and then the up counter restarts increment.

When the up counter overflows during capturing, the overflow flag TA0SR<TA0OVF> is set to "1". At this time, an INTTCA0 interrupt request occurs if the overflow interrupt control TA0CR<TA0OVE> is set to "1".

The capture completion flags (TA0SR<TA0CPFA, TA0CPFB> and the overflow flag (TA0SR<TA0OVF>) are cleared to "0" automatically when TA0SR is read.

The captured value must be read from TA0DRB (and also from TA0DRA for the double-edge capture) before the next trigger edge is detected. If the captured value is not read, it becomes undefined. TA0DRA and TA0DRB must be read by using a 16-bit access instruction.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

Note 1: After the timer is started, if the edge opposite to the selected trigger edge is detected first, no capture is executed and no INTTCA0 interrupt request occurs. In this case, the capture starts when the selected trigger edge is detected next.

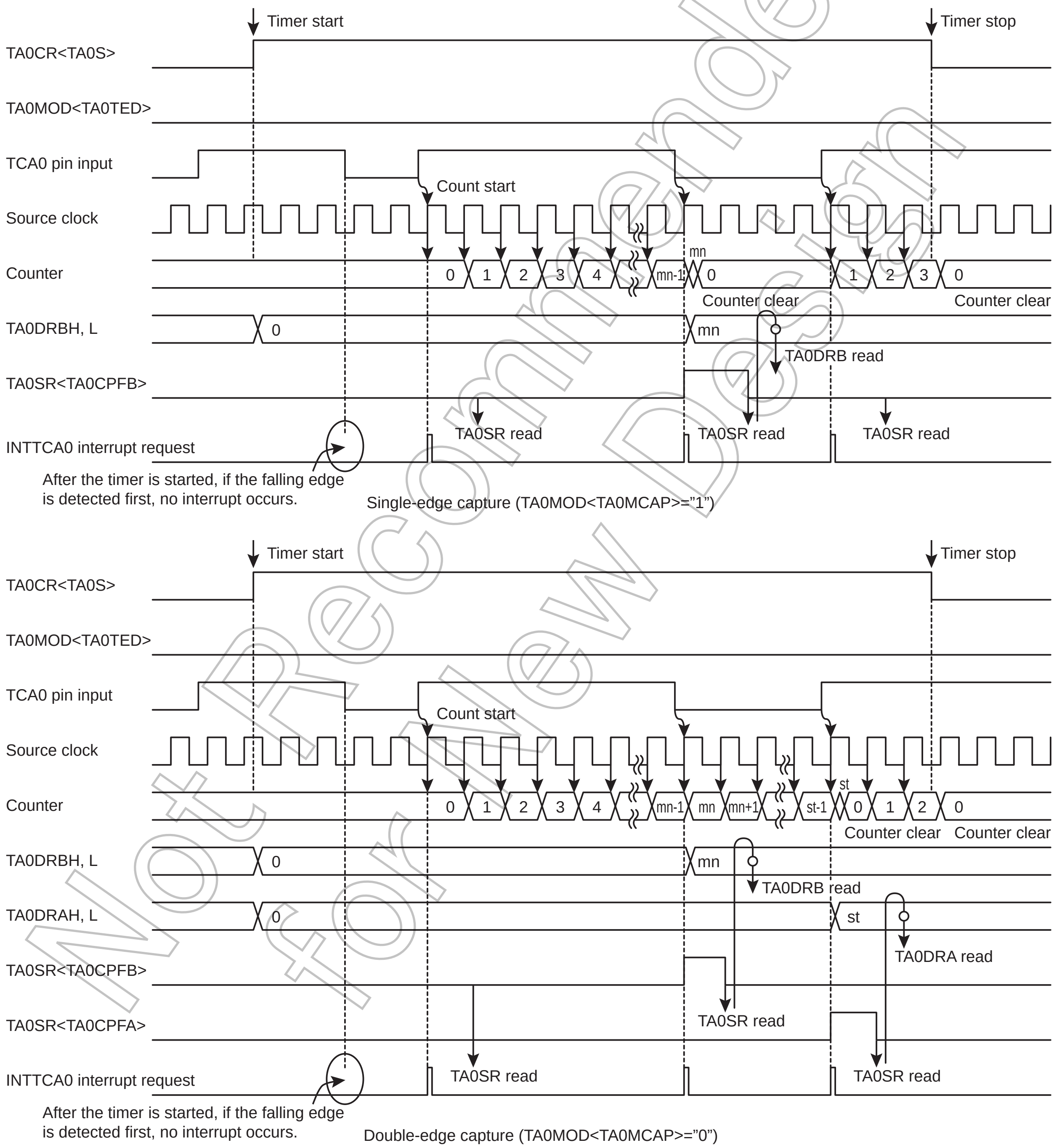


Figure 13-7 Pulse Width Measurement Mode Timing Chart

13.4.5.3 Capture process

Figure 13-8 shows an example of the capture process for INTTCA0 interrupt subroutine. The capture edge or overflow state can be easily judged by status register (TA0SR).

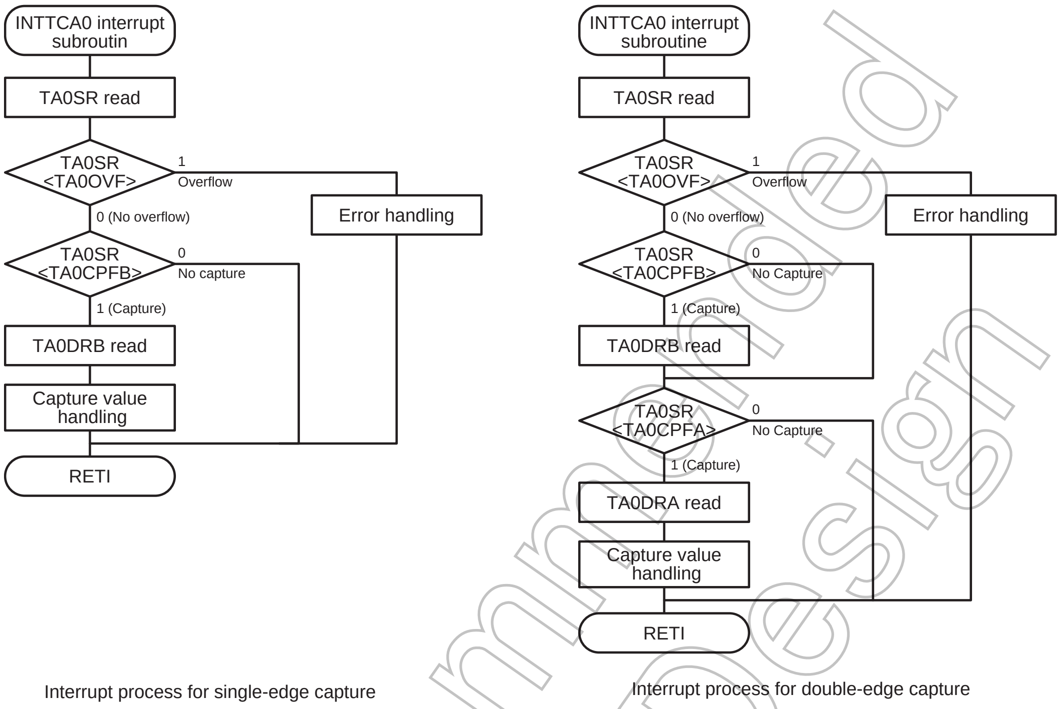


Figure 13-8 Example of capture process

13.4.6 Programmable pulse generate (PPG) mode

In the PPG output mode, an arbitrary duty pulse is output by two timer registers.

13.4.6.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "011" activates the PPG output mode. Select the source clock at TA0MOD<TA0CK>. Select continuous or one-shot PPG output at TA0CR<TA0MPPG>.

Set the PPG output cycle at TA0DRA and set the time until the output is reversed first at TA0DRB. Be sure to set register values so that TA0DRA is larger than TA0DRB.

Note that this mode uses the $\overline{\text{PPGA0}}$ pin. the $\overline{\text{PPGA0}}$ pin must be set to the output mode beforehand in port settings.

Set the initial state of the $\overline{\text{PPGA0}}$ pin at the timer flip-flop TA0CR<TA0TFF>. Setting TA0CR<TA0TFF> to "1" selects the "H" level as the initial state of the $\overline{\text{PPGA0}}$ pin. Setting TA0CR<TA0TFF> to "0" selects the "L" level as the initial state of the $\overline{\text{PPGA0}}$ pin.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE, TA0TFF> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.6.2 Operation

after the timer is started, the up counter increments .

When a match between the up counter value and the value set to timer register B (TA0DRB) is detected, the $\overline{\text{PPGA0}}$ pin is changed to the "H" level if TA0CR<TA0TFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the "L" level if TA0CR<TA0TFF> is "1".

Subsequently, the up counter continues counting. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, the $\overline{\text{PPGA0}}$ pin is changed to the "L" level if TA0CR<TA0TEFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the "H" level if TA0CR<TA0TFF> is "1". At this time, an INTTCA0 interrupt request occurs. If the PPG output control TA0CR<TA0MPPG> is set to "1" (one-shot), TA0CR<TA0S> is automatically cleared to "0" and the timer stops.

If TA0CR<TA0MPPG> is set to "0" (continuous), the up counter is cleared to "0x0000" and continues counting and PPG output. When TA0CR<TA0S> is set to "0" (including the auto stop by the one-shot operation) during the PPG output, the $\overline{\text{PPGA0}}$ pin returns to the level set in TA0CR<TA0TFF>.

TA0CR<TA0MPPG> can be changed during the operation. Changing TA0CR<TA0MPPG> from "1" to "0" during the operation cancels the one-shot operation and enables the continuous operation. Changing TA0CR<TA0MPPG> from "0" to "1" during the operation clears TA0CR<TA0S> to "0" and stops the timer automatically after the current pulse output is completed.

Timer registers A and B can be set to the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer. When the values set to TA0DRA and TA0DRB are changed during the PPG output with the double buffer enabled, the writing to TA0DRA and TA0DRB will not immediately become effective but will become effective when a match between TA0DRA and the up counter is detected. If the double buffer is disabled, the writing to TA0DRA and TA0DRB will become effective immediately. If the written value is smaller than the up counter value, the up counter overflows. After a cycle, the counter match process is executed to reverse the output.

13.4.6.3 Register buffer configuration

(1) Temporary buffer

The TMP89CM42 contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL (TA0DRBL), the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH (TA0DRBH), the set value is stored into the double buffer or TA0DRAH (TA0DRBH). At the same time, the set value in the temporary buffer is stored into the double buffer or TA0DRAL (TA0DRBL). (This structure is designed to enable the set values of the lower-level register and the higher-level register simultaneously.) Therefore, when setting data to TA0DRA (TA0DRB), be sure to write the data into TA0DRAL and TA0DRAH (TA0DRBL and TA0DRBH) in this order.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89CM42, the double buffer can be used by setting TA0CR<TA0DBF>. Setting TA0CR<TA0DBF> to "0" disables the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L (TA0DRBH/L) compare the last set values to the counter value. If a match is detected, an INTTCA0 interrupt request is generated and the double buffer set value is stored into TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TA0DRAH/L (TA0DRBH/L), the double buffer value (the last set value) is read, not the TA0DRAH/L (TA0DRBH/L) values (the current effective values).

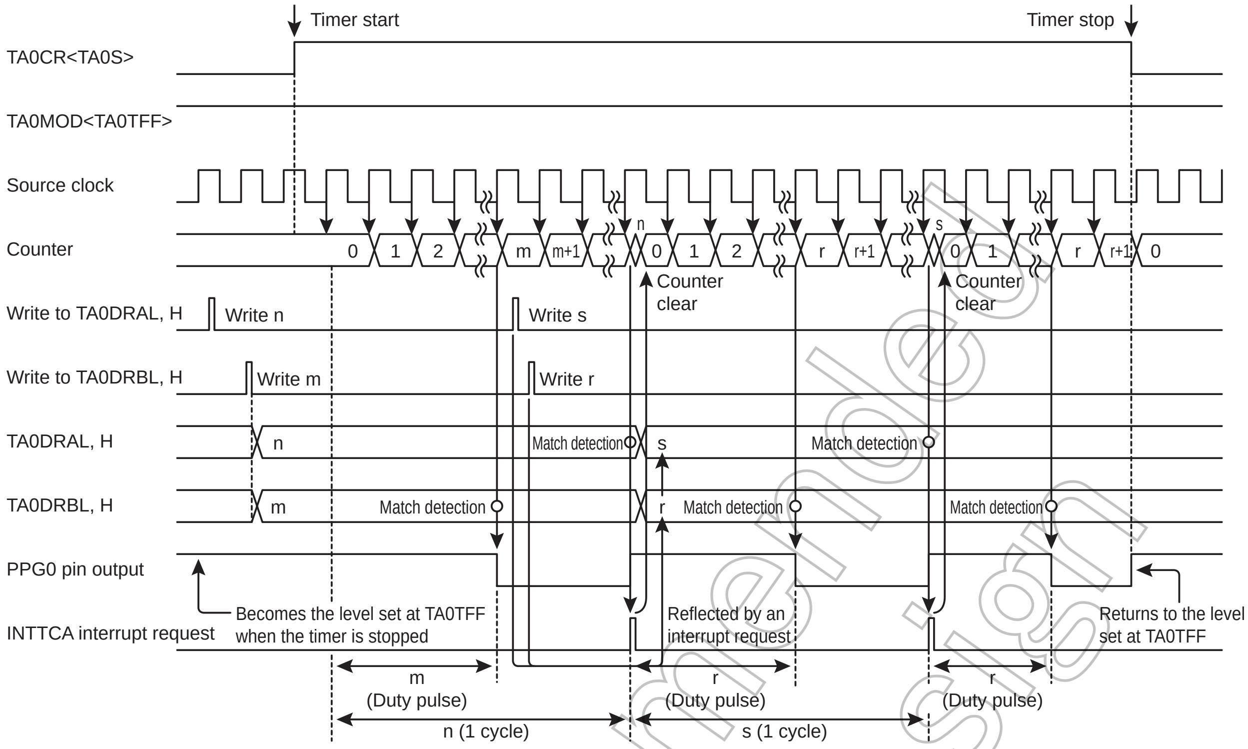
  When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L (TA0DRBH/L).

- When the double buffer is disabled

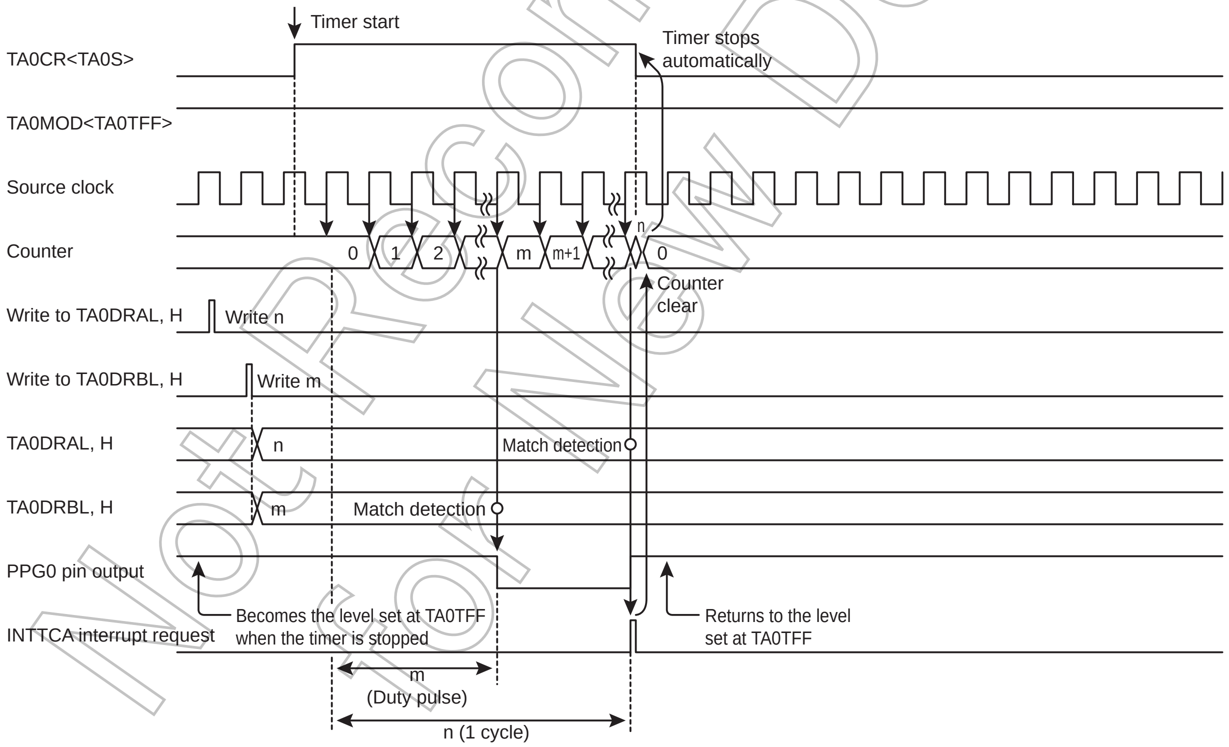
  When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is immediately stored in TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

  If the values set to TA0DRAH/L (TA0DRBH/L) are smaller than the up counter value, the up counter overflows and the match detection is executed using a new set value. As a result, the output pulse width may be longer than the set time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into TA0DRAH/L (TA0DRBH/L).



Continuous (TA0CR<TA0MPPG>="0")
Double buffer (TA0MOD<TA0DBE>="1")



One-shot (TA0CR<TA0MPPG>="1")

Figure 13-9 PPG Mode Timing Chart

13.5 Noise Canceller

The digital noise canceller can be used in the operation modes that use the TCA0 pin.

13.5.1 Setting

When the digital noise canceller is used, the input level is sampled at the sampling intervals set at TA0CR<TA0NC>. When the same level is detected three times consecutively, the level of the input to the timer is changed.

Setting TA0CR<TA0NC> to any values than "00" allows the noise canceller to start operation, regardless of the TA0CR<TA0S> value.

When the noise canceller is used, allow the timer to start after a period of time that is equal to four times the sampling interval after TA0CR<TA0NC> is set has elapsed. This stabilizes the input signal.

Set TA0CR<TA0NC> while the timer is stopped (TA0CR<TA0S> = "0"). When TA0CR<TA0S> is "1", writing is ignored.

In the SLOW 1/2 or SLEEP 1 mode, setting TA0CR<TA0NC> to "11" selects fs/2 as the source clock for the operation. Setting TA0CR<TA0NC> to "00" disables the noise canceller. Setting TA0CR<TA0NC> to "01" or "10" disables the TCA0 pin input.

Table 13-4 Noise Cancel Time (fcgck = 10 [MHz])

TA0NC	Sampling interval	Time removed as noise	Time regarded as signal
00	None	-	-
01	200 ns (2/fcgck)	600 ns or less	800 ns or more
10	400 ns (4/fcgck)	1.2 μs or less	1.6 μs or more
11	25.6 μs (256/fcgck)	76.8 μs or less	102.4 μs or more

13.6 Revision History

Rev	Description
RB000	"Figure 13-1 Timer Counter A0" "13.4.5.2 Operation" "Figure 13-7 Pulse Width Measurement Mode Timing Chart" Revised condition and timing of INTTA0 interrupt request generating for pulse width measurement mode.
RB001	Revised hex character from "H" to "0x".
	"13.4.5.1 Setting" Added TA0DRA and TA0DRB description (Initialized register).
	"13.4.5.3 Capture process" Added new chapter.
RB002	"Figure 13-7 Pulse Width Measurement Mode Timing Chart" Revised TA0MOD<TA0MCAP> value.
	Revised interrupt name from "INTTA0" to "INTTCA0". Revised timer input name from "TA0" to "TCA0".

14. 8-bit Timer Counter (TC0)

The TMP89CM42 contains 4 channels of high-performance 8-bit timer counters (TC0). Each timer can be used for time measurement and pulse output with a prescribed width. Two 8-bit timer counters are cascadable to form a 16-bit timer.

This chapter describes 2 channels of 8-bit timer counters 00 and 01. For 8-bit timer counters 02 and 03, replace the SFR addresses and pin names as shown in Table 14-1 and Table 14-2.

Table 14-1 SFR Address Assignment

	16-bit mode	T0xREG (Address)	T0xPWM (Address)	T0xMOD (Address)	T0xxCR (Address)	Low power consumption register
Timer counter 00	Lower	T00REG (0x0026)	T00PWM (0x0028)	T00MOD (0x002A)	T001CR (0x002C)	POFFCR0 <TC001EN>
Timer counter 01	Higher	T01REG (0x0027)	T01PWM (0x0029)	T01MOD (0x002B)		
Timer counter 02	Lower	T02REG (0x0F88)	T02PWM (0x0F8A)	T02MOD (0x0F8C)	T023CR (0x0F8E)	POFFCR0 <TC023EN>
Timer counter 03	Higher	T03REG (0x0F89)	T03PWM (0x0F8B)	T03MOD (0x0F8D)		

Table 14-2 Pin Names

	Timer input pin	PWM output pin	PPG output pin
Timer counter 00	TC00 pin	$\overline{\text{PWM0}}$ pin	$\overline{\text{PPG0}}$ pin
Timer counter 01	TC01 pin	$\overline{\text{PWM1}}$ pin	$\overline{\text{PPG1}}$ pin
Timer counter 02	TC02 pin	$\overline{\text{PWM2}}$ pin	$\overline{\text{PPG2}}$ pin
Timer counter 03	TC03 pin	$\overline{\text{PWM3}}$ pin	$\overline{\text{PPG3}}$ pin

14.1 Configuration

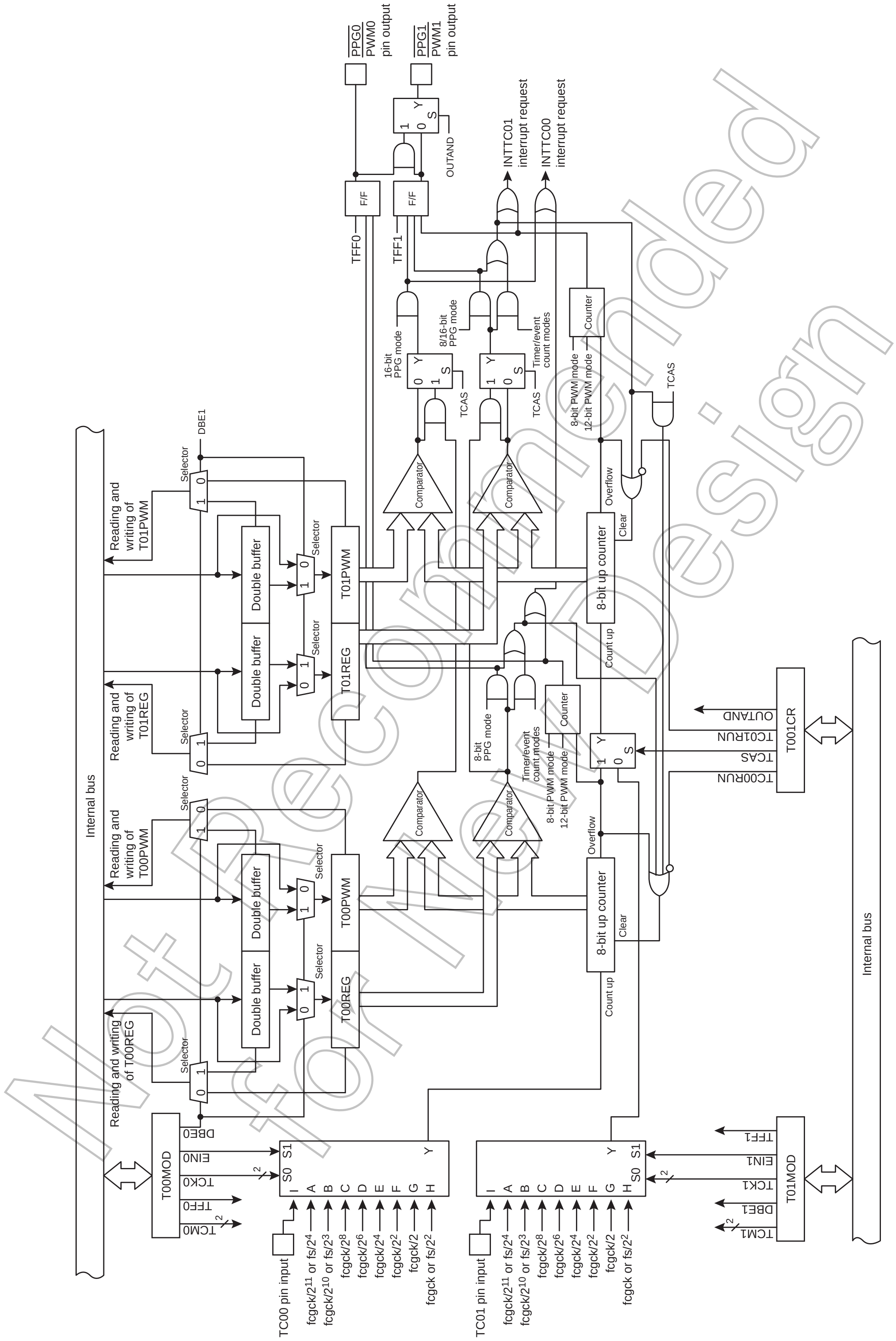


Figure 14-1 8-bit Timer Counters 00 and 01

14.2 Control

14.2.1 Timer counter 00

The timer counter 00 is controlled by the timer counter 00 mode register (T00MOD) and two 8-bit timer registers (T00REG and T00PWM).

Timer register 00

T00REG (0x0026)

	15	14	13	12	11	10	9	8
Bit Symbol	T00REG							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer register 00

T00PWM (0x0028)

	7	6	5	4	3	2	1	0
Bit Symbol	T00PWM							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Note 1: For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "14.4.3 8-bit pulse width modulation (PWM) output mode" and "14.4.7 12-bit pulse width modulation (PWM) output mode".

Timer counter 00 mode register

T00MOD (0x002A)		7	6	5	4	3	2	1	0
	Bit Symbol	TFF0	DBE0	TCK0			EIN0	TCM0	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF0	Timer F/F0 control	0	Clear		
		1	Set		
DBE0	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TCK0	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"	
		000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$
		001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$
		010	$fcgck/2^8$	$fcgck/2^8$	-
		011	$fcgck/2^6$	$fcgck/2^6$	-
		100	$fcgck/2^4$	$fcgck/2^4$	-
		101	$fcgck/2^2$	$fcgck/2^2$	-
		110	fcgck/2	fcgck/2	-
		111	fcgck	fcgck	$fs/2^2$
EIN0	Selection for using external source clock	0	Select the internal clock as the source clock.		
		1	Select an external clock as the source clock. (the falling edge of the TC00 pin)		
TCM0	Operation mode selection	00	8-bit timer/event counter modes		
		01	8-bit timer/event counter modes		
		10	8-bit pulse width modulation output (PWM) mode		
		11	8-bit programmable pulse generate (PPG) mode		

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T00MOD while the timer is stopped. Writing data into T00MOD is invalid during the timer operation.

Note 3: In the 8-bit timer/event modes, the TFF0 setting is invalid. In this mode, when the $\overline{PWM0}$ and $\overline{PPG0}$ pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4: When EIN0 is set to "1" and the external clock input is selected as the source clock, the TCK0 setting is ignored.

Note 5: When the T001CR<TCAS> bit is "1", timer 00 operates in the 16-bit mode. The T00MOD setting is invalid and timer 00 cannot be used independently in this mode. When the $\overline{PWM0}$ and $\overline{PPG0}$ pins are set to the function output pins in the port setting, the pins always output the "H" level.

Note 6: When the 16-bit mode is selected at T001CR<TCAS>, the timer start is controlled at T001CR<T01RUN>. Timer 00 is not started by writing data into T001CR<T00RUN>.

14.2.2 Timer counter 01

Timer counter 01 is controlled by timer counter 01 mode register (T01MOD) and two 8-bit timer registers (T01REG and T01PWM).

Timer register 01

T01REG		15	14	13	12	11	10	9	8
(0x0027)	Bit Symbol	T01REG							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x0029)	Bit Symbol	T01PWM							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note 1: For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "14.4.3 8-bit pulse width modulation (PWM) output mode" and "14.4.7 12-bit pulse width modulation (PWM) output mode".

Timer counter 01 mode register

T01MOD		7	6	5	4	3	2	1	0
(0x002B)	Bit Symbol	TFF1	DBE1	TCK1			EIN1	TCM1	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF1	Timer F/F1 control	0	Clear		
		1	Set		
DBE1	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TCK1	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"	
		000	fcgck/2^{11}	fs/2^4	fs/2^4
		001	fcgck/2^{10}	fs/2^3	fs/2^3
		010	fcgck/2^8	fcgck/2^8	-
		011	fcgck/2^6	fcgck/2^6	-
		100	fcgck/2^4	fcgck/2^4	-
		101	fcgck/2^2	fcgck/2^2	-
		110	fcgck/2	fcgck/2	-
		111	fcgck	fcgck	fs/2^2
EIN1	Selection for using external source clock	0	Select the internal clock as the source clock.		
		1	Select an external clock as the source clock. (the falling edge of the TC01 pin)		
TCM1	Operation mode selection		T001CR<TCAS>="0" (8-bit mode)	T001CR<TCAS>="1" (16-bit mode)	
		00	8-bit timer/event counter modes	16-bit timer/event counter modes	
		01	8-bit timer/event counter modes	16-bit timer/event counter modes	
		10	8-bit pulse width modulation output (PWM) mode	12-bit pulse width modulation output (PWM) mode	
		11	8-bit programmable pulse generate (PPG) mode	16-bit programmable pulse generate (PPG) mode	

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T01MOD while the timer is stopped. Writing data into T01MOD is invalid during the timer operation.

Note 3: In the 8-bit timer/event modes, the TFF1 setting is invalid. In this mode, when the $\overline{\text{PWM1}}$ and $\overline{\text{PPG1}}$ pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4: When EIN1 is set to "1" and the external clock input is selected as the source clock, the TCK1 setting is ignored.

14.2.3 Common to timer counters 00 and 01

Timer counters 00 and 01 have the low power consumption register (POFFCR0) and timer 00 and 01 control registers in common.

Low power consumption register 0

POFFCR0 (0x0F74)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	TC023EN	TC001EN	-	-	TCA1EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 control	0	Disable
		1	Enable
TC001EN	TC00,01 control	0	Disable
		1	Enable
TCA1EN	TCA1 control	0	Disable
		1	Enable
TCA0EN	TCA0 control	0	Disable
		1	Enable

Timer counter 01 control register

T001CR		7	6	5	4	3	2	1	0
(0x002C)	Bit Symbol	-	-	-	-	OUTAND	TCAS	T01RUN	T00RUN
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

OUTAND	Timers 00 and 01 output control	0	Output the timer 00 output from the $\overline{\text{PWM0}}$ and $\overline{\text{PPG0}}$ pins and the timer 01 output from the $\overline{\text{PWM1}}$ and $\overline{\text{PPG1}}$ pins.
		1	Output a pulse that is a logical ANDed product of the outputs of timers 00 and 01 from the $\overline{\text{PWM1}}$ and $\overline{\text{PPG1}}$ pins.
TCAS	Timers 00 and 01 cascade control	0	Use timers 00 and 01 independently (8-bit mode).
		1	Cascade timers 00 and 01 (16-bit mode).
T01RUN	Timer 01 control Timers 00/01 control (16-bit mode)	0	Stop and clear the counter
		1	Start
T00RUN	Timer 00 control	0	Stop and clear the counter
		1	Start

Note 1: When STOP mode is started, T00RUN and T01RUN are cleared to "0" and the timers stop. Set T001CR again to use timers 00 and 01 after STOP mode is released.

Note 2: When a read instruction is executed on T001CR, bits 7 to 4 are read as "0".

Note 3: When OUTAND is "1", output is obtained from the $\overline{\text{PWM1}}$ and $\overline{\text{PPG1}}$ pins only. There is no timer output to the $\overline{\text{PWM0}}$ and $\overline{\text{PPG0}}$ pins. If the PWM0 and PPG0 pins are set as the function output pins in the port setting, the pins always output "H".

Note 4: OUTAND and TCAS can be changed only when both TC01RUN and TC00RUN are "0". When either TC01RUN or TC00RUN is "1" or both are "1", the register values remain unchanged by executing write instructions on OUTAND and TCAS. OUTAND and TCAS can be changed at the same time as TC01RUN and TC00RUN are changed from "0" to "1".

14.2.4 Operation modes and usable source clocks

The operations modes of the 8-bit timers and the usable source clocks are listed below.

Table 14-3 Operation Modes and Usable Source Clocks (NORMAL1/2 and IDLE1/2 modes)

TCK0		000	001	010	011	100	101	110	111	TC0i pin input
Operation mode		$fcgck/2^{11}$ or $fs/2^4$	$fcgck/2^{10}$ or $fs/2^3$	$fcgck/2^8$	$fcgck/2^6$	$fcgck/2^4$	$fcgck/2^2$	fcgck/2	fcgck	
8-bit timer modes	8-bit timer	O	O	O	O	O	O	O	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	O	O	O	O	O	O	-
	8-bit PPG	O	O	O	O	O	O	O	O	-
16-bit timer modes	16-bit timer	O	O	O	O	O	O	O	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	O	O	O	O	O	O	O
	16-bit PPG	O	O	O	O	O	O	O	O	O

Note 1: O: Usable, -: Unusable

Note 2: Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3: When the low-frequency clock, fs, is not oscillating, it must not be selected as the source clock. If fs is selected when it is not oscillating, no source clock is supplied to the timer, and the timer remains stopped.

Note 4: i=0, 1 (i=0 only in the 16-bit modes)

Note 5: The operation modes of the 8-bit timers and the usable source clocks are listed below.

Table 14-4 Operation Modes and Usable Source Clocks (SLOW1/2 and SLEEP1 modes)

TCK0		000	001	010	011	100	101	110	111	TC0i pin input
Operation mode		$fs/2^4$	$fs/2^3$	-	-	-	-	-	$fs/2^2$	
8-bit timer modes	8-bit timer	O	O	-	-	-	-	-	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	-	-	-	-	-	O	-
	8-bit PPG	O	O	-	-	-	-	-	O	-
16-bit timer modes	16-bit timer	O	O	-	-	-	-	-	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	-	-	-	-	-	O	O
	16-bit PPG	O	O	-	-	-	-	-	O	O

Note 1: O: Usable, -: Unusable

Note 2: Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3: i=0, 1 (i=0 only in the 16-bit modes)

14.3 Low Power Consumption Function

Timer counters 00 and 01 have the low power consumption registers (POFFCR0) that save power when the timers are not used.

Setting POFFCR0<TC001EN> to "0" disables the basic clock supply to timer counters 00 and 01 to save power. Note that this renders the timers unusable. Setting POFFCR0<TC001EN> to "1" enables the basic clock supply to timer counters 00 and 01 and allows the timers to operate.

After reset, POFFCR0<TC001EN> are initialized to "0", and this makes the timers unusable. When using the timers for the first time, be sure to set POFFCR0<TC001EN> to "1" in the initial setting of the program (before the timer control registers are operated).

Do not change POFFCR0<TC001EN> to "0" during the timer operation. Otherwise timer counters 00 and 01 may operate unexpectedly.

14.4 Functions

Timer counters TC00 and TC01 have 8-bit modes in which they are used independently and 16-bit modes in which they are cascaded.

The 8-bit modes include four operation modes; the 8-bit timer mode, the 8-bit event counter mode, the 8-bit pulse width modulation output (PWM) mode and the 8-bit programmable pulse generated output (PPG) mode.

The 16-bit modes include four operation modes; the 16-bit timer mode, the 16-bit event counter mode, the 12-bit PWM mode and the 16-bit PPG mode.

14.4.1 8-bit timer mode

In the 8-bit timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times. The operation of TC00 is described below, and the same applies to the operation of TC01. (Replace TC00- by TC01-).

14.4.1.1 Setting

TC00 is put into the 8-bit timer mode by setting T00MOD<TCM0> to "00" or "01", T001CR<TCAS> to "0" and T00MOD<EIN0> to "0". Select the source clock at T00MOD<TCK0>. Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

14.4.1.2 Operation

Setting T001CR<T00RUN> to "1" allows the 8-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR<T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

14.4.1.3 Double buffer

The double buffer can be used for T00REG by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00REG during the timer operation, the set value is initially stored in the double buffer, and T00REG is not immediately updated. T00REG compares the previous set value with the up counter value. When the values match, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00REG. Subsequently, the match detection is executed using a new set value.

  When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in both the double buffer and T00REG.

- When the double buffer is disabled

  When a write instruction is executed on T00REG during the timer operation, the set value is immediately stored in T00REG. Subsequently, the match detection is executed using a new set value.

  If the value set to T00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If the value set to T00REG is equal to the up counter value, the match detection is executed immediately after data is written into T00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock (Figure 14-3). If these are problems, enable the double buffer.

When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in T00REG.

When a read instruction is executed on T00REG, the last value written into T00REG is read out, regardless of the T00MOD<DBE0> setting.

Table 14-5 8-bit Timer Mode Resolution and Maximum Time Setting

T00MOD <TCK0>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	52.2ms	124.5ms
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	26.1ms	62.3ms
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	6.5ms	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	1.6ms	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	408μs	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	102μs	-
110	fcgck/2	fcgck/2	-	200ns	-	51μs	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	25.5μs	31.1ms

(Example) Operate TC00 in the 8-bit timer mode with the operation clock of $fcgck/2^2$ [Hz] and generate interrupts at 64 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xE8       ; Selects the 8-bit timer mode and fcgck/2^2
LD      (T00REG),0xA0       ; Sets the timer register (64μs / (2^2/fcgck) = 0xA0)
SET     (T001CR).0          ; Starts TC00
```

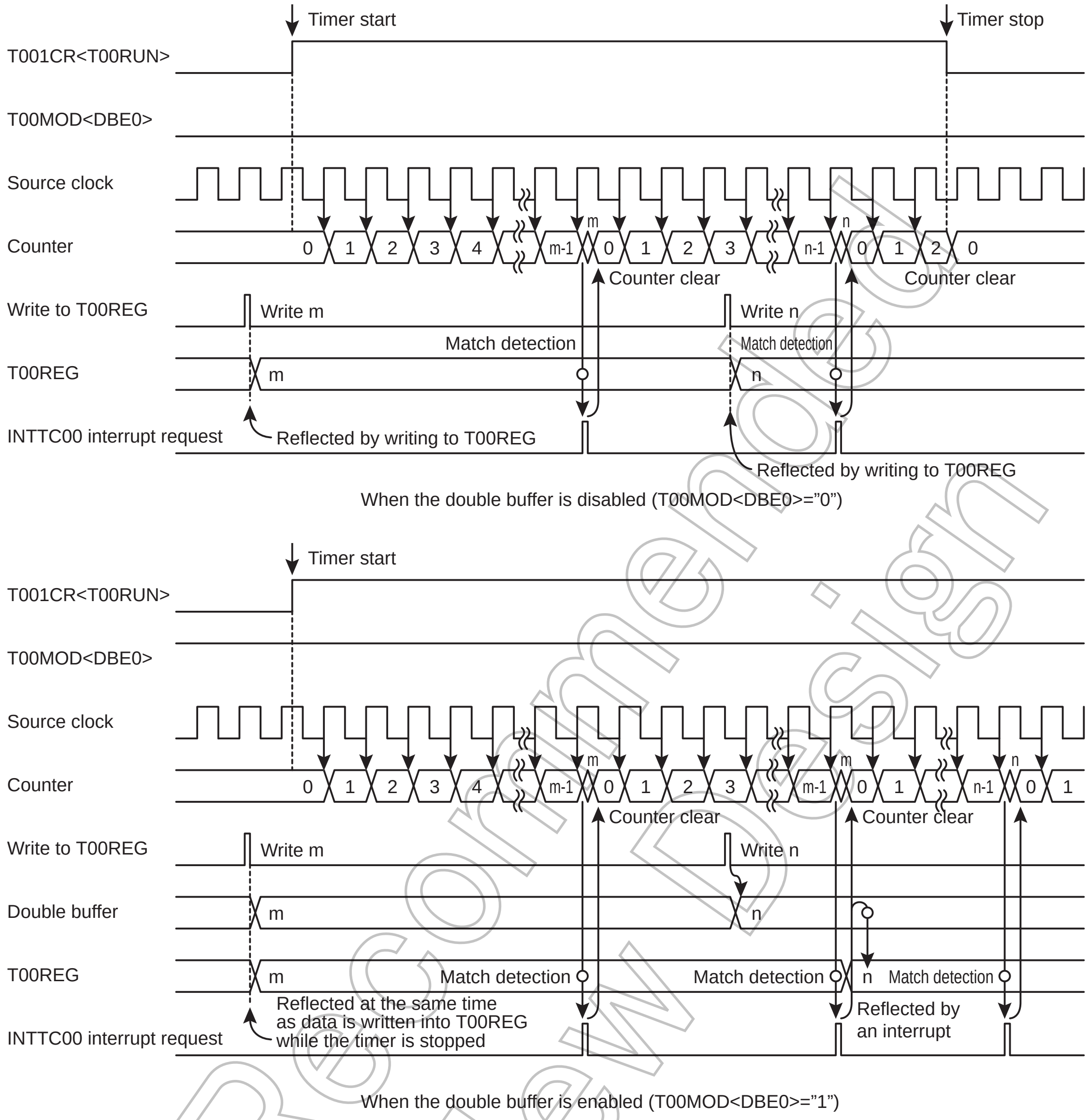


Figure 14-2 Timer Mode Timing Chart

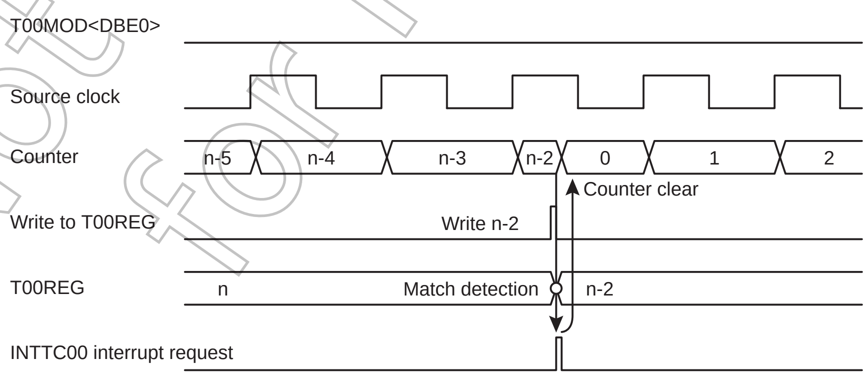


Figure 14-3 Operation When T00REG and the Up Counter Have the Same Value

14.4.2 8-bit event counter mode

In the 8-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 or TC01 pin. The operation of TC00 is described below, and the same applies to the operation of TC01.

14.4.2.1 Setting

TC00 is put into the 8-bit event counter mode by setting T00MOD<TCM0> to "00", T001CR<TCAS> to "0" and T00MOD<EIN0> to "1". Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

14.4.2.2 Operation

Setting T001CR<T00RUN> to "1" allows the 8-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up-counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR<T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

The maximum frequency to be supplied is fcgck/2^2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/24 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

14.4.2.3 Double buffer

Refer to "14.4.1.3 Double buffer".

(Example) Operate TC00 in the 8-bit event counter mode and generate an interrupt each time 16 falling edges are detected at the TC00 pin.

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xC4       ; Selects to the 8-bit event counter mode
LD      (T00REG),0x10       ; Sets the timer register
SET     (T001CR).0          ; Starts TC00
```

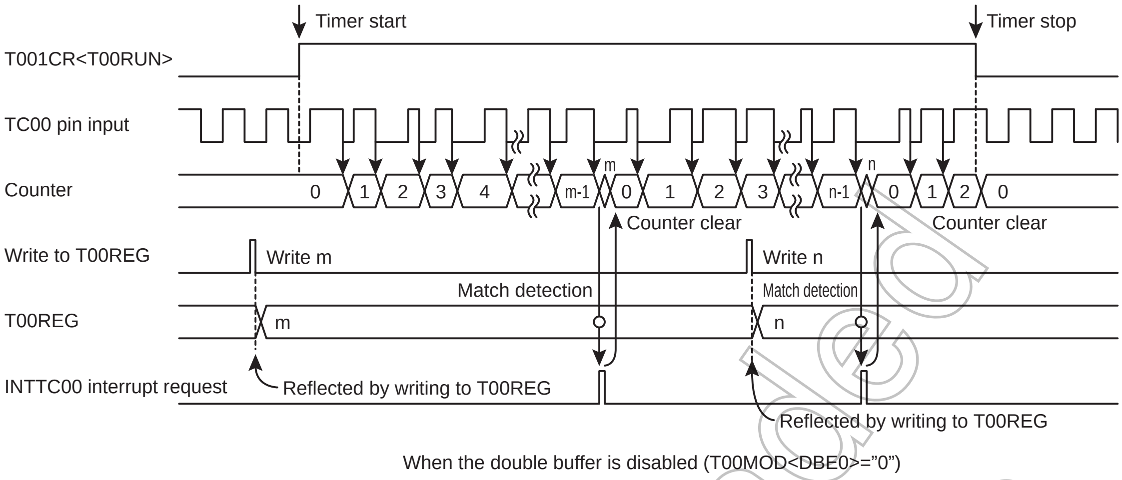


Figure 14-4 Event Counter Mode Timing Chart

14.4.3 8-bit pulse width modulation (PWM) output mode

The pulse-width modulated pulses with a resolution of 7 bits are output in the 8-bit PWM mode. An additional pulse can be added to the 2 × n-th duty pulse. This enables PWM output with a resolution nearly equivalent to 8 bits. (n=1, 2, 3...)

The operation of TC00 is described below, and the same applies to the operation of TC01.

14.4.3.1 Setting

TC00 is put into the 8-bit PWM mode by setting T00MOD<TCM0> to "10" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the count value to be used for the match detection and the additional pulse value at the PWM register T00PWM.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

In the 8-bit PWM mode, the T00PWM register is configured as follows:

Timer register 00

T00PWM (0x0028)		7	6	5	4	3	2	1	0
	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM (0x0029)		7	6	5	4	3	2	1	0
	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

PWMDUTY is a 7-bit register used to set the duty pulse width value (the time before the first output change) in a cycle (128 counts of the source clock).

PWMAD is a register used to set the additional pulse. When PWMAD is "1", an additional pulse that corresponds to 1 count of the source clock is added to the 2 × n-th duty pulse (n=1, 2, 3...). In other words, the 2 × n-th duty pulse has the output of PWMDUTY+1.

The additional pulse is not added when PWMAD is "0".

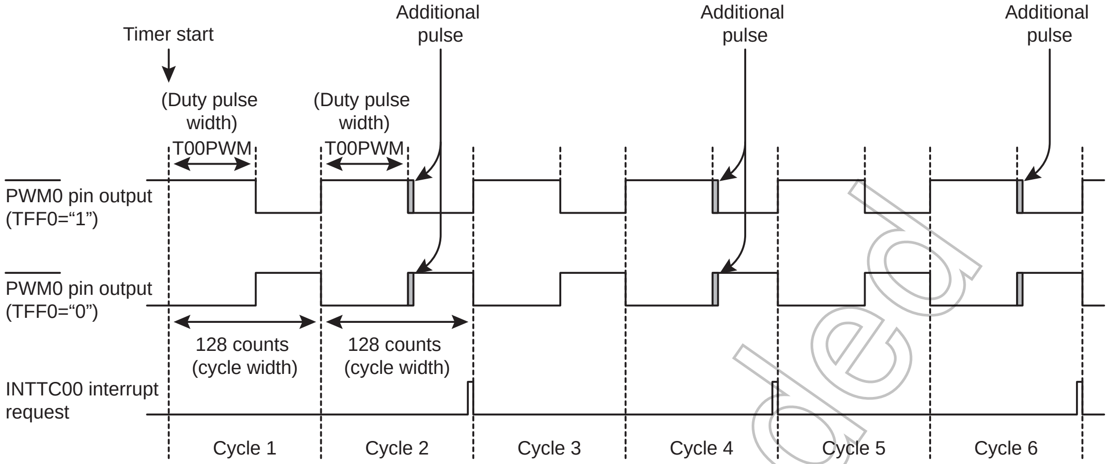


Figure 14-5 $\overline{\text{PWM0}}$ Pulse Output

Set the initial state of the $\overline{\text{PWM0}}$ pin at T00MOD<TFF0>. Setting T00MOD<TFF0> to "0" selects the "L" level as the initial state of the $\overline{\text{PWM0}}$ pin. Setting T00MOD<TFF0> to "1" selects the "H" level as the initial state of the $\overline{\text{PWM0}}$ pin. If the $\overline{\text{PWM0}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PWM0}}$ pin. Table 14-6 shows the list of output levels of the $\overline{\text{PWM0}}$ pin.

Table 14-6 List of Output Levels of $\overline{\text{PWM0}}$ Pin

TFF0	$\overline{\text{PWM0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM <PWMDUTY> matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

And by setting "1" to T001CR<OUTAND> bit, a logical product (AND) pulse of TC00 and TC01's output can be output to $\overline{\text{PWM0}}$ pin. By using this function, the remote-control waveform can be created eaily.

14.4.3.2 Operations

Setting T001CR<T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 7 bits of the up counter value and the value set to T00PWM<PWMDUTY> is detected, the output of the $\overline{\text{PWM0}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM0}}$ pin changes from the "L" to "H" level. When T00MOD<TFF0> is "1", the $\overline{\text{PWM0}}$ pin changes from the "H" to "L" level.

If T00PWM<PWMAD> is "1", an additional pulse that corresponds to 1 count of the source clock is added at the 2 × n-th match detection (n=1, 2, 3...). In other words, the $\overline{\text{PWM0}}$ pin output is reversed at the timing of T00PWM<PWMDUTY>+1. When T00MOD<TFF0> is "0", the period of the "L" level becomes longer than the value set to T00<PWMDUTY> by 1 source clock. When T00MOD<TFF0> is "1", the period of the "H" level becomes longer than the value set to T00PWM<PWMDUTY> by 1 source clock. This function allows two cycles of output pulses to be handled with a resolution nearly equivalent to 8 bits.

No additional pulse is inserted when T00PWM<PWMAD> is "0".

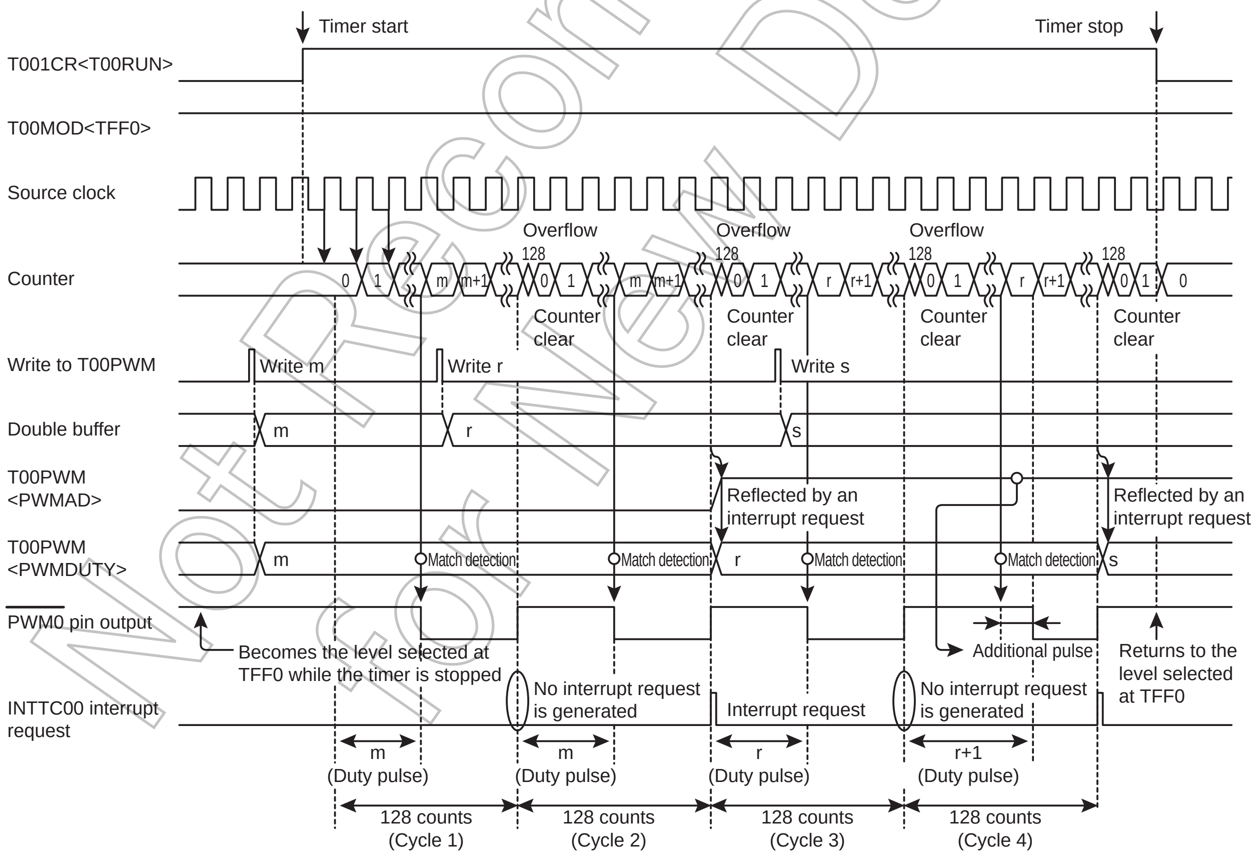
Subsequently, the up counter continues counting up. When the up counter value reaches 128, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the $\overline{\text{PWM0}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM0}}$ pin changes from the "H" to "L" level. When T00MOD<TFF0>

is "1", the $\overline{\text{PWM0}}$ pin changes from the "L" to "H" level. If the 2 × n-th overflow occurs at this time, an INTTC00 interrupt request is generated. (No interrupt request is generated at the 2 × n-th -1 overflow.) Subsequently, the up counter continues counting up.

When T001CR<T00RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM0}}$ pin returns to the level selected at T00MOD<TFF0>.

(Example) Operate TC00 in the 8-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 11.6 μs (fcgck = 10 MHz)
(Actually, output a total duty pulse of 23.2 μs in 2 cycles (51.2 μs))

```
SET   (P7FC).0          ; Sets P7FC0 to "1"
SET   (P7CR).0          ; Sets P7CR0 to "1"
LD    (POFFCR0),0x10    ; Sets TC001EN to "1"
DI                      ; Sets the interrupt master enable flag to "disable"
SET   (EIRH).4          ; Sets the INTTC00 interrupt enable register to "1"
EI                      ; Sets the interrupt master enable flag to "enable"
LD    (T00MOD),0xF2     ; Selects the 8-bit PWM mode and fcgck/2
LD    (T00PWM),0x74     ; Sets the timer register (duty pulse)
                        ; (11.6μs × 2) / (2/fcgck) = 0x74
SET   (T001CR).0        ; Starts TC00
```



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 14-6 8-bit PWM Mode Timing Chart

14.4.3.3 Double buffer

The double buffer can be used for T00PWM by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00PWM during the timer operation, the set value is first stored in the double buffer, and T00PWM is not updated immediately. T00PWM compares the previous set value with the up counter value. When the 2 × n-th overflow occurs, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00PWM. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T00PWM, the value in the double buffer (the last set value) is read out, not the T00PWM value (the currently effective value).

  When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM.

- When the double buffer is disabled

  When a write instruction is executed on T00PWM during the timer operation, the set value is immediately stored in T00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T00PWM is smaller than the up counter value, the $\overline{\text{PWM0}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM. Therefore, the timing of changing the $\overline{\text{PWM0}}$ pin may not be an integral multiple of the source clock (Figure 14-7). Similarly, if T00PWM is set during the additional pulse output, the timing of changing the $\overline{\text{PWM0}}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in T00PWM.

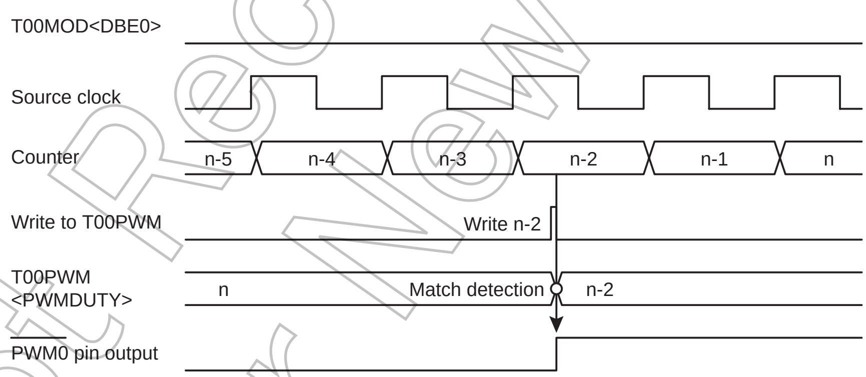


Figure 14-7 Operation When T00PWM and the Up Counter Have the Same Value

Table 14-7 Resolutions and Cycles in the 8-bit PWM Mode

T00MOD <TCK0>	Source clock [Hz]			Resolution		7-bit cycle (period × 2)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	26.2ms (52.4ms)	62.5ms (125ms)
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	13.1ms (26.2ms)	31.3ms (62.5ms)
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	3.3ms (6.6ms)	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	819.2μs (1638.4μs)	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	204.8μs (409.6μs)	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	51.2μs (102.4μs)	-
110	fcgck/2	fcgck/2	-	200ns	-	25.6μs (51.2μs)	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	12.8μs (25.6μs)	15.6ms (31.3ms)

14.4.4 8-bit programmable pulse generate (PPG) output mode

In the 8-bit PPG mode, the pulses with arbitrary duty and cycle are output by using the T00REG and T00PWM registers.

By setting the T001CR<OUTAND> register, a pulse that is a logical ANDed product of the TC00 and TC01 outputs can be output to the TC01 pin. This function facilitates the generation of remote-controlled waveforms, for example.

The operation of TC00 is described below, and the same applies to the operation of TC01.

14.4.4.1 Setting

TC00 is put into the 8-bit PPG mode by setting T00MOD<TCM0> to "10" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the duty pulse width at T00PWM and the cycle width at T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

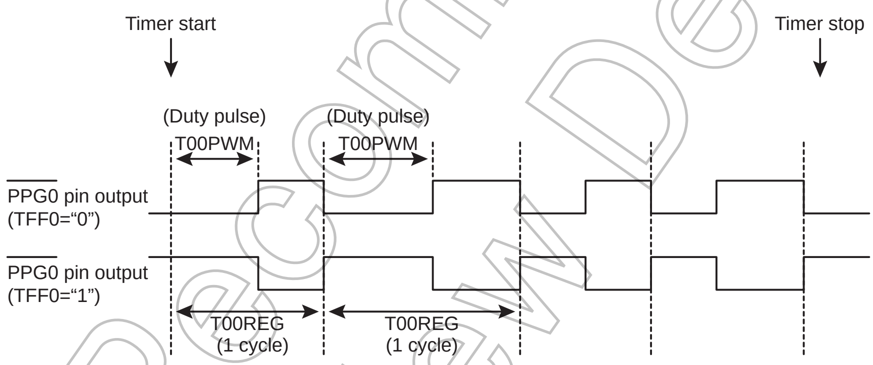


Figure 14-8 $\overline{\text{PPG0}}$ Pulse Output

Set the initial state of the $\overline{\text{PPG0}}$ pin at T00MOD<TFF0>. Setting T00MOD<TFF0> to "0" selects the "L" level as the initial state of the $\overline{\text{PPG0}}$ pin. Setting T00MOD<TFF0> to "1" selects the "H" level as the initial state of the $\overline{\text{PPG0}}$ pin. If the $\overline{\text{PPG0}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PPG0}}$ pin. Table 14-8 shows the list of output levels of the $\overline{\text{PPG0}}$ pin.

Table 14-8 List of Output Levels of $\overline{\text{PPG0}}$ Pin

TFF0	$\overline{\text{PPG0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM matched	T00REG matched	Operation stop-ped (initial state)
0	L	H	L	L
1	H	L	H	H

Setting the T001CR<OUTAND> bit to "1" allows the $\overline{\text{PPG0}}$ pin to output a pulse that is a logical ANDed product of the TC00 and TC01 outputs.

14.4.4.2 Operation

Setting T001CR<T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the internal up counter value and the value set to T00PWM is detected, the output of the $\overline{\text{PPG0}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PPG0}}$ pin changes from the "L" to "H" level. When T00MOD<TFF0> is "1", the $\overline{\text{PPG0}}$ pin changes from the "H" to "L" level.

Subsequently, the up counter continues counting up. When a match between the up counter value and T00REG is detected, the output of the $\overline{\text{PPG0}}$ pin is reversed again. When T00MOD<TFF0> is "0", the $\overline{\text{PPG0}}$ pin changes from the "H" to "L" level. When T00MOD<TFF0> is "1", the $\overline{\text{PPG0}}$ pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated.

When T001CR<T00RUN> is set to "0" during the operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PPG0}}$ pin returns to the level selected at T00MOD<TFF0>.

14.4.4.3 Double buffer

The double buffer can be used for T00PWM and T00REG by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00PWM (T00REG) during the timer operation, the set value is first stored in the double buffer, and T00PWM (T00REG) is not updated immediately. T00PWM (T00REG) compares the previous set value with the up counter value. When an INTTC00 interrupt request is generated, the double buffer set value is stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T00PWM (T00REG) value (the currently effective value).

  When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM (T00REG).

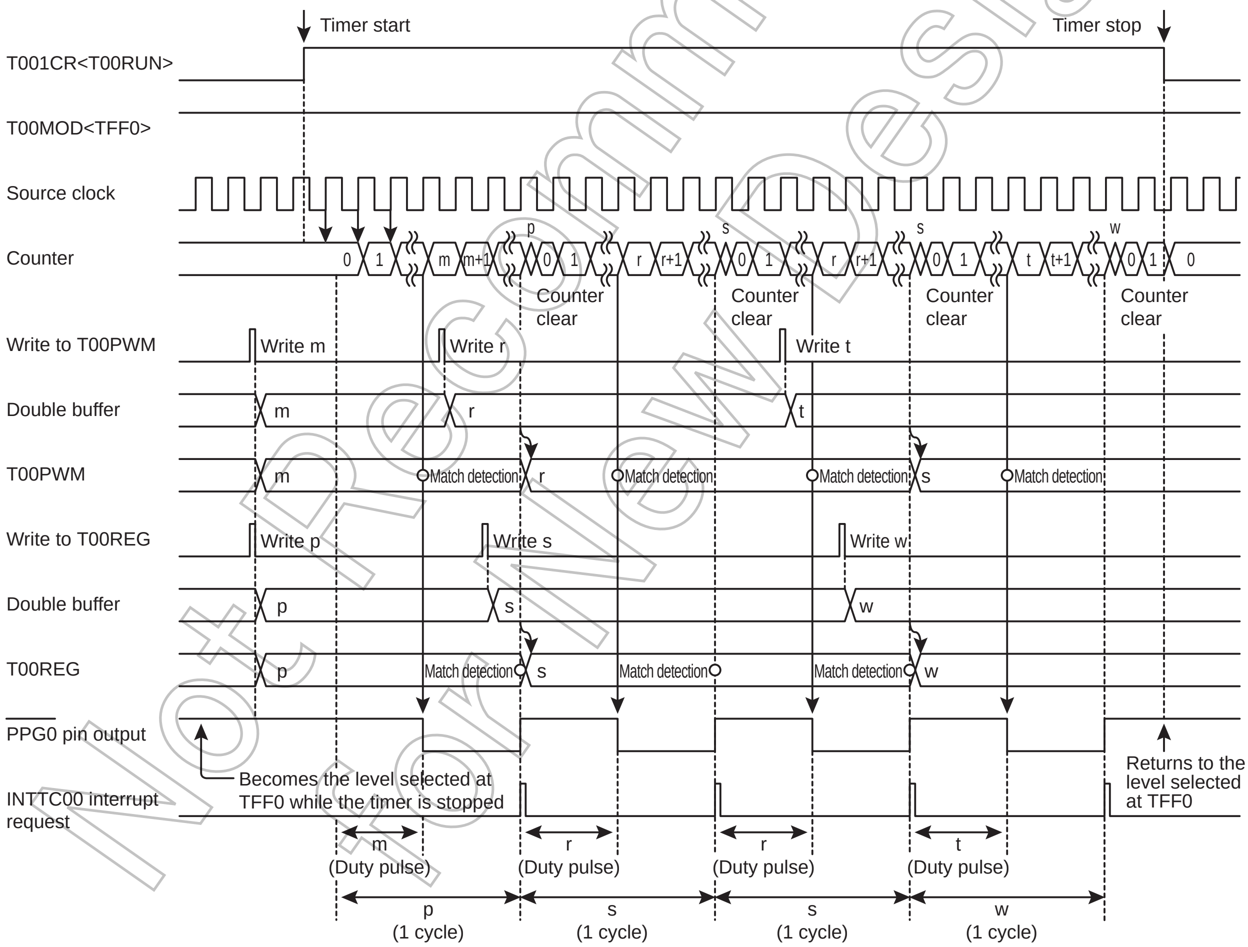
- When the double buffer is disabled

  When a write instruction is executed on T00PWM (T00REG) during the timer operation, the set value is immediately stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value. If the value set to T00PWM (T00REG) is smaller than the up counter value, the $\overline{\text{PPG0}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T00PWM (T00REG) is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM (T00REG). Therefore, the timing of changing the $\overline{\text{PPG0}}$ pin may not be an integral multiple of the source clock (Figure 14-10). If these are problems, enable the double buffer.

  When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in T00PWM (T00REG).

(Example) Operate TC00 in the 8-bit PPG mode with the operation clock of fcgck/2 and output the 8μs duty pulse in 32μs cycles (fcgck = 10 MHz)

```
SET     (P7FC).0            ; Sets P7FC0 to "1"
SET     (P7CR).0            ; Sets P7CR0 to "1"
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xF3       ; Selects the 8-bit PPG mode and fcgck/2
LD      (T00REG),0xA0       ; Sets the timer register (cycle)
                            ; 32μs / (2/fcgck) = 0xA0
LD      (T00PWM),0x28       ; Sets the timer register (duty pulse)
                            ; 8μs / (2/fcgck) = 0x28
SET     (T001CR).0          ; Starts TC00
```



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 14-9 8-bit PPG Mode Timing Chart

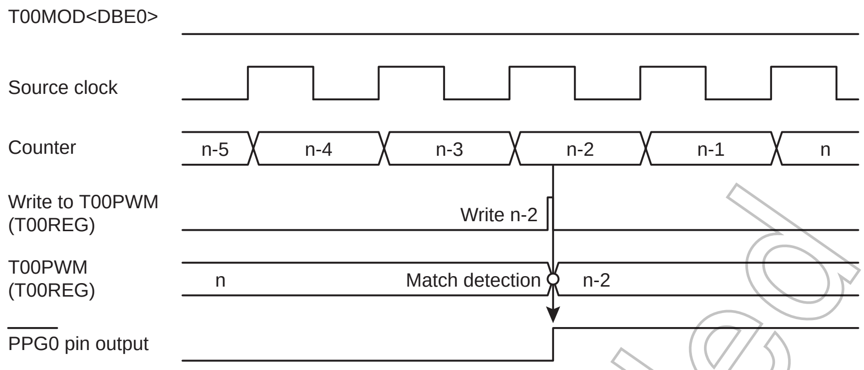


Figure 14-10 Operation When T00PWM (T00REG) and the Up Counter Have the Same Value

14.4.5 16-bit timer mode

In the 16-bit timer mode, TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

14.4.5.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN1> to "0". Select the source clock at T01MOD<TCK1>.

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

14.4.5.2 Operations

Setting T001CR<T01RUN> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

14.4.5.3 Double buffer

The double buffer can be used for T01+00REG by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is first stored in the double buffer, and T01+00REG is not updated immediately. T01+00REG compares the previous set value with the up counter value. When the values are matched, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

  When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00REG.

- When the double buffer is disabled

  When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is immediately stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

  If the value set to T01+00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If the value set to T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in T01+00REG.

When a read instruction is executed on T01+00REG, the last value written into T01+00REG is read out, regardless of the T00MOD<DBE1> setting.

(Example) Operate TC00 and TC01 in the 16-bit timer mode with the operation clock of fcgck/2 [Hz] and generate interrupts at 96 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10     ; Sets TC001EN to "1"
DI                         ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4           ; Sets the INTTC00 interrupt enable register to "1"
EI                         ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF0      ; Selects the 16-bit timer mode and fcgck/2
LD      (T00REG),0xE0      ; Sets the timer register (96μs / (2/fcgck) = 0x1E0)
LD      (T01REG),0x01      ; Sets the timer register
LD      (T001CR),0x06      ; Starts TC00 and TC001 (16-bit mode)
```

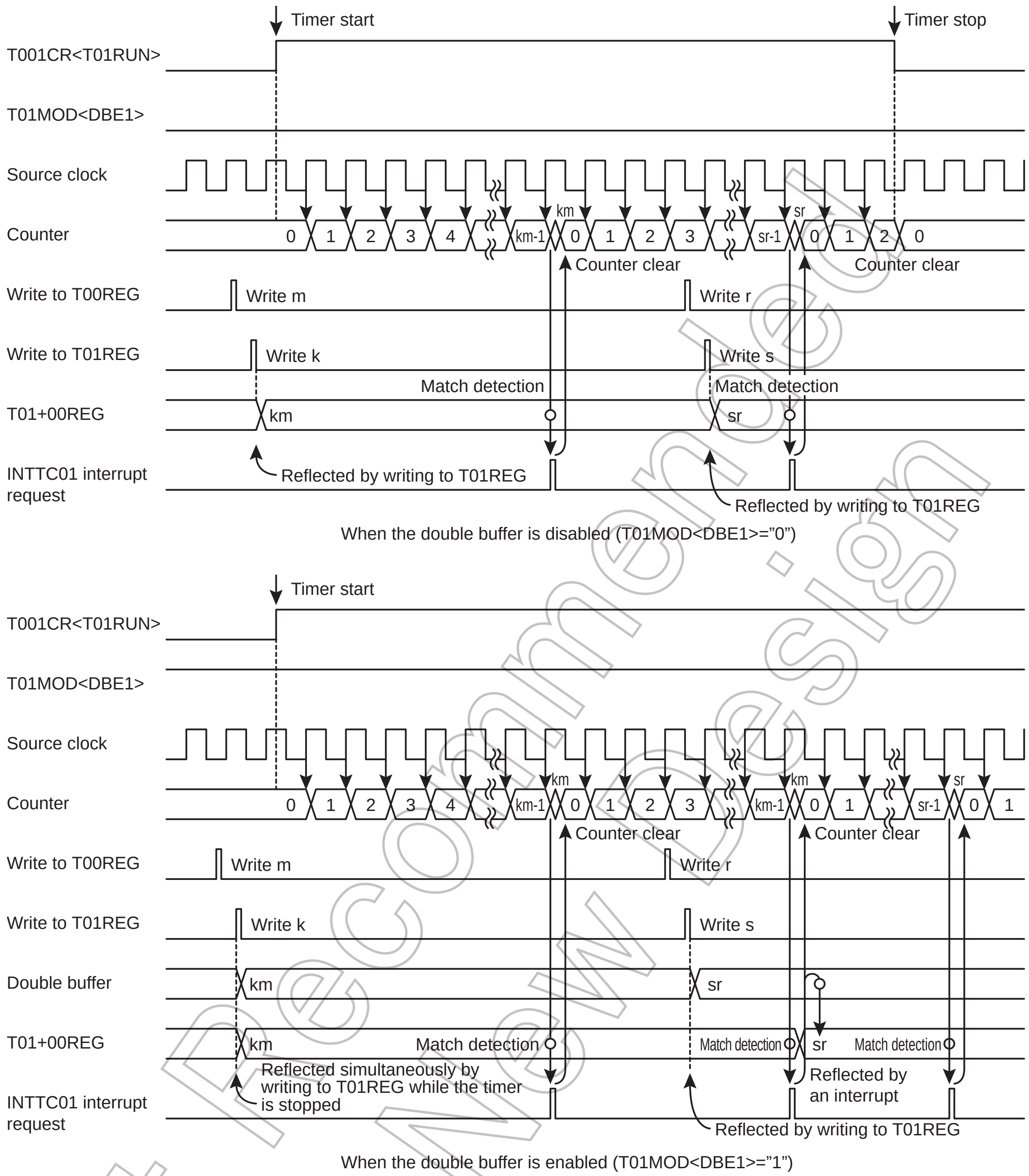


Figure 14-11 16-bit Timer Counter Timing Chart

Table 14-9 16-bit Timer Mode Resolution and Maximum Time Setting

T01MOD <TCK1>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	13.4s	32s
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	6.7s	16s
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	1.7s	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	419.4ms	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	104.9ms	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	26.2ms	-
110	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	6.6ms	8s

14.4.6 16-bit event counter mode

In the 16-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 pin. TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

14.4.6.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN0> to "1".

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and set the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

14.4.6.2 Operations

Setting T001CR<T01RUN> to "1" allows the 16-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

14.4.6.3 Double buffer

Refer to 14.4.5.3.

(Example) Operate TC00 and TC01 in the 16-bit event counter mode and generate an interrupt each time the 384th falling edge is detected at the TC00 pin

```
LD      (POFFCR0),0x10    ; Sets TC001EN to "1"
DI                        ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4          ; Sets the INTTC00 interrupt enable register to "1"
EI                        ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xC4     ; Selects the 16-bit event counter mode
LD      (T00REG),0x80     ; Sets the timer register
LD      (T01REG),0x10     ; Sets the timer register
LD      (T001CR),0x06     ; Starts TC00 and TC001 (16-bit mode)
```

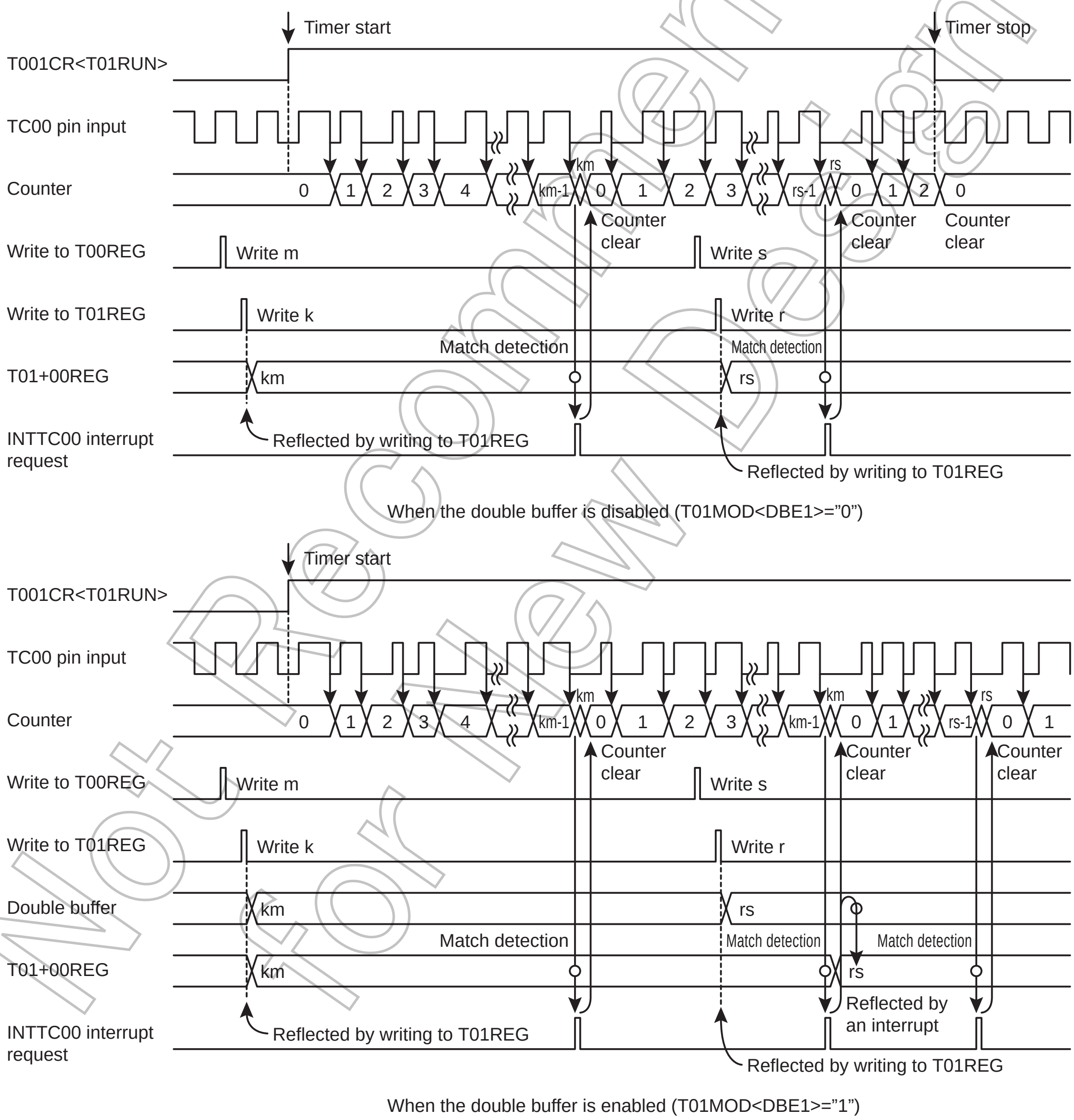


Figure 14-12 16-bit Event Counter Mode Timing Chart

14.4.7 12-bit pulse width modulation (PWM) output mode

In the 12-bit PWM output mode, TC00 and TC01 are cascaded to output the pulse-width modulated pulses with a resolution of 8 bits. An additional pulse of 4 bits can be inserted, which enables PWM output with a resolution nearly equivalent to 12 bits.

14.4.7.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 12-bit PWM mode is selected by setting T01MOD<TCM1> to "10". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN1> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

Set the count value to be used for the match detection and the additional pulse value as a 12-bit value at the timer registers T00PWM and T01PWM. Set bits 11 to 8 of the 12-bit value at the lower 4 bits of T01PWM and set bits 7 to 0 at T00PWM. Refer to the following table for the register configuration. Hereinafter, the 12-bit value specified by the combined setting of T00PWM and T01PWM is indicated as T01+00PWM. The timer register settings are reflected on the double buffer or T01+00PWM when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00PWM and T01PWM in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x0028)	Bit Symbol	PWMDUTYL				PWMAD3	PWMAD2	PWMAD1	PWMAD0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x0029)	Bit Symbol					PWMDUTYH			
	Read/Write					R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Bits 7 to 4 of T01PWM are not used in the 12-bit PWM mode. However, data can be written to these bits of T01PWM and the written values are read out as they are when the bits are read. Normally, set these bits to "0".

PWMDUTYH and PWMDUTYL are 4-bit registers. They are combined to set an 8-bit value of duty pulse width (time before the first change in the output) for one cycle (256 counts of the source clock). Hereinafter, an 8-bit value specified by the combined setting of PWMDUTYH and PWMDUTYL is indicated as PWMDUTY.

PWMAD3 to 0 are the additional pulse setting register. Additional pulses can be inserted in specific cycles of the duty pulse by setting each bit to "1". The additional pulses are inserted in the positions listed in Table 14-10. PWMAD 3 to 0 can be combined to specify the number of times of inserting the additional pulses in 16 cycles to any number from 1 to 16. Examples of inserting additional pulses are shown in Figure 14-13.

Table 14-10 Cycles in Which Additional Pulses Are Inserted

	Cycles in which additional pulses are inserted among cycles 1 to 16
PWMAD0="1"	9
PWMAD1="1"	5, 13
PWMAD2="1"	3, 7, 11, 15
PWMAD3="1"	2, 4, 6, 8, 10, 12, 14, 16

Set the initial state of the $\overline{\text{PWM1}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the "L" level as the initial state of the $\overline{\text{PWM1}}$ pin. Setting T01MOD<TFF1> to "1" selects the "H" level as the initial state of the $\overline{\text{PWM1}}$ pin. If the $\overline{\text{PWM1}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the $\overline{\text{PWM1}}$ pin. Table 14-11 shows the list of output levels of the $\overline{\text{PWM1}}$ pin.

Table 14-11 List of Output Levels of $\overline{\text{PWM1}}$ Pin

TFF1	$\overline{\text{PWM1}}$pin output level			
	Before the start of operation (initial state)	PWMDUTY matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

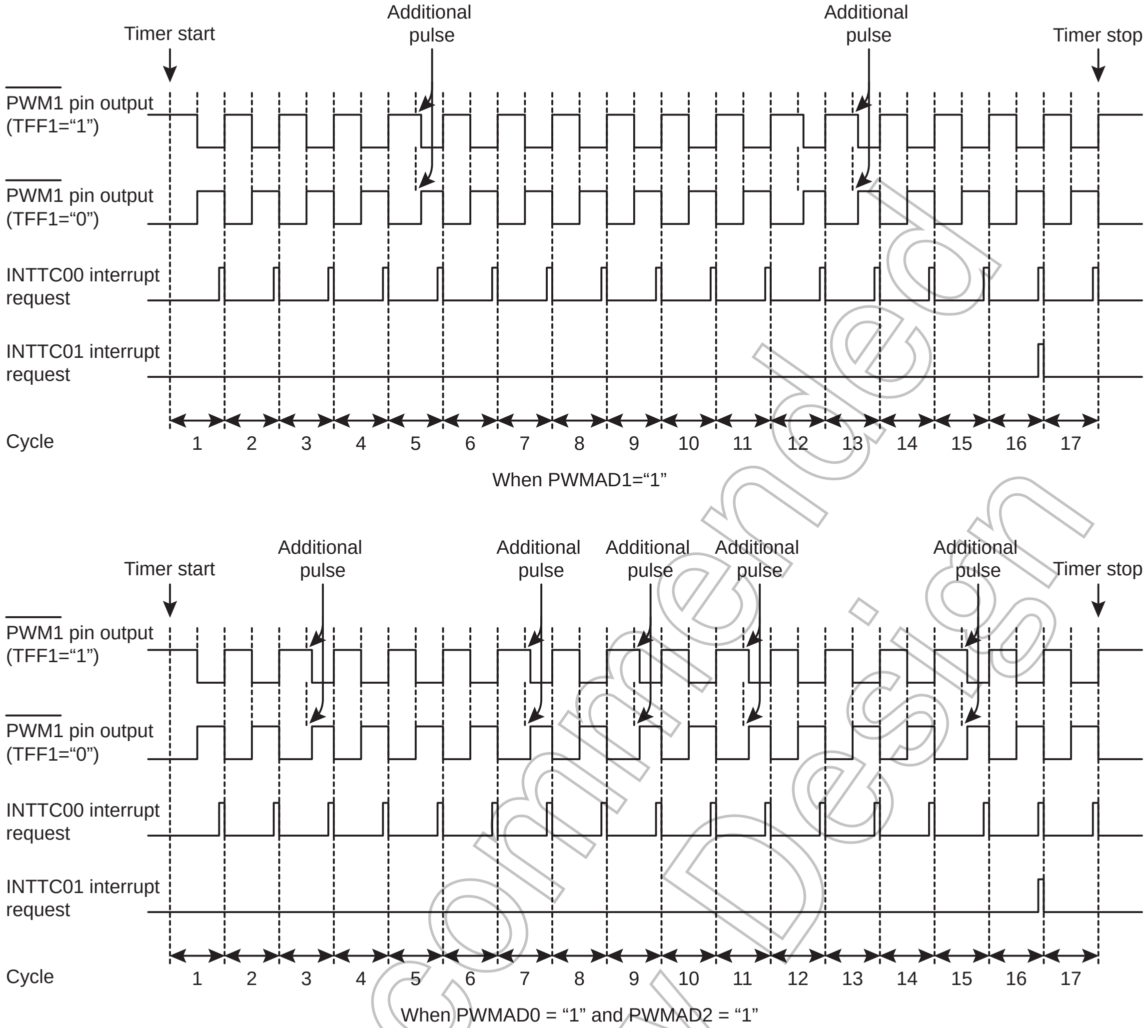


Figure 14-13 Examples of Inserting Additional Pulses

14.4.7.2 Operations

Setting T001CR<T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 8 bits of the up counter value and the value set to PWMDUTY is detected, the output of the $\overline{\text{PWM1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM1}}$ pin changes from the "L" to "H" level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM1}}$ pin changes from the "H" to "L" level.

If any of PWMAD3 to 0 is "1", an additional pulse that corresponds to 1 count of the source clock is inserted in specific cycles of the duty pulse. In other words, the $\overline{\text{PWM1}}$ pin output is reversed at the timing of PWMDUTY+1. When T00MOD<TFF0> is "0", the period of the "L" level becomes longer than the value set to PWMDUTY by 1 source clock. When T00MOD<TFF0> is "1", the period of the "H" level becomes longer than the value set to PWMDUTY by 1 source clock. This function allows 16 cycles of output pulses to be handled with a resolution nearly equivalent to 12 bits.

No additional pulse is inserted when PWMAD3 to 0 are all "0".

Subsequently, the up counter continues counting up. When the up counter value reaches 256, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the $\overline{\text{PWM1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM1}}$ pin changes from the "H" to "L" level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM1}}$ pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated (an INTTC00 interrupt request is generated each time an overflow occurs.) An INTTC01 interrupt request is generated at the 16 × n-th overflow (n=1, 2, 3...). Subsequently, the up counter continues counting up.

When T001CR<T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM1}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

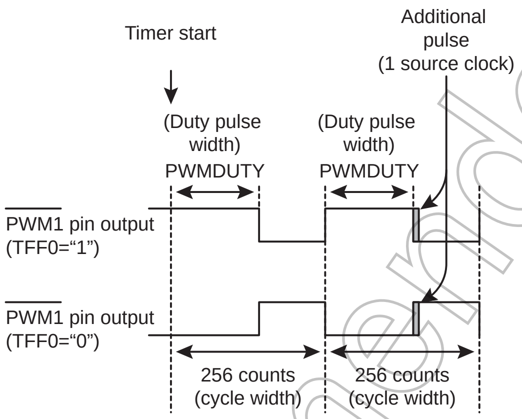


Figure 14-14 $\overline{\text{PWM1}}$ Pin Output

14.4.7.3 Double buffer

The double buffer can be used for T01+00PWM by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is first stored in the double buffer, and T01+00PWM is not updated immediately. T01+00PWM compares the previous set value with the up counter value. When the 16 × n-th overflow occurs, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00PWM. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T01+00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T01+00PWM value (the currently effective value).

  When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00PWM.

- When the double buffer is disabled

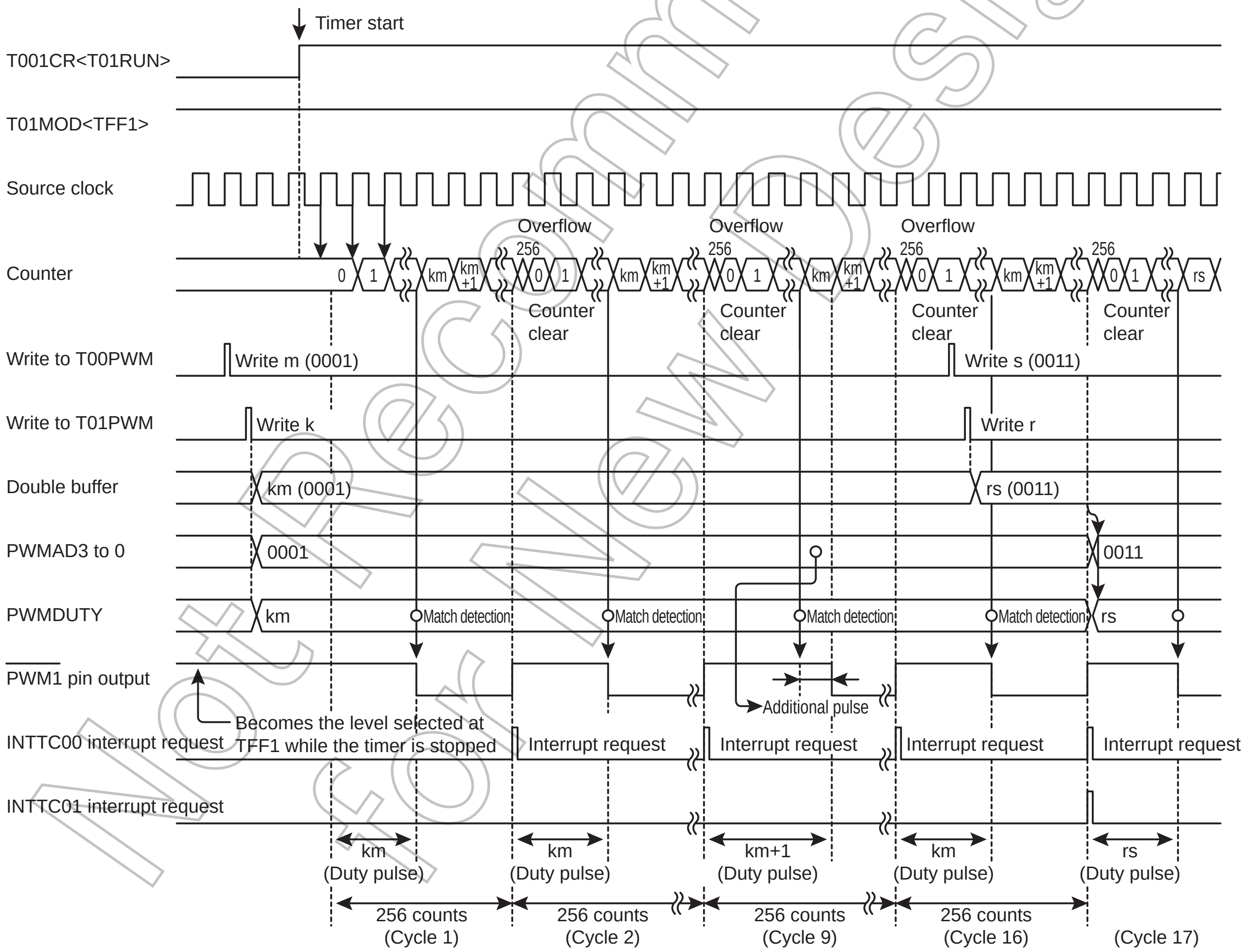
  When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is immediately stored in T01+00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T01+00PWM is smaller than the up counter value, the $\overline{\text{PWM1}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM. Therefore, the timing of changing the $\overline{\text{PWM1}}$ pin may not be an integral multiple of the source clock. Similarly, if T01+00PWM is set during the additional pulse output, the timing of changing the $\overline{\text{PWM1}}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in T01+00PWM.

(Example) Operate TC00 and TC01 in the 12-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 14.0625 μs in 51.2μs cycles (fcgck = 10 MHz)

(Actually, output a duty pulse of 225 μs in total in 16 cycles (819.2 μs))

```
SET     (P7FC).1            ; Sets P7FC1 to "1"
SET     (P7CR).1            ; Sets P7CR1 to "1"
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF2       ; Selects the 12-bit PWM mode and fcgck/2
LD      (T00PWM),0x65       ; Sets the timer register (duty pulse)
                            ; (14.0625μs × 16) / (2/fcgck) = 0x465
LD      (T01PWM),0x04       ; Sets the timer register (duty pulse)
LD      (T001CR),0x06       ; Starts TC00 and TC01
```



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 14-15 12-bit PWM Mode Timing Chart

Table 14-12 Resolutions and Cycles in the 12-bit PWM Mode

T01MOD <TCK1>	Source clock [Hz]			Resolution		8-bit cycle (period × 16)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	52.4ms (838.9ms)	125ms (2000ms)
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	26.2ms (419.4ms)	62.5ms (1000ms)
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	6.6ms (104.9ms)	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	1.6ms (26.2ms)	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	409.6μs (6.6ms)	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	102.4μs (1.6ms)	-
110	fcgck/2	fcgck/2	-	200ns	-	51.2μs (819.2μs)	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	25.6μs (409.6μs)	31.3ms (500ms)

14.4.8 16-bit programmable pulse generate (PPG) output mode

In the 16-bit PPG mode, TC00 and TC01 are cascaded to output the pulses that have a resolution of 16 bits and arbitrary pulse width and duty. Two 16-bit registers, T01+00REG and T01+00PWM, are used to output the pulses. This enables output of longer pulses than an 8-bit timer.

14.4.8.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit PPG mode is selected by setting T01MOD<TCM1> to "11". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN0> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Set the count value that corresponds to a cycle as a 16-bit value at the timer registers T01REG and T00REG. Set the count value that corresponds to a duty pulse as a 16-bit value at T01PWM and T00PWM (hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG, and the 16-bit value specified by the combined setting of T01PWM and T00PWM is indicated as T01+00PWM). The timer register settings are reflected on the double buffer or T01+00PWM and T01+00REG when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00REG, T01REG and T00PWM before executing a write instruction on T01PWM. (When data is written to T01PWM, the set values of the four timer registers become effective at the same time.)

Set the initial state of the $\overline{\text{PPG1}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the "L" level as the initial state of the $\overline{\text{PPG1}}$ pin. Setting T01MOD<TFF1> to "1" selects the "H" level as the initial state of the $\overline{\text{PPG1}}$ pin. If the $\overline{\text{PPG1}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the $\overline{\text{PPG1}}$ pin. Table 14-13 shows the list of output levels of the $\overline{\text{PPG1}}$ pin.

Table 14-13 List of Output Levels of $\overline{\text{PPG1}}$ Pin

TFF1	$\overline{\text{PPG1}}$ pin output level			
	Before the start of operation (initial state)	T01+00PWM matched	T01+00REG matched	Operation stop-ped (initial state)
0	L	H	L	L
1	H	L	H	H

14.4.8.2 Operations

Setting T001CR<T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the up counter value and the value set to T01+00PWM is detected, the output of the $\overline{\text{PPG1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PPG1}}$ pin changes from the "L" to "H" level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG1}}$ pin changes from the "H" to "L" level. At this time, an INTTC00 interrupt request is generated.

The up counter continues counting up. When a match between the up counter value and the value set to T01+00REG is detected, the output of the $\overline{\text{PPG1}}$ pin is reversed again. When T01MOD<TFF1> is "0", the $\overline{\text{PPG1}}$ pin changes from the "H" to "L" level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG1}}$ pin changes from the "L" to "H" level. At this time, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000".

When T001CR<T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x0000". The $\overline{\text{PPG1}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

14.4.8.3 Double buffer

The double buffer can be used for T01+00PWM and T01+00REG by setting T01MOD<DBE1>. The double buffer is enabled by setting T01MOD<DBE1> to "0" or disabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are first stored in the double buffer, and T01+00PWM and T01+00REG are not updated immediately. T01+00PWM and T01+00REG compare the previous set values with the up counter value. When a match between the up counter value and the T01+00REG set value is detected, an INTTC01 interrupt request is generated and the double buffer set values are stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in both the double buffer and T01+00PWM and T01+00REG.

- When the double buffer is disabled

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are immediately stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

  If the value set to T01+00PWM or T01+00REG is smaller than the up counter value, the $\overline{\text{PPG1}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM or T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM and T01 +00REG. Therefore, the timing of changing the $\overline{\text{PPG1}}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in T01+00PWM and T01+00REG.

When read instructions are executed on T01+00PWM and T01+00REG, the last value written into T01 +00REG is read out, regardless of the T00MOD<DBE1> setting.

(Example) Operate TC00 and TC01 in the 16-bit PPG mode with the operation clock of fcgck/2 and output the 68μs duty pulse in 96μs cycles (fcgck = 10 MHz)

```
SET    (P7FC).1            ; Sets P7FC0 to "1"
SET    (P7CR).1            ; Sets P7CR0 to "1"
LD     (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                         ; Sets the interrupt master enable flag to "disable"
SET    (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                         ; Sets the interrupt master enable flag to "enable"
LD     (T01MOD),0xF3       ; Selects the 8-bit PPG mode and fcgck/2
LD     (T00REG),0xE0       ; Sets the timer register (cycle)
LD     (T01REG),0x01       ; Sets the timer register (cycle)
                           ; 96μs / (2/fcgck) = 0x01E0
LD     (T00PWM),0x54       ; Sets the timer register (duty pulse)
LD     (T01PWM),0x01       ; Sets the timer register (duty pulse)
                           ; 68μs / (2/fcgck) = 0x0154
LD     (T001CR),0x06       ; Starts TC00 and TC01
```

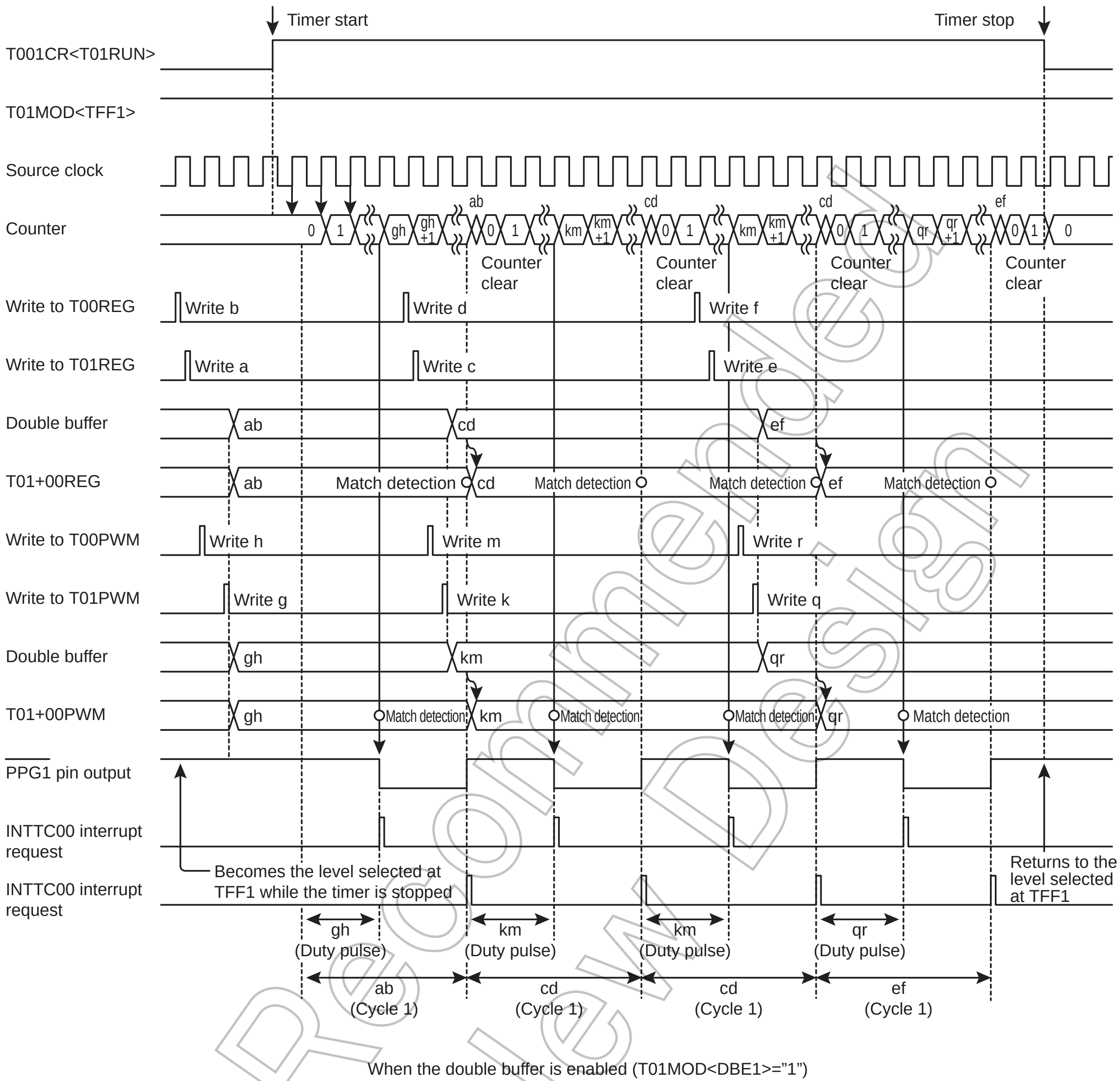


Figure 14-16 16-bit PPG Output Mode Timing Chart

14.5 Revision History

Rev	Description
RA003	Revised interrupt name from "INTT00" and "INTT01" to "INTTC00" and "INTTC01". Added upper bar to PWM and PPG label.
RA004	"14.4.3 8-bit pulse width modulation (PWM) output mode" Revised Exsample program.
	"Figure 14-15 12-bit PWM Mode Timing Chart" Revised each item name.
RA005	"Figure 14-1 8-bit Timer Counters 00 and 01" Revised source clock from "fc" to "fcgck". "14.4.7 12-bit pulse width modulation (PWM) output mode" Revised Example program.

15. Real Time Clock (RTC)

The real time clock is a function that generates interrupt requests at certain intervals using the low-frequency clock.

The number of interrupts is counted by the software to realize the clock function.

The real time clock can be used only in the operation modes where the low-frequency clock oscillates, except for SLEEP0.

15.1 Configuration

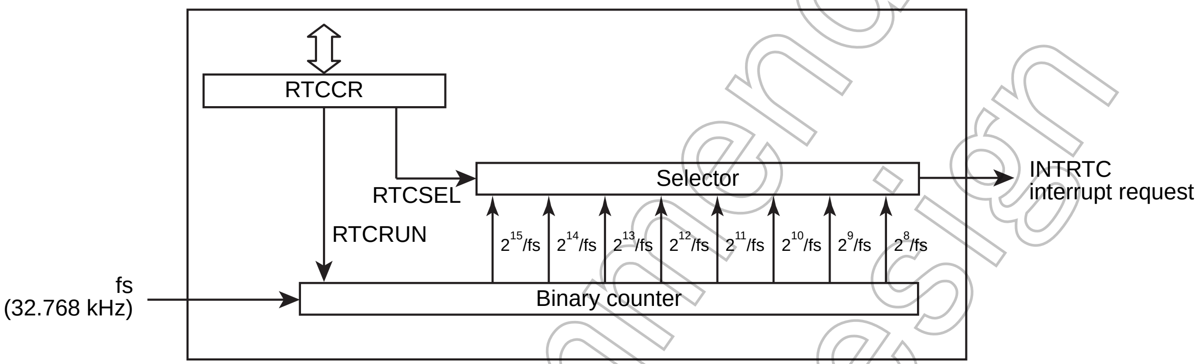


Figure 15-1 Real Time Clock

15.2 Control

The real time clock is controlled by following resisters.

Low power consumption register 2

POFFCR2		7	6	5	4	3	2	1	0
(0x0F76)	Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0	Disable
		1	Enable
SIO0EN	SIO0 control	0	Disable
		1	Enable

Real time clock control register

RTCCR (0x0FC8)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	RTCSEL			RTCRUN
	Read/Write	R	R	R	R	R/W			R/W
	After reset	0	0	0	0	0	0	0	0

RTCSEL	Selects the interrupt generation interval	000 : 2^{15}/fs (1.000 [s] @fs=32.768kHz) 001 : 2^{14}/fs (0.500 [s] @fs=32.768kHz) 010 : 2^{13}/fs (0.250 [s] @fs=32.768kHz) 011 : 2^{12}/fs (125.0 [ms] @fs=32.768kHz) 100 : 2^{11}/fs (62.50 [ms] @fs=32.768kHz) 101 : 2^{10}/fs (31.25 [ms] @fs=32.768kHz) 110 : 2^{9}/fs (15.62 [ms] @fs=32.768kHz) 111 : 2^{8}/fs (7.81 [ms] @fs=32.768kHz)
RTCRUN	Enables/disables the real time clock operation	0 : Disable 1 : Enable

Note 1: fs: Low-frequency clock [Hz]

Note 2: RTCCR<RTCSEL> can be rewritten only when RTCCR<RTCRUN> is "0". If data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective. RTCCR<RTCSEL> can be rewritten at the same time as enabling the real time clock, but it cannot be rewritten at the same time as disabling the real time clock.

Note 3: If the real time clock is enabled and when 1) SYSCR2<XTEN> is cleared to "0" to stop the low-frequency clock oscillation circuit or 2) the operation is changed to the STOP mode or the SLEEP0 mode, the data in RTCCR<RTCSEL> is maintained and RTCCR<RTCRUN> is cleared to "0".

15.3 Function

15.3.1 Low Power Consumption Function

Real time clock has the low power consumption registers (POFFCR2) that save power when the real time clock is not being used.

Setting POFFCR2<RTCEN> to "0" disables the basic clock supply to real time clock to save power. Note that this renders the real time clock unusable. Setting POFFCR2<RTCEN> to "1" enables the basic clock supply to real time clock and allows the real time clock to operate.

After reset, POFFCR2<RTCEN> are initialized to "0", and this renders the real time clock unusable. When using the real time clock for the first time, be sure to set POFFCR2<RTCEN> to "1" in the initial setting of the program (before the real time clock control registers are operated).

Do not change POFFCR2<RTCEN> to "0" during the real time clock operation. Otherwise real time clock may operate unexpectedly.

15.3.2 Enabling/disabling the real time clock operation

Setting RTCCR<RTCRUN> to "1" enables the real time clock operation. Setting RTCCR<RTCRUN> to "0" disables the real time clock operation.

RTCCR<RTCRUN> is cleared to "0" just after reset release.

15.3.3 Selecting the interrupt generation interval

The interrupt generation interval can be selected at RTCCR<RTCSEL>.

RTCCR<RTCSEL> can be rewritten only when RTCCR<RTCRUN> is "0". If data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective.

RTCCR<RTCSEL> can be rewritten at the same time as enabling the real time clock operation, but it cannot be rewritten at the same time as disabling the real time clock operation.

15.4 Real Time Clock Operation

15.4.1 Enabling the real time clock operation

Set the interrupt generation interval to RTCCR<RTCSEL>, and at the same time, set RTCCR<RTCRUN> to "1".

When RTCCR<RTCRUN> is set to "1", the binary counter for the real time clock starts counting of the low-frequency clock.

When the interrupt generation interval selected at RTCCR<RTCSEL> is reached, a real time clock interrupt request (INTRTC) is generated and the counter continues counting.

15.4.2 Disabling the real time clock operation

Clear RTCCR<RTCRUN> to "0".

When RTCCR<RTCRUN> is cleared to "0", the binary counter for the real time clock is cleared to "0" and stops counting of the low-frequency clock.

16. Asynchronous Serial Interface (UART)

The TMP89CM42 contains 2 channels of asynchronous serial interfaces (UART).

This chapter describes asynchronous serial interface 0 (UART0). For UART1, replace the SFR addresses and pin names as shown in Table 16-1 and Table 16-2.

Table 16-1 SFR Address Assignment

	UARTxCR1 (address)	UARTxCR2 (address)	UARTxDR (address)	UARTxSR (address)	RDxBUF (address)	TDxBUF (address)
UART0	UART0CR1 (0x001A)	UART0CR2 (0x001B)	UART0DR (0x001C)	UART0SR (0x001D)	RD0BUF (0x001E)	TD0BUF (0x001E)
UART1	UART1CR1 (0x0F54)	UART1CR2 (0x0F55)	UART1DR (0x0F56)	UART1SR (0x0F57)	RD1BUF (0x0F58)	TD1BUF (0x0F58)

Table 16-2 Pin Names

	Serial data input pin	Serial data output pin
UART0	RXD0 pin	TXD0 pin
UART1	RXD1 pin	TXD1 pin

16.1 Configuration

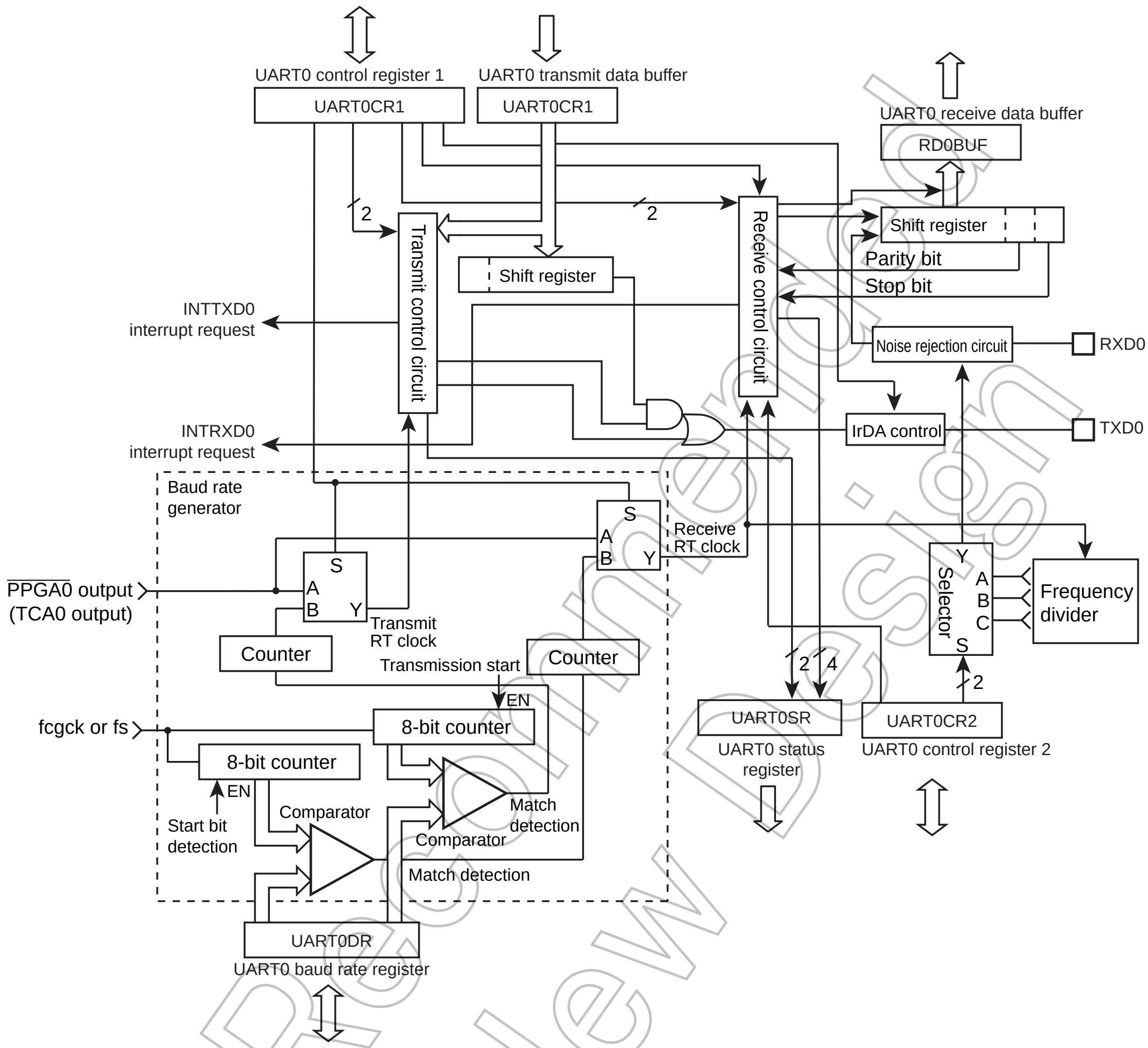


Figure 16-1 Asynchronous Serial Interface (UART)

16.2 Control

UART0 is controlled by the low power consumption registers (POFFCR1), UART0 control registers 1 and 2 (UART0CR1 and UART0CR2) and the UART0 baud rate register (UART0DR). The operating status can be monitored using the UART status register (UART0SR).

Low power consumption register 1

POFFCR1 (0x0F75)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	SBI0EN	-	-	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0 1	Disable Enable
UART1EN	UART1 control	0 1	Disable Enable
UART0EN	UART0 control	0 1	Disable Enable

UART0 control register 1

UART0CR1		7	6	5	4	3	2	1	0
(0x001A)	Bit Symbol	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
	After reset	0	0	0	0	0	0	0	0

TXE	Transmit operation	0: 1:	Disable Enable	
RXE	Receive operation	0: 1:	Disable Enable	
STOPBT	Transmit stop bit length	0: 1:	1 bit 2 bits	
EVEN	Parity selection	0: 1:	Odd-numbered parity Even-numbered parity	
PE	Parity addition	0: 1:	No parity Parity added	
IRDASEL	TXD pin output selection	0: 1:	UART output IrDA output	
BRG	Transfer base clock selection		When SYSCR2<SYSCK> is "0"	When SYSCR2<SYSCK> is "1"
		0:	fcgck	fs
		1:	TCA0 output	

Note 1: fcgck, Gear clock; fs, Low-frequency clock

Note 2: If the TXE or RXE bit is set to "0" during the transmission or receiving of data, the operation is not disabled until the data transfer is completed. At this time, the data stored in the transmit data buffer is discarded.

Note 3: EVEN, PE and BRG settings are common to transmission and receiving.

Note 4: Set RXE and TXE to "0" before changing BRG.

Note 5: When BRG is set to the TCA0 output, the RT clock becomes asynchronous and the start bit of the transmitted/received data may get shorter by a maximum of (UART0DR+1)/(Transfer base clock frequency)[s].

If the pin is not used for the TCA0 output, control the TCA0 output by using the port function control register.

Note 6: To prevent STOPBT, EVEN, PE, IRDASEL and BRG from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "16.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed ".

Note 7: When the STOP, IDLE0 or SLEEP0 mode is activated, TXE and RXE are cleared to "0" and the UART stops. Other bits keep their values.

UART0 control register 2

UART0CR2		7	6	5	4	3	2	1	0
(0x001B)	Bit Symbol	-	-	RTSEL			RXDNC		STOPBR
	Read/Write	R	R	R/W			R/W		R/W
	After reset	0	0	0	0	0	0	0	0

			Odd-numbered bits of transfer frame	Even-numbered bits of transfer frame
RTSEL	Selects the number of RT clocks	000:	16 clocks	16 clocks
		001:	16 clocks	17 clocks
		010:	15 clocks	15 clocks
		011:	15 clocks	16 clocks
		100:	17 clocks	17 clocks
		101:	Reserved	
		11*:	Reserved	
RXDNC	Selects the RXD input noise rejection time (Time of pulses to be removed as noise)	00: 01: 10: 11:	No noise rejection 1 x (UART0DR+1)/(Transfer base clock frequency) [s] 2 x (UART0DR+1)/(Transfer base clock frequency) [s] 4 x (UART0DR+1)/(Transfer base clock frequency) [s]	
STOPBR	Receive stop bit length	0: 1:	1 bit 2 bits	

Note 1: When a read instruction is executed on UART0CR2, bits 7 and 6 are read as "0".

Note 2: RTSEL can be set to two kinds of RT clocks for the even- and odd-numbered bits of the transfer frame. For details, refer to "16.8.1 Transfer baud rate calculation method".

Note 3: For details of the RXDNC noise rejection time, refer to "16.10 Received Data Noise Rejection".

Note 4: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0CR2 remains unchanged.

Note 5: When STOPBR is set to 2 bits, the first bit of the stop bits (during data receiving) is not checked for a framing error.

Note 6: To prevent RTSEL, RXDNC and STOPBR from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "16.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed ".

UART0 baud rate register

UART0DR		7	6	5	4	3	2	1	0
(0x001C)	Bit Symbol	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

Note 1: Set UART0CR1<RXE> and UART0CR1<TXE> to "0" before changing UART0DR. For the set values, refer to "16.8 Transfer Baud Rate".

Note 2: When UART0CR1<BRG> is set to the TCA0 output, the value set to UART0DR has no meaning.

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0DR remains unchanged.

UART0 status register

UART0SR		7	6	5	4	3	2	1	0
(0x001D)	Bit Symbol	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

PERR	Parity error flag	0: 1:	No parity error Parity error
FERR	Framing error flag	0: 1:	No framing error Framing error
OERR	Overrun error flag	0: 1:	No overrrun error Overrun error
RBSY	Receive busy flag	0: 1:	Before receiving or end of receiving On receiving
RBFL	Receive buffer full flag	0: 1:	Receive buffer empty Receive buffer full
TBSY	Transmit busy flag	0: 1:	Before transmission or end of transmission On transmitting
TBFL	Transmit buffer full flag	0: 1:	Transmit buffer empty Transmit buffer full (Transmit data writing is completed)

Note 1: TBFL is cleared to "0" automatically after an INTTXD0 interrupt request is generated, and is set to "1" when data is set to TD0BUF.

Note 2: When a read instruction is executed on UART0SR, bit 4 is read as "0".

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, each bit of UART0SR is cleared to "0" and the UART stops.

UART0 receive data buffer

RD0BUF		7	6	5	4	3	2	1	0
(0x001E)	Bit Symbol	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

Note 1: When the STOP, IDLE0 or SLEEP0 mode is activated, the RD0BUF values become undefined. If received data is required, read it before activating the mode.

UART0 transmit data buffer

TD0BUF		7	6	5	4	3	2	1	0
(0x001E)	Bit Symbol	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
	Read/Write	W	W	W	W	W	W	W	W
	After reset	0	0	0	0	0	0	0	0

Note 1: When the STOP, IDLE0 or SLEEP0 mode is activated, the TD0BUF values become undefined.

16.3 Low Power Consumption Function

UART0 has a low power consumption register (POFFCR1) that saves power consumption when the UART function is not used.

Setting POFFCR1<UART0EN> to "0" disables the basic clock supply to UART0 to save power. Note that this renders the UART unusable. Setting POFFCR1<UART0EN> to "1" enables the basic clock supply to UART0 and renders the UART usable.

After reset, POFFCR1<UART0EN> is initialized to "0", and this renders the UART unusable. When using the UART for the first time, be sure to set POFFCR1<UART0EN> to "1" in the initial setting of the program (before the UART control register is operated).

Do not change POFFCR1<UART0EN> to "0" during the UART operation, otherwise UART0 may operate unexpectedly.

16.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed

The TMP89CM42 has a function that protects the registers from being changed so that the UART communication settings (for example, stop bit and parity) are not changed accidentally during the UART operation.

Specific bits of UART0CR1 and UART0CR2 can be changed only under the conditions shown in Table 16-3. If a write instruction is executed on the register when it is protected from being changed, the bits remain unchanged and keep their previous values.

Table 16-3 Changing of UART0CR1 and UART0CR2

Bit to be changed	Function	Conditions that allow the bit to be changed			
		UART0CR1 <TXE>	UART0SR <TBSY>	UART0CR1 <RXE>	UART0SR <RBSY>
UART0CR1<STOPBT>	Transmit stop bit length	Both of these bits are "0"		-	-
UART0CR1<EVEN>	Parity selection	All of these bits are "0"			
UART0CR1<PE>	Parity addition				
UART0CR1<IRDASEL>	TXD pin output selection	Both of these bits are "0"		-	-
UART0CR1<BRG>	Transfer base clock selection	All of these bits are "0"			
UART0CR2<RTSEL>	Selection of number of RT clocks				
UART0CR2<RXDNC>	Selection of RXD pin input noise rejection time	-	-	Both of these bits are "0"	
UART0CR2<STOPBR>	Receive stop bit length				

16.5 Activation of STOP, IDLE0 or SLEEP0 Mode

16.5.1 Transition of register status

When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically and each register becomes the status as shown in Table 16-4. For the registers that do not hold their values, make settings again as needed after the operation mode is recovered.

Table 16-4 Transition of Register Status

	7	6	5	4	3	2	1	0
UART0CR1	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Cleared to 0	Cleared to 0	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	-
UART0CR2	-	-	RTSEL			RXDNC		STOPBR
	-	-	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
UART0SR	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
	Cleared to 0	Cleared to 0	Cleared to 0	-	Cleared to 0	Cleared to 0	Cleared to 0	Cleared to 0
UART0DR	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
RD0BUF	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate
TD0BUF	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate

16.5.2 Transition of TXD pin status

When the IDLE0, SLEEP0 or STOP mode is activated, the TXD pin reverts to the status shown in Table 16-5, whether data is transmitted/received or the operation is stopped.

Table 16-5 TXD Pin Status When the STOP, IDLE0 or SLEEP0 Mode Is Activated

UART0CR1 <IRDASEL>	IDLE0 or SLEEP0 mode	STOP mode	
		SYSCR1<OUTEN>="1"	SYSCR1<OUTEN>="0"
"0"	H level	H level	Hi-Z
"1"	L level	L level	

16.6 Transfer Data Format

The UART transfers data composed of the following four elements. The data from the start bit to the stop bit is collectively defined as a "transfer frame". The start bit consists of 1 bit (L level) and the data consists of 8 bits. Parity bits are determined by UART0CR1<PE> that selects the presence or absence of parity and UART0CR1<EVEN> that selects even- or odd-numbered parity. The bit length of the stop bit can be selected at UART0CR1<STBT>.

Figure 16-2 shows the transfer data format.

- Start bit (1 bit)
- Data (8 bits)
- Parity bit (selectable from even-numbered, odd-numbered or no parity)
- Stop bit (selectable from 1 bit or 2 bits)

Transfer frame

PE	STBT	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Figure 16-2 Transfer Data Format

16.7 Infrared Data Format Transfer Mode

The TXD0 pin can output data in the infrared data format (IrDA) by the setting of the IrDA output control register. Setting UART0CR1<IRDASEL> to "1" allows the TXD0 pin to output data in the infrared data format.

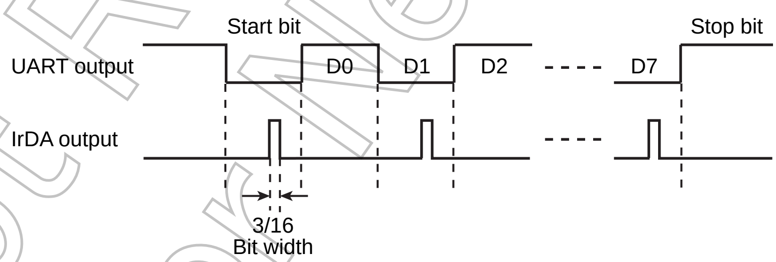


Figure 16-3 Example of Infrared Data Format (Comparison between Normal Output and IrDA Output)

16.8 Transfer Baud Rate

The transfer baud rate of UART is set by UART0CR1<BRG>, UART0DR and UART0CR2<RTSEL>. The settings of UART0DR and UART0CR2<RTSEL> for general baud rates and operating frequencies are shown below.

For independent calculation of transfer baud rates, refer to "16.8.1 Transfer baud rate calculation method".

Table 16-6 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fcgck=10 to 1 MHz, UART0CR2<RXDNC>=0y00)

Basic baud rate [baud]	Register	Operating frequency										
		10MHz	8MHz	7.3728 MHz	6.144 MHz	6MHz	5MHz	4.9152 MHz	4.19MHz	4MHz	2MHz	1MHz
128000	UART0DR	0x04	0x03	-	0x02	0x02	-	-	0x01	0x01	0x00	-
	RTSEL	0y011	0y011	-	0y000	0y011	-	-	0y001	0y011	0y011	-
	Error	(+0.81%)	(+0.81%)	-	(0%)	(+0.81%)	-	-	(-0.80%)	(+0.81%)	(+0.81%)	-
115200	UART0DR	0x04	0x03	0x03	-	0x02	-	-	-	0x01	0x00	-
	RTSEL	0y100	0y100	0y000	-	0y100	-	-	-	0y100	0y100	-
	Error	(+2.12%)	(+2.12%)	(0%)	-	(+2.12%)	-	-	-	(+2.12%)	(+2.12%)	-
76800	UART0DR	0x07	0x06	0x05	0x04	0x04	0x03	0x03	-	0x02	-	-
	RTSEL	0y001	0y010	0y000	0y000	0y011	0y001	0y000	-	0y100	-	-
	Error	(-1.36%)	(-0.79%)	(0%)	(0%)	(+0.81%)	(-1.36%)	(0%)	-	(+2.12%)	-	-
62500	UART0DR	0x09	0x07	0x06	0x05	0x05	0x04	0x04	0x03	0x03	0x01	0x00
	RTSEL	0y000	0y000	0y100	0y001	0y000	0y000	0y011	0y100	0y000	0y000	0y000
	Error	(0%)	(0%)	(-0.87%)	(-0.70%)	(0%)	(0%)	(+1.48%)	(-1.41%)	(0%)	(0%)	(0%)
57600	UART0DR	0x0A	0x08	0x07	0x06	0x06	0x04	0x04	-	0x03	0x01	0x00
	RTSEL	0y000	0y011	0y000	0y010	0y010	0y100	0y100	-	0y100	0y100	0y100
	Error	(-1.36%)	(-0.44%)	(0%)	(+1.59%)	(-0.79%)	(+2.12%)	(+0.39%)	-	(+2.12%)	(+2.12%)	(+2.12%)
38400	UART0DR	0x10	0x0C	0x0B	0x09	0x09	0x07	0x07	0x06	0x06	0x02	-
	RTSEL	0y011	0y000	0y000	0y000	0y011	0y001	0y000	0y011	0y010	0y100	-
	Error	(-1.17%)	(+0.16%)	(0%)	(0%)	(+0.81%)	(-1.36%)	(0%)	(+0.57%)	(-0.79%)	(+2.12%)	-
19200	UART0DR	0x22	0x19	0x17	0x13	0x12	0x10	0x0F	0x0D	0x0C	0x06	0x02
	RTSEL	0y010	0y000	0y000	0y000	0y001	0y011	0y000	0y011	0y000	0y010	0y100
	Error	(-0.79%)	(+0.16%)	(0%)	(0%)	(-0.32%)	(-1.17%)	(0%)	(+0.57%)	(+0.16%)	(-0.79%)	(+2.12%)
9600	UART0DR	0x40	0x30	0x2F	0x27	0x26	0x22	0x1F	0x1C	0x19	0x0C	0x06
	RTSEL	0y000	0y100	0y000	0y000	0y000	0y010	0y000	0y010	0y000	0y000	0y010
	Error	(+0.16%)	(+0.04%)	(0%)	(0%)	(+0.16%)	(-0.79%)	(0%)	(+0.34%)	(+0.16%)	(+0.16%)	(-0.79%)
4800	UART0DR	0x8A	0x64	0x5F	0x4F	0x4D	0x40	0x3F	0x34	0x30	0x19	0x0C
	RTSEL	0y010	0y001	0y000	0y000	0y000	0y000	0y000	0y001	0y100	0y000	0y000
	Error	(-0.08%)	(+0.01%)	(0%)	(0%)	(+0.16%)	(+0.16%)	(0%)	(-0.18%)	(+0.04%)	(+0.16%)	(+0.16%)
2400	UART0DR	0xF4	0xC9	0xBF	0x9F	0x92	0x8A	0x7F	0x6C	0x64	0x30	0x19
	RTSEL	0y100	0y001	0y000	0y000	0y100	0y010	0y000	0y000	0y001	0y100	0y000
	Error	(+0.04%)	(+0.01%)	(0%)	(0%)	(+0.04%)	(-0.08%)	(0%)	(+0.11%)	(+0.01%)	(+0.04%)	(+0.16%)
1200	UART0DR	-	-	-	-	-	0xF4	0xFF	0xE8	0xC9	0x64	0x30
	RTSEL	-	-	-	-	-	0y100	0y000	0y010	0y001	0y001	0y100
	Error	-	-	-	-	-	(+0.04%)	(+0%)	(-0.10%)	(+0.01%)	(+0.01%)	(+0.04%)

Table 16-7 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fs=32.768 kHz, UART0CR2<RXDNC>=0y00)

Basic baud rate [baud]	Register	Operating frequency
		32.768 kHz
300	UART0DR	0x06
	RTSEL	0y011
	Error	(+0.67%)
150	UART0DR	0x0D
	RTSEL	0y011
	Error	(+0.67%)
134	UART0DR	0x0E
	RTSEL	0y001
	Error	(-1.20%)
110	UART0DR	0x11
	RTSEL	0y001
	Error	(+0.30%)
75	UART0DR	0x1C
	RTSEL	0y010
	Error	(+0.44%)

Note 1: The overall error from the basic baud rate must be within ±3%. Even if the overall error is within ±3%, the communication may fail due to factors such as frequency errors in external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.

16.8.1 Transfer baud rate calculation method

16.8.1.1 Bit width adjustment using UART0CR2<RTSEL>

The bit width of transmitted/received data can be finely adjusted by changing UART0CR2<RTSEL>. The number of RT clocks per bit can be changed in a range of 15 to 17 clocks by changing UART0CR2<RTSEL>. The RT clock is the transfer base clock, which is the pulses obtained by counting the clock selected at UART0CR1<BRG> the number of times of (UART0DR set value) + 1. Especially, when UART0CR2<RTSEL> is set to "0y001" or "0y011", two types of RT clocks alternate at each bit, so that the pseudo baud rates of RT × 15.5 clocks and RT × 16.5 clocks can be generated. The number of RT clocks per bit of transfer frame is shown in Figure 16-4.

For example, when fcgck is 4 [MHz], UART0CR2<RTSEL> is set to "0y000" and UART0DR is set to "0x19", the baud rate calculated using the formula in Figure 16-4 is expressed as:

fcgck / (16 × (UART0DR + 1) = 9615 [baud]

These settings generate a baud rate close to 9600 [baud] (+0.16%).

Transfer frame

PE	STBT	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

RTSEL	Number of RT clocks												Generated baud rate
000	16	16	16	16	16	16	16	16	16	16	16	16	$\frac{fcgck}{16 \times (UARTDR+1)}$ [baud]
001	16	17	16	17	16	17	16	17	16	17	16	17	$\frac{fcgck}{16.5 \times (UARTDR+1)}$ [baud]
010	15	15	15	15	15	15	15	15	15	15	15	15	$\frac{fcgck}{15 \times (UARTDR+1)}$ [baud]
011	15	16	15	16	15	16	15	16	15	16	15	16	$\frac{fcgck}{15.5 \times (UARTDR+1)}$ [baud]
100	17	17	17	17	17	17	17	17	17	17	17	17	$\frac{fcgck}{17 \times (UARTDR+1)}$ [baud]

*When BRG is set to fcgck

Figure 16-4 Fine Adjustment of Baud Rate Clock Using UART0CR2<RTSEL>

16.8.1.2 Calculation of set values of UART0CR2<RTSEL> and UART0DR

The set value of UART0DR for an operating frequency and baud rate can be calculated using the calculation formula shown in Figure 16-5. For example, to generate a basic baud rate of 38400 [baud] with fcgck=4 [MHz], calculate the set value of UART0DR for each setting of UART0CR2<RTSEL> and compensate the calculated value to a positive number to obtain the generated baud rate as shown in Figure 16-6. Basically, select the set value of UART0CR2<RTSEL> that has the smallest baud rate error from among the generated baud rates. In Figure 16-6, the setting of UART0CR2<RTSEL>="0y010" has the smallest error among the calculated baud rates, and thus the generated baud rate is 38095 [baud] (−0.79%) against the basic baud rate of 38400 [baud].

Note: The error from the basic baud rate should be accurate to within ±3%. Even if the error is within ±3%, the communication may fail due to factors such as frequency errors of external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.

RTSEL	UARTDR set value
000	$UARTDR = \frac{fcgck\ [Hz]}{16 \times A\ [baud]} - 1$
001	$UARTDR = \frac{fcgck\ [Hz]}{16.5 \times A\ [baud]} - 1$
010	$UARTDR = \frac{fcgck\ [Hz]}{15 \times A\ [baud]} - 1$
011	$UARTDR = \frac{fcgck\ [Hz]}{15.5 \times A\ [baud]} - 1$
100	$UARTDR = \frac{fcgck\ [Hz]}{17 \times A\ [baud]} - 1$

Figure 16-5 UART0DR Calculation Method (When BRG Is Set to fcgck)

RTSEL	UARTDR calculation	Generated baud rate
000	UARTDR = $\frac{4000000\ [Hz]}{16 \times 38400\ [baud]} - 1 \approx 6$	$\frac{4000000\ [Hz]}{16 \times (6 + 1)} = 35714$ [baud] (−6.99%)
001	UARTDR = $\frac{4000000\ [Hz]}{16.5 \times 38400\ [baud]} - 1 \approx 5$	$\frac{4000000\ [Hz]}{16.5 \times (5 + 1)} = 40404$ [baud] (+5.22%)
010	UARTDR = $\frac{4000000\ [Hz]}{15 \times 38400\ [baud]} - 1 \approx 6$	$\frac{4000000\ [Hz]}{15 \times (6 + 1)} = 38095$ [baud] (−0.79%)
011	UARTDR = $\frac{4000000\ [Hz]}{15.5 \times 38400\ [baud]} - 1 \approx 6$	$\frac{4000000\ [Hz]}{15.5 \times (6 + 1)} = 36866$ [baud] (−3.99%)
100	UARTDR = $\frac{4000000\ [Hz]}{17 \times 38400\ [baud]} - 1 \approx 5$	$\frac{4000000\ [Hz]}{17 \times (5 + 1)} = 39216$ [baud] (+2.12%)

Figure 16-6 Example of UART0DR Calculation

16.9 Data Sampling Method

The UART receive control circuit starts RT clock counting when it detects a falling edge of the input pulses to the RXD0 pin. 15 to 17 RT clocks are counted per bit and each clock is expressed as RTn (n=16 to 0). In a bit that has 17 RT clocks, RT16 to RT0 are counted. In a bit that has 16 RT clocks, RT15 to RT0 are counted. In a bit that has 15 RT clocks, RT14 to RT0 are counted (Decrement). During counting of RT8 to RT6, the UART receive control circuit samples the input pulses to the RXD0 pin to make a majority decision. The same level detected twice or more from among three samplings is processed as the data for the bit.

The number of RT clocks can be changed in a range of 15 to 17 by setting UART0CR2<RTSEL>. However, sampling is always executed in RT8 to RT6, even if the number of RT clocks is changed (Figure 16-7).

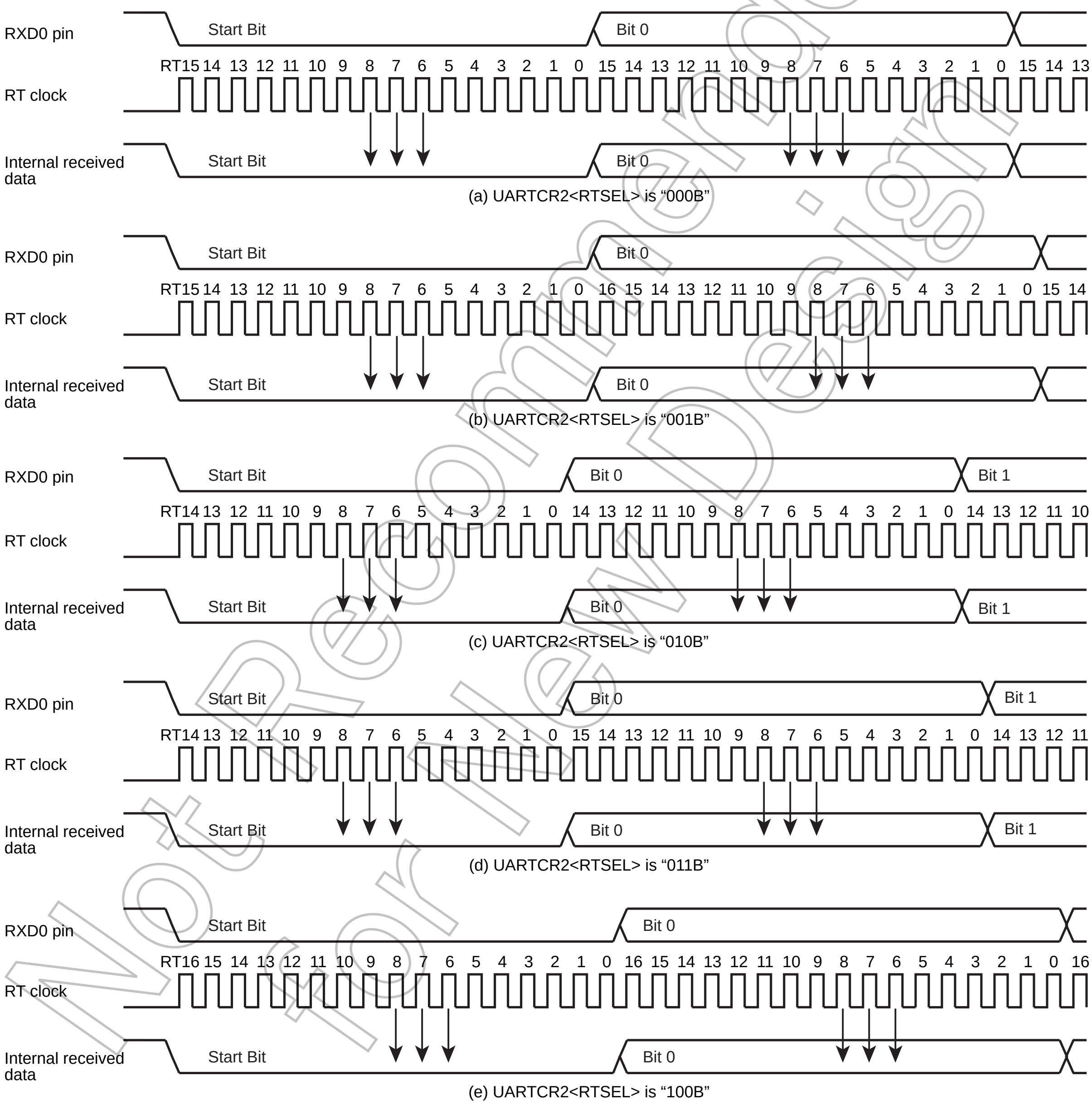


Figure 16-7 Data Sampling in Each Case of UARTCR2<RTSEL>

If "1" is detected in sampling of the start bit, for example, due to the influence of noise, RT clock counting stops and the data receiving is suspended. Subsequently, when a falling edge is detected in the input pulses to the RXD0 pin, RT clock counting restarts and the data receiving restarts with the start bit.

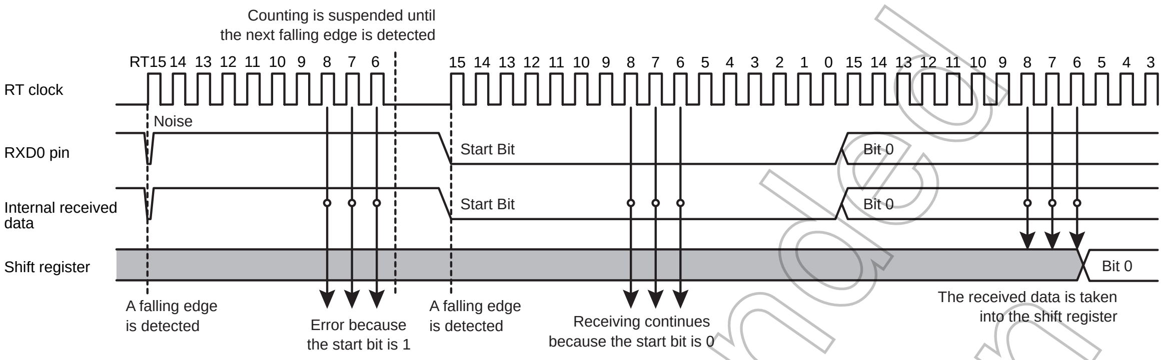


Figure 16-8 Start Bit Sampling

16.10 Received Data Noise Rejection

When noise rejection is enabled at UART0CR2<RXDNC>, the time of pulses to be regarded as signals is as shown in Table 16-8.

Table 16-8 Received Data Noise Rejection Time

RXDNC	Noise rejection time [s]	Time of pulses to be regarded as signals
00	No noise rejection	-
01	(UART0DR+1)/(Transfer base clock frequency)	2 × (UART0DR+1)/(Transfer base clock frequency)
10	2 × (UART0DR+1)/(Transfer base clock frequency)	4 × (UART0DR+1)/(Transfer base clock frequency)
11	4 × (UART0DR+1)/(Transfer base clock frequency)	8 × (UART0DR+1)/(Transfer base clock frequency)

Note 1: The transfer base clock frequency is the clock frequency selected at UARTCR1<BRG>.

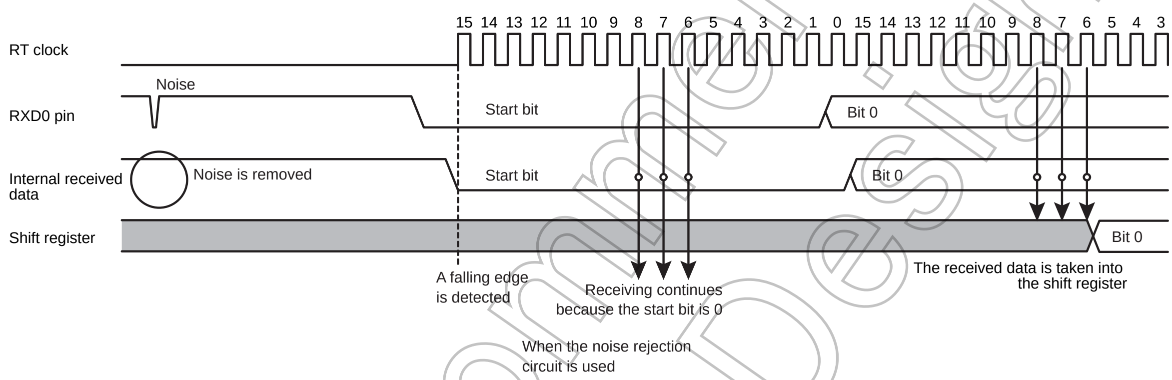


Figure 16-9 Received Data Noise Rejection

16.11 Transmit/Receive Operation

16.11.1 Data transmit operation

Set UART0CR1<TXE> to "1". Check UART0SR<TBFL> = "0", and then write data into TD0BUF (transmit data buffer). Writing data into TD0BUF sets UART0SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD0 pin. The data output includes a start bit, stop bits whose number is specified in UART0CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR. When data transmission starts, the transmit buffer full flag UART0SR<TBFL> is cleared to "0" and an INTTXD0 interrupt request is generated.

Note 1: After data is written into TD0BUF, if new data is written into TD0BUF before the previous data is transferred to the shift register, the new data is written over the previous data and is transferred to the shift register.

Note 2: Under the conditions shown in Table 16-9, the TXD0 pin output is fixed at the L or H level according to the setting of UART0CR1<IRDASEL>.

Table 16-9 TXD0 Pin Output

<table>
<tr><th rowspan="2">Condition</th><th colspan="2">TXD0 pin output</th></tr>
<tr><th>IRDASEL="0"</th><th>IRDASEL="1"</th></tr>
<tr><td>When UART0CR1<TXE> is "0"</td><td rowspan="3">H level</td><td rowspan="3">L level</td></tr>
<tr><td>From when "1" is written to UART0CR1<TXE> to when the transmitted data is written to TD0BUF</td></tr>
<tr><td>When the STOP, IDLE0 or SLEEP0 mode is active</td></tr>
</table>

16.11.2 Data receive operation

Set UART0CR1<RXE> to "1". When data is received via the RXD0 pin, the received data is transferred to RD0BUF (receive data buffer). At this time, the transmitted data includes a start bit, stop bit(s) and a parity bit if parity addition is specified. When the stop bit(s) are received, data only is extracted and transferred to RD0BUF (receive data buffer). Then the receive buffer full flag UART0SR<RBFL> is set and an INTRXD0 interrupt request is generated. Set the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR.

If an overrun error occurs when data is received, the data is not transferred to RD0BUF (receive data buffer) but discarded; data in the RD0BUF is not affected.

16.12 Status Flag

16.12.1 Parity error

When the parity determined using the receive data bits differs from the received parity bit, the parity error flag UART0SR<PERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR<PERR> is "1" when UART0SR is read, UART0SR<PERR> will be cleared to "0" when RD0BUF is read subsequently. (The RD0BUF read value becomes undefined.)

If UART0SR<PERR> is set to "1" after UART0SR is read, UART0SR<PERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<PERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

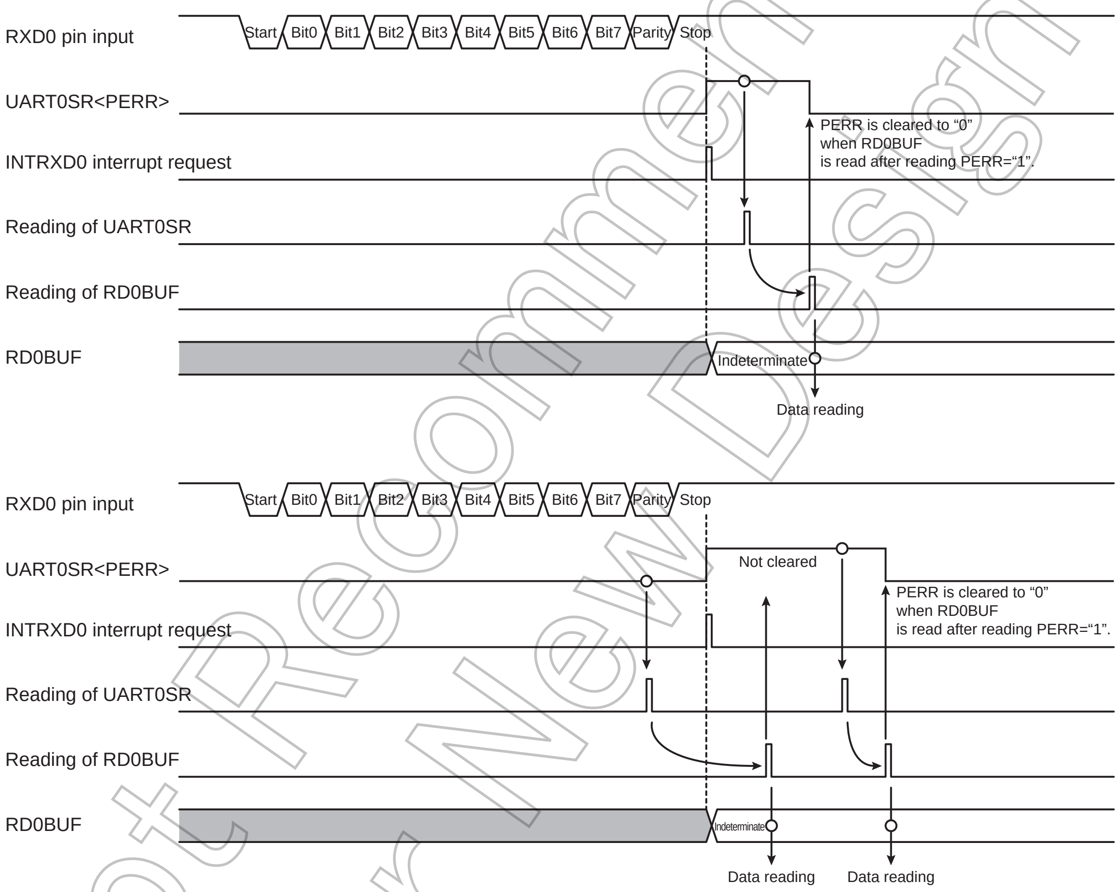


Figure 16-10 Occurrence of Parity Error

16.12.2 Framing Error

If the internal and external baud rates differ or "0" is sampled as the stop bit of received data due to the influence of noise on the RXD0 pin, the framing error flag UART0SR<FERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR<FERR> is "1" when UART0SR is read, UART0SR<FERR> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR<FERR> is set to "1" after UART0SR is read, UART0SR<FERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<FERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

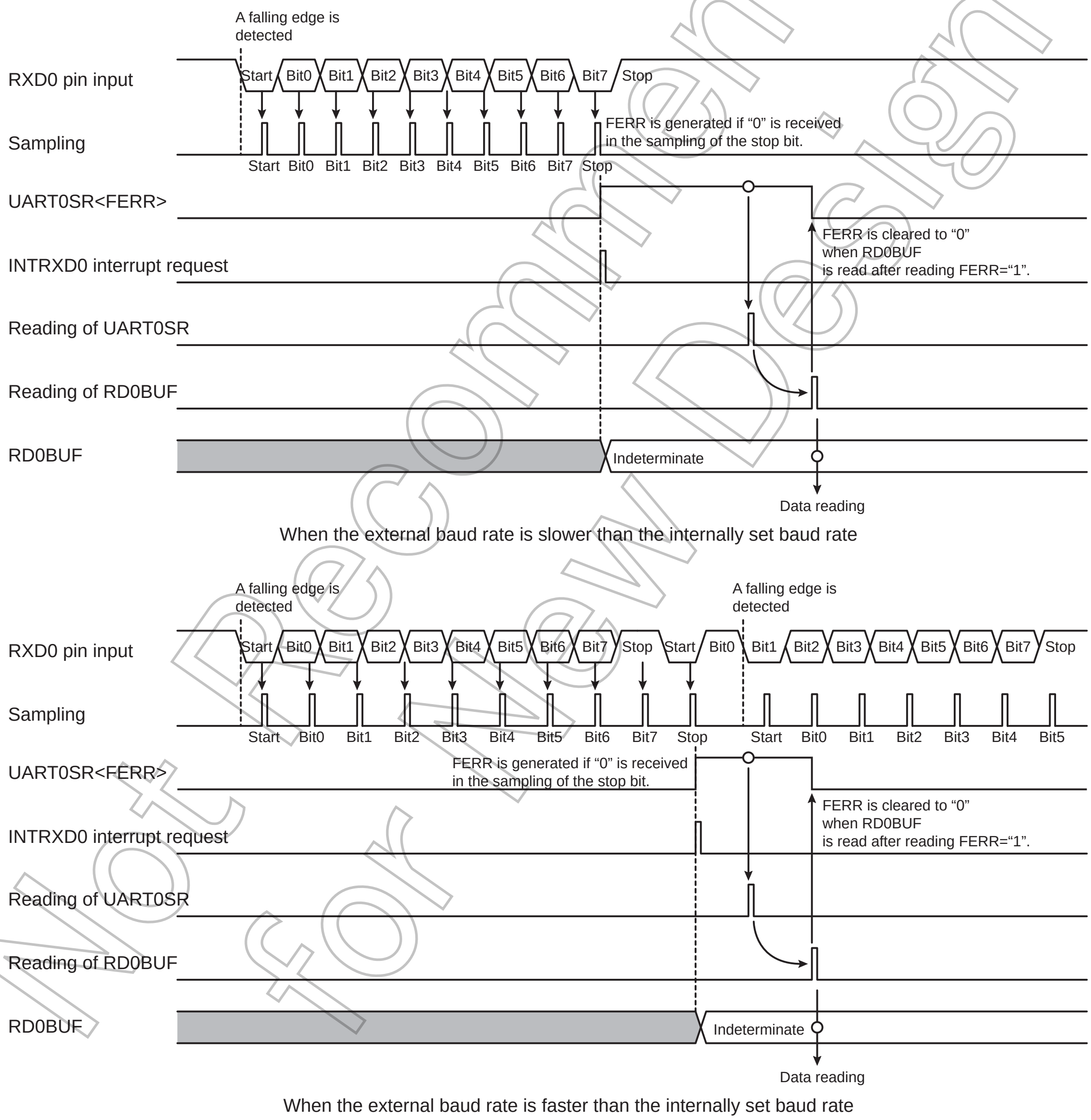


Figure 16-11 Occurrence of Framing Error

16.12.3 Overrun error

If receiving of all data bits is completed before the previous received data is read from RD0BUF, the overrun error flag UART0SR<OERR> is set to "1" and an INTRXD0 interrupt request is generated. The data received at the occurrence of the overrun error is discarded and the previous received data is maintained. Subsequently, if data is received while UART0SR<OERR> is still "1", no INTRXD0 interrupt request is generated, and the received data is discarded. (Figure 16-12)

Note that parity or framing errors in the discarded received data cannot be detected. (These error flags are not set.) That is to say, if these errors are detected together with an overrun error during the reading of UART0SR, they have occurred in the previous received data (the data stored in RD0BUF). (Figure 16-13)

If UART0SR<OERR> is "1" when UART0SR is read, UART0SR<OERR> will be cleared to "0" when RD0BUF is read subsequently. (Figure 16-14)

If UART0SR<OERR> is set to "1" after UART0SR is read, UART0SR<OERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<OERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read. (Figure 16-14)

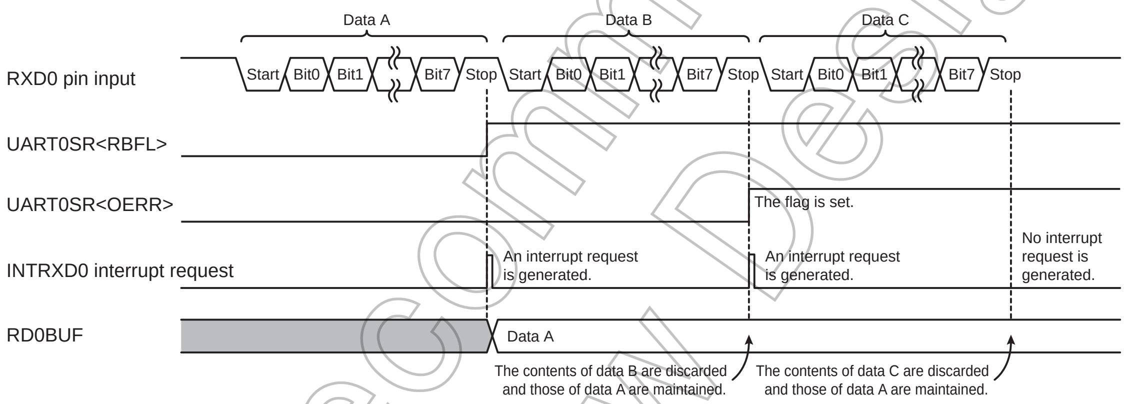


Figure 16-12 Generation of INTRXD0 Interrupt Request

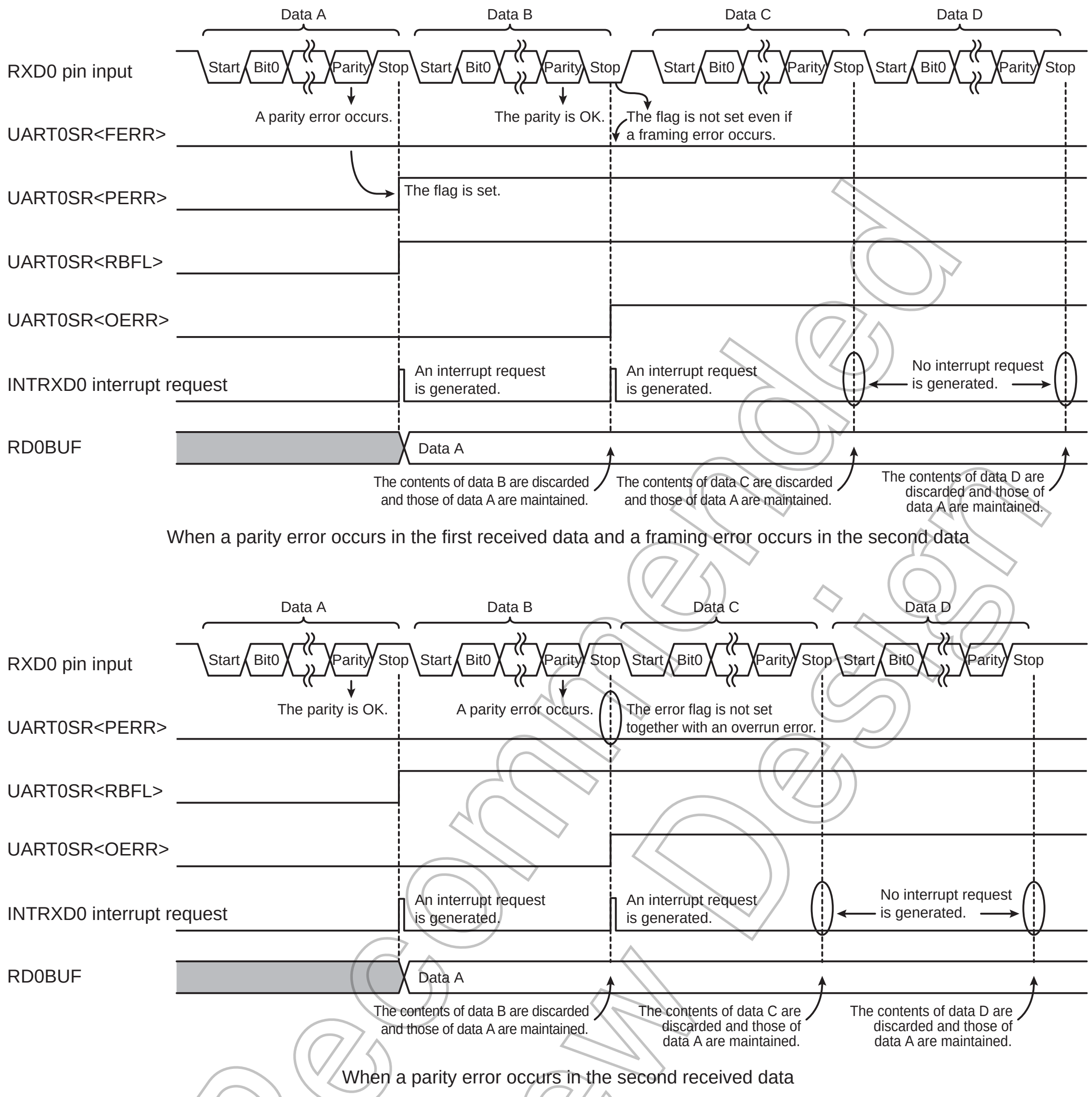
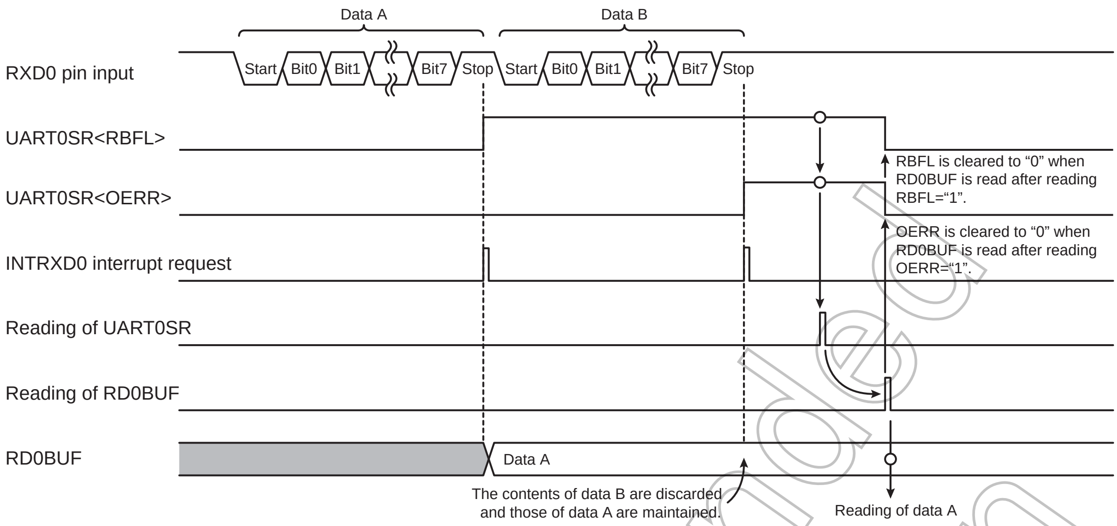


Figure 16-13 Framing/Parity Error Flags When an Overrun Error Occurs



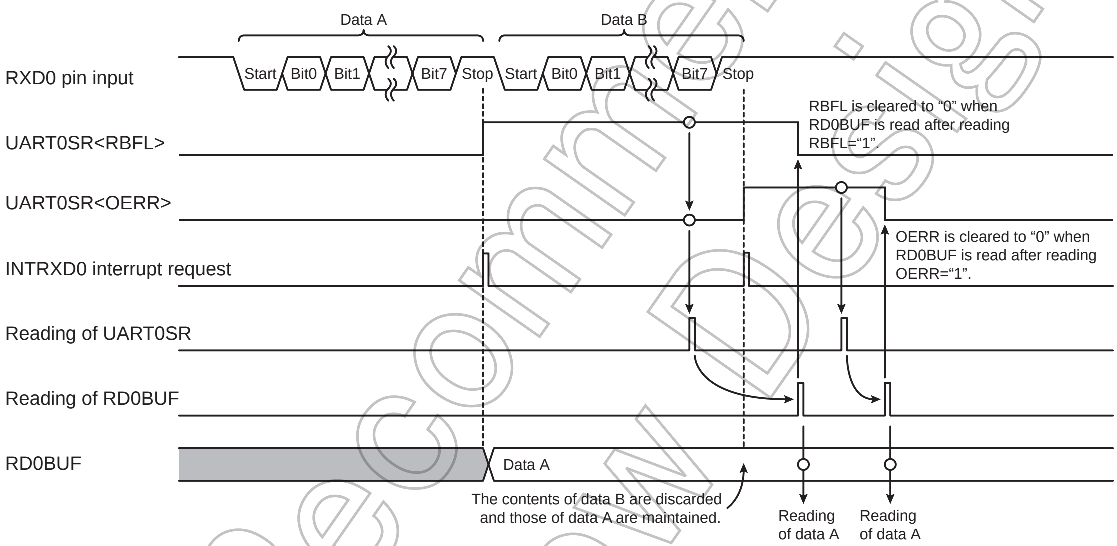


Figure 16-14 Clearance of Overrun Error Flag

16.12.4 Receive Data Buffer Full

Loading the received data in RD0BUF sets UART0SR<RBFL> to "1".

If UART0SR<RBFL> is "1" when UART0SR is read, UART0SR<RBFL> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR<RBFL> is set to "1" after UART0SR is read, UART0SR<RBFL> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<RBFL> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

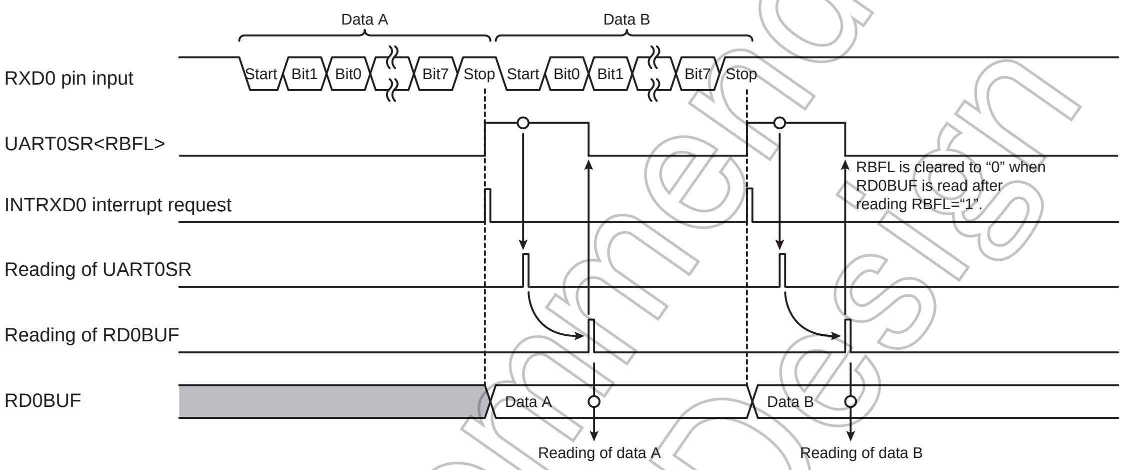


Figure 16-15 Occurrence of Receive Data Buffer Full

16.12.5 Transmit busy flag

If transmission is completed with no waiting data in TD0BUF (when UART0SR<TBFL>="0"), UART0SR<TBSY> is cleared to "0". When transmission is restarted after data is written into TD0BUF, UART0SR<TBSY> is set to "1". At this time, an INTTXD0 interrupt request is generated.

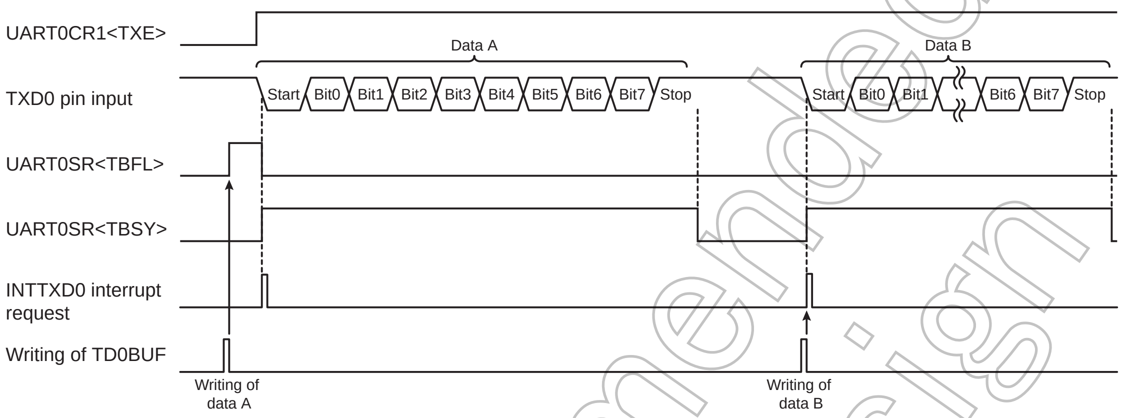


Figure 16-16 Transmit Busy Flag and Occurrence of Transmit Buffer Full

16.12.6 Transmit Buffer Full

When TD0BUF has no data, or when data in TD0BUF is transferred to the transmit shift register and transmission is started, UART0SR<TBFL> is cleared to "0". At this time, an INTTXD0 interrupt request is generated.

Writing data into TD0BUF sets UART0SR<TBFL> to "1".

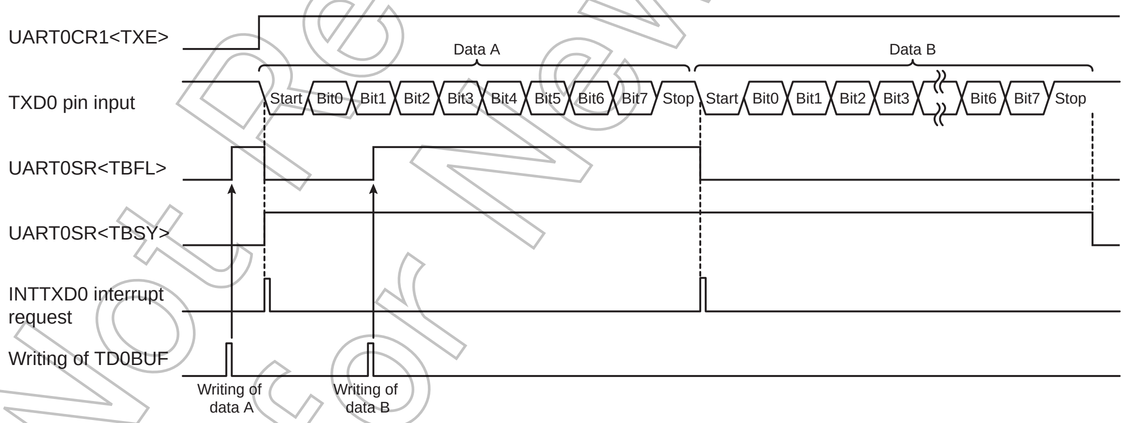


Figure 16-17 Occurrence of Transmit Buffer Full

16.13 Receiving Process

Figure 16-18 shows an example of the receiving process. Details of flag judgments in the processing are shown in Table 16-10 and Table 16-11.

If any framing error or parity error is detected, the received data has erroneous value(s). Execute the error handling, for example, by discarding the received data read from RD0BUF and receiving the data again.

If any overrun error is detected, the receiving of one or more pieces of data is unfinished. It is impossible to determine the number of pieces of data that could not be received. Execute the error handling, for example, by receiving data again from the beginning of the transfer. Basically, an overrun error occurs when the internal software processing cannot follow the data transfer speed. It is recommended to slow the transfer baud rate or modify the software to execute flow control.

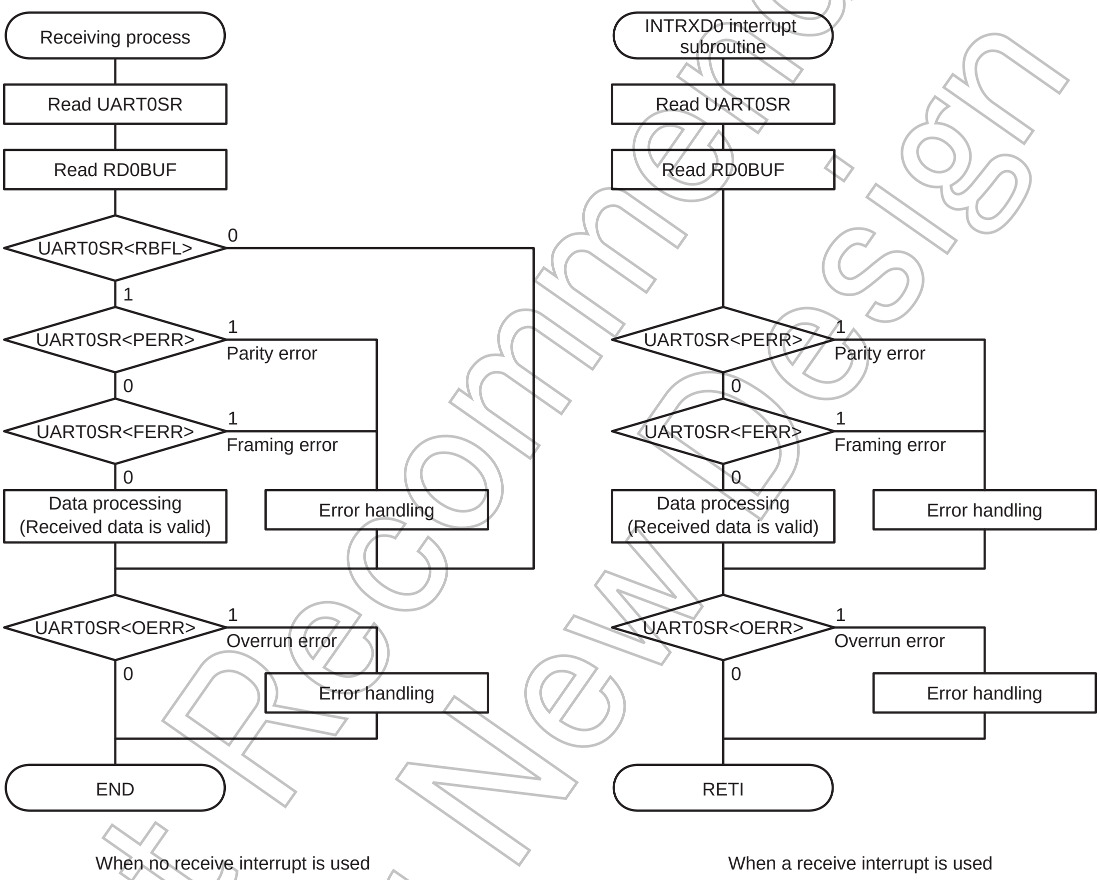


Figure 16-18 Example of Receiving Process

Note 1: If multiple interrupts are used in the INTRXD0 interrupt subroutine, the interrupt should be enabled after reading UART0SR and RD0BUF.

Table 16-10 Flag Judgments When No Receive Interrupt Is Used

RBFL	FERR/PERR	OERR	State
0	-	0	Data has not been received yet.
0	-	1	Some pieces of data could not be received during the previous data receiving process (Receiving of next data is completed in the period from when UART0SR is read to when RD0BUF is read in the previous data receiving process.)
1	0	0	Receiving has been completed properly.
1	0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	1	0	Received data has erroneous value(s).
1	1	1	Received data has erroneous value(s) and some pieces of data could not be received.

Table 16-11 Flag Judgments When a Receive Interrupt Is Used

FERR/PERR	OERR	State
0	0	Receiving has been completed properly.
0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	0	Received data has erroneous value(s).
1	1	Received data has erroneous value(s) and some pieces of data could not be received.

16.14 AC Properties

16.14.1 IrDA properties

(V_{SS} = 0 V, Topr = −40 to 85°C)

Item	Condition	Min	Typ.	Max	Unit
TXD output pulse time (RT clock × (3/16))	Transfer baud rate = 2400 bps	-	78.13	-	μs
	Transfer baud rate = 9600 bps	-	19.53	-	
	Transfer baud rate = 19200 bps	-	9.77	-	
	Transfer baud rate = 38400 bps	-	4.88	-	
	Transfer baud rate = 57600 bps	-	3.26	-	
	Transfer baud rate = 115200 bps	-	1.63	-	

16.15 Revision History

Rev	Description
RA001	Revised Table 16-6.
	"16.8.1.1 Bit width adjustment using UART0CR2<RTSEL>" Changed example from fcgck=8MHz to fcgck=4MHz. "16.8.1.2 Calculation of set values of UART0CR2<RTSEL> and UART0DR" Changed example from fcgck=6MHz to fcgck=4MHz. "Figure 16-6 Example of UART0DR Calculation" Changed example from fcgck=6MHz to fcgck=4MHz.
	"Figure 16-1 Asynchronous Serial Interface (UART)" Added $\overline{\text{PPGA0}}$ output to TCA0 output.

17. Synchronous Serial Interface (SIO)

The TMP89CM42 contains 1 channel of high-speed 8-bit serial interfaces of the clock synchronization type.

Table 17-1 SFR Address Assignment

	SIOxCR (address)	SIOxSR (address)	SIOxBUF (address)
Serial interface 0	SIO0CR (0x001F)	SIO0SR (0x0020)	SIO0BUF (0x0021)

Table 17-2 Pin Names

	Serial clock input/output pin	Serial data input pin	Serial data output pin
Serial interface 0	SCLK0 pin	SI0 pin	SO0 pin

17.1 Configuration

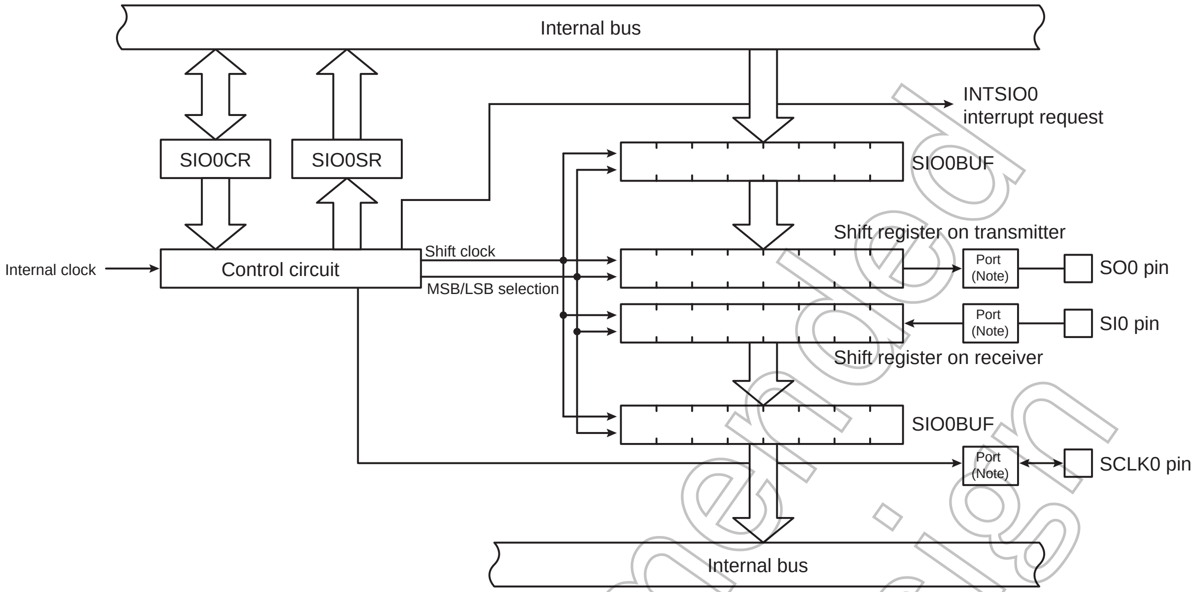


Figure 17-1 Serial Interface

Note: The serial interface input/output pins are also used as the I/O ports. The I/O port register settings are required to use these pins for a serial interface. For details, refer to the chapter of I/O ports.

17.2 Control

The synchronous serial interface SIO0 is controlled by the low power consumption registers (POFFCR2), the serial interface data buffer register (SIO0BUF), the serial interface control register (SIO0CR) and the serial interface status register (SIO0SR).

Low power consumption register 2

POFFCR2 (0x0F76)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0	Disable
		1	Enable
SIO0EN	SIO0 control	0	Disable
		1	Enable

Serial interface buffer register

SIO0BUF (0x0021)		7	6	5	4	3	2	1	0
	Bit Symbol	SIO0BUF							
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Serial interface buffer register

SIO0BUF (0x0021)		7	6	5	4	3	2	1	0
	Bit Symbol	SIO0BUF							
	Read/Write	W							
	After reset	1	1	1	1	1	1	1	1

Note 1: SIO0BUF is the data buffer for both transmission and reception. The last received data is read each time SIO0BUF is read. If SIO0BUF has never received data, it is read as "0". When data is written into it, the data is treated as the transmit data.

Serial interface control register

SIO0CR (0x001F)		7	6	5	4	3	2	1	0
	Bit Symbol	SIOEDG	SIOCKS			SIODIR	SIOS	SIOM	
	Read/Write	R/W	R/W			R/W	R/W	R/W	
	After reset	0	0	0	0	0	0	0	0

SIOEDG	Transfer edge selection	0 1	0: Receive data at a rising edge and transmit data at a falling edge 1: Transmit data at a rising edge and receive data at a falling edge	
SIOCKS	Serial clock selection [Hz]		NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode
		000	fcgck/2^9	-
		001	fcgck/2^6	-
		010	fcgck/2^5	-
		011	fcgck/2^4	-
		100	fcgck/2^3	-
		101	fcgck/2^2	-
		110	fcgck/2	fs/2^3
		111	External clock input	
SIODIR	Transfer format (MSB/LSB) selection	0 1	LSB first (transfer from bit 0) MSB first (transfer from bit 7)	
SIOS	Transfer operation start/stop instruction	0 1	0: Operation stop (reserved stop) 1: Operation start	
SIOM	Transfer mode selection and operation	00 01 10 11	Operation stop (forced stop) 8-bit transmit mode 8-bit receive mode 8-bit transmit and receive mode	

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: After the operation is started by writing "1" to SIOS, writing to SIOEDG, SIOCKS and SIODIR is invalid until SIO0SR<SIOF> becomes "0". (SIOEDG, SIOCKS and SIODIR can be changed at the same time as changing SIOS from "0" to "1".)

Note 3: After the operation is started by writing "1" to SIOS, no values other than"00" can be written to SIOM until SIOF becomes "0" (if a value from "01" to "11" is written to SIOM, it is ignored). The transfer mode cannot be changed during the operation.

Note 4: SIOS remains at "0", if "1" is written to SIOS when SIOM is "00" (operation stop).

Note 5: When SIO is used in SLOW1/2 or SLEEP1 mode, be sure to set SIOCKS to "110". If SIOCKS is set to any other value, SIO will not operate. When SIO is used in SLOW1/2 or SLEEP1 mode, execute communications with SIOCKS="110" in advance or change SIOCKS after SIO is stopped.

Note 6: When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared to "00" and SIO stops the operation. At the same time, SIOS is cleared to "0". However, the values set for SIOEDG, SIOCKS and SIODIR are maintained.

Serial interface status register

SIO0SR		7	6	5	4	3	2	1	0
(0x0020)	Bit Symbol	SIOF	SEF	OERR	REND	UERR	TBFL	-	-
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

SIOF	Serial transfer operation status monitor	0	Transfer not in progress
		1	Transfer in progress
SEF	Shift operation status monitor	0	Shift operation not in progress
		1	Shift operation in progress
OERR	Receive overrun error flag	0	No overrun error has occurred
		1	At least one overrun error has occurred
REND	Receive completion flag	0	No data has been received since the last receive data was read out
		1	At least one data receive operation has been executed
UERR	Transmit underrun error flag	0	No transmit underrun error has occurred
		1	At least one transmit underrun error has occurred
TBFL	Transmit buffer full flag	0	The transmit buffer is empty
		1	The transmit buffer has the data that has not yet been transmitted

Note 1: The OERR and UERR flags are cleared by reading SIO0SR.

Note 2: The REND flag is cleared by reading SIO0BUF.

Note 3: Writing "00" to SIO0CR<SIOM> clears all the bits of SIO0SR to "0", whether the serial interface is operating or not. When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared to "00" and all the bits of SIO0SR are cleared to "0".

Note 4: Bit 1 to 0 of SIO0SR are read "0".

17.3 Low Power Consumption Function

Serial interface 0 has the low power consumption registers (POFFCR2) that save power when the serial interface is not being used.

Setting POFFCR2<SIO0EN> to "0" disables the basic clock supply to serial interface 0 to save power. Note that this renders the serial interface unusable. Setting POFFCR2<SIO0EN> to "1" enables the basic clock supply to serial interface 0 and allows the serial interface to operate.

After reset, POFFCR2<SIO0EN> are initialized to "0", and this renders the serial interface unusable. When using the serial interface for the first time, be sure to set POFFCR2<SIO0EN> to "1" in the initial setting of the program (before the serial interface control registers are operated).

Do not change POFFCR2<SIO0EN> to "0" during the serial interface operation. Otherwise serial interface 0 may operate unexpectedly.

17.4 Functions

17.4.1 Transfer format

The transfer format can be set to either MSB or LSB first by using SIO0CR<SIODIR>. Setting SIO0CR<SIODIR> to "0" selects LSB first as the transfer format. In this case, the serial data is transferred in sequence from the least significant bit.

Setting SIO0CR<SIODIR> to "1" selects MSB first as the transfer format. In this case, the serial data is transferred in sequence from the most significant bit.

17.4.2 Serial clock

The serial clock can be selected by using SIO0CR<SIOCKS>.

Setting SIO0CR<SIOCKS> to "000" to "110" selects the internal clock as the serial clock. In this case, the serial clock is output from the SCLK0 pin. The serial data is transferred in synchronization with the edge of the SCLK0 pin output.

Setting SIO0CR<SIOCKS> to "111" selects an external clock as the serial clock. In this case, an external serial clock must be input to the SCLK0 pin. The serial data is transferred in synchronization with the edge of the external clock.

The serial data transfer edge can be selected for both the external and internal clocks. For details, refer to "17.4.3 Transfer edge selection".

Table 17-3 Transfer Baud Rate

SIO0CR <SIOCKS>	Serial clock [Hz]		fcgck=4MHz		fcgck=8MHz		fcgck=10MHz		fs=32.768kHz	
	NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)
000	$fcgck/2^9$	-	128	7.813k	64	15.625k	51.2	19.531k	-	-
001	$fcgck/2^6$	-	16	62.5k	8	125k	6.4	156.25k	-	-
010	$fcgck/2^5$	-	8	125k	4	250k	3.2	312.5k	-	-
011	$fcgck/2^4$	-	4	250k	2	500k	1.6	625k	-	-
100	$fcgck/2^3$	-	2	500k	1	1M	0.8	1.25M	-	-
101	$fcgck/2^2$	-	1	1M	0.5	2M	0.4	2.5M	-	-
110	fcgck/2	$fs/2^3$	0.5	2M	0.25	4M	0.2	5M	244	4k

17.4.3 Transfer edge selection

The serial data transfer edge can be selected by using SIOCR<SIOEDG>.

Table 17-4 Transfer Edge Selection

SIO0CR<SIOEDG>	Data transmission	Data reception
0	Falling edge	Rising edge
1	Rising edge	Falling edge

When SIOCR<SIOEDG> is "0", the data is transmitted in synchronization with the falling edge of the clock and the data is received in synchronization with the rising edge of the clock.

When SIOCR<SIOEDG> is "1", the data is transmitted in synchronization with the rising edge of the clock and the data is received in synchronization with the falling edge of the clock.

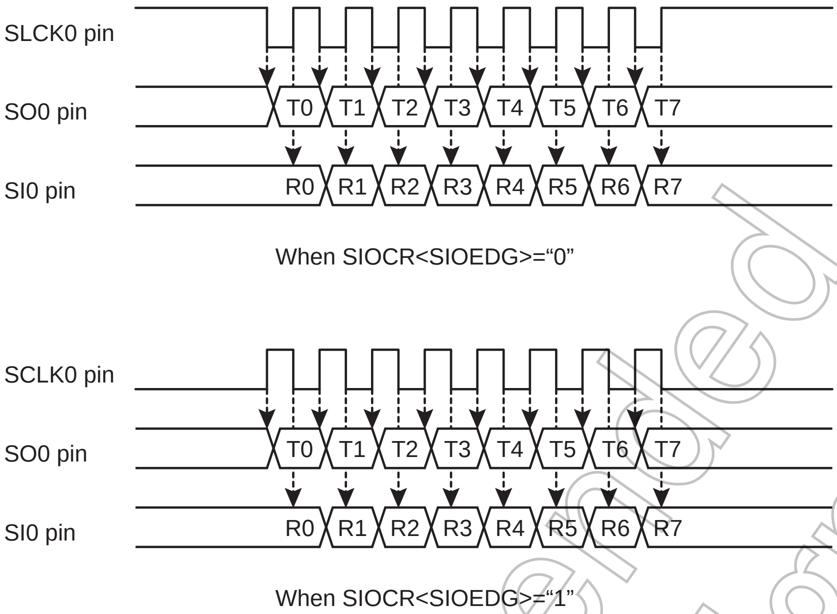
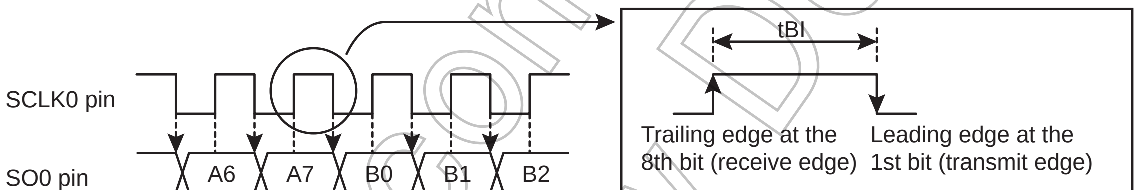


Figure 17-2 Transfer Edge

Note: When an external clock input is used, 4/fcgck or longer is needed between the receive edge at the 8th bit and the transfer edge at the first bit of the next transfer.



Symbol	Name	Minimum time
tBI	Interval time between bytes	4/fcgck

Figure 17-3 Interval time between bytes

17.5 Transfer Modes

17.5.1 8-bit transmit mode

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

17.5.1.1 Setting

Before starting the transmit operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

The transmit operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

17.5.1.2 Starting the transmit operation

The transmit operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and then transmitted as the serial data from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>. The serial data becomes undefined if the transmit operation is started without writing any transmit data to SIO0BUF.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of the serial data is output.

17.5.1.3 Transmit buffer and shift operation

If data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

If data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

17.5.1.4 Operation on completion of transmission

The operation on completion of the data transmission varies depending on the operating clock and the state of SIO0SR<TBFL>.

(1) When the internal clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the internal clock is used, the serial clock and data output is stopped until the next transmit data is written into SIO0BUF (automatic wait).

When the subsequent data is written into SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the serial clock, and the transmit operation is restarted. An INTSIO0 interrupt request is generated at the restart of the transmit operation.

(2) When an external clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SO pin keeps last output value. When an external serial clock is input to the SCLK0 pin after completion of the data transmission, an undefined value is transmitted and the transmit underrun error flag SIO0SR<UERR> is set to "1".

If a transmit underrun error occurs, data must not be written to SIO0BUF during the transmission of an undefined value. (It is recommended to finish the transmit operation by setting SIO0CR<SIOS> to "0" or force the transmit operation to stop by setting SIO0CR<SIOM> to "00".)

The transmit underrun error flag SIO0SR<UERR> is cleared by reading SIO0SR.

(3) When an internal or external clock is used and SIO0SR<TBFL> is "1"

When the data transmission is completed, SIO0SR<TBFL> is cleared to "0". The data in SIO0BUF is transferred to the shift register and the transmission of subsequent data is started. At this time, SIO0SR<SEF> is set to "1" and an INTSIO0 interrupt request is generated.

17.5.1.5 Stopping the transmit operation

Set SIO0CR<SIOS> to "0" to stop the transmit operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the transmit operation is stopped immediately and an INTSIO0 interrupt request is generated. When SIO0SR<SEF> is "1", the transmit operation is stopped after all the data in the shift register is transmitted (reserved stop). At this time, an INTSIO0 interrupt request is generated again.

When the transmit operation is completed, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "H" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

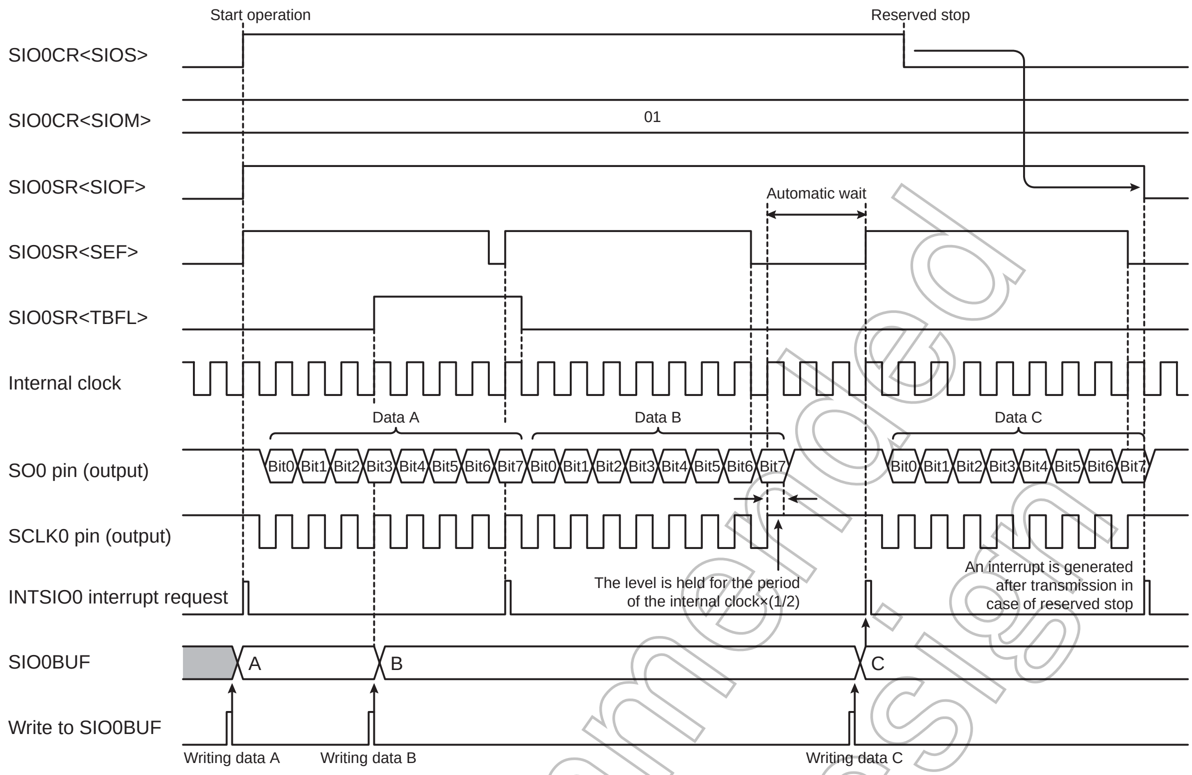


Figure 17-4 8-bit Transmit Mode (Internal Clock and Reserved Stop)

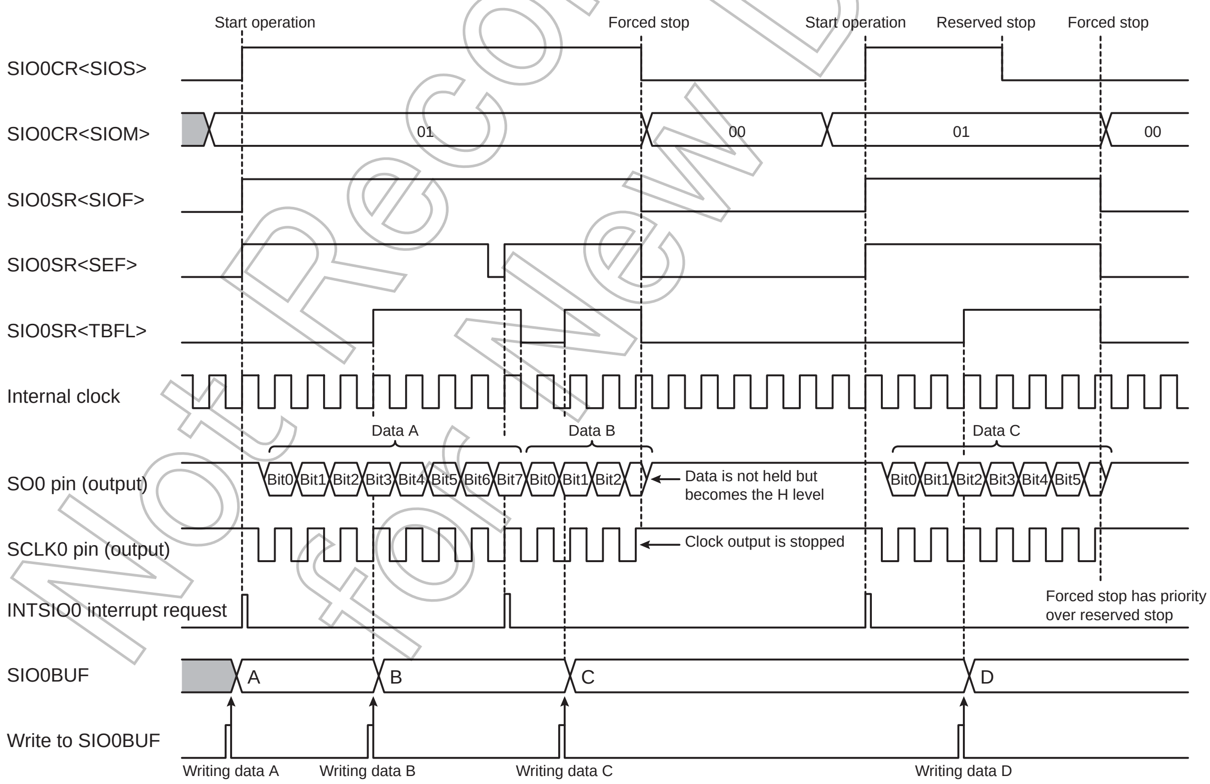


Figure 17-5 8-bit Transmit Mode (Internal Clock and Forced Stop)

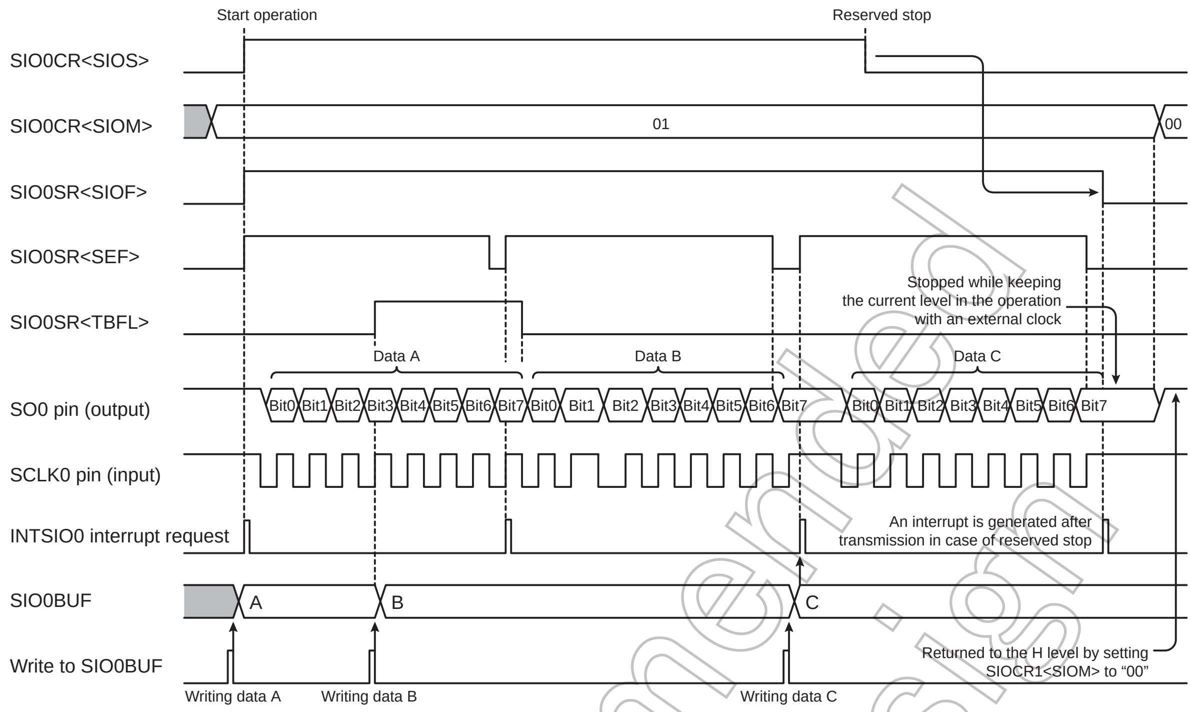


Figure 17-6 8-bit Transmit Mode (External Clock and Reserved Stop)

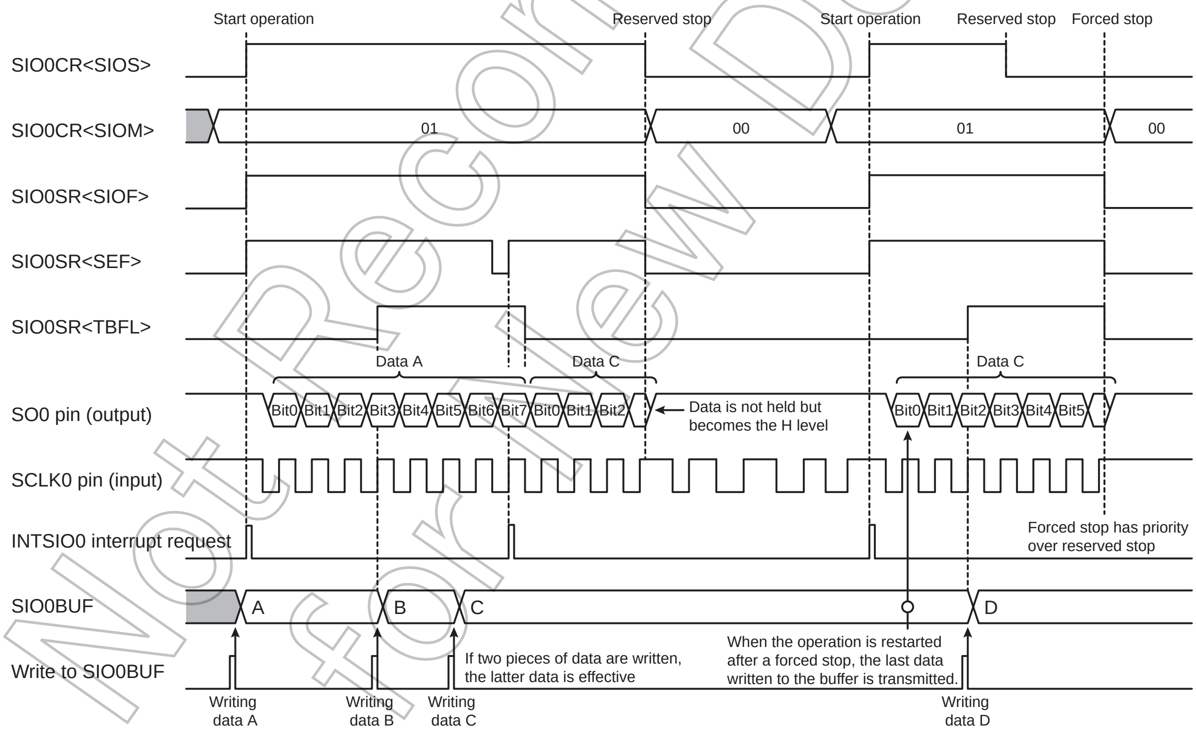


Figure 17-7 8-bit Transmit Mode (External Clock and Forced Stop)

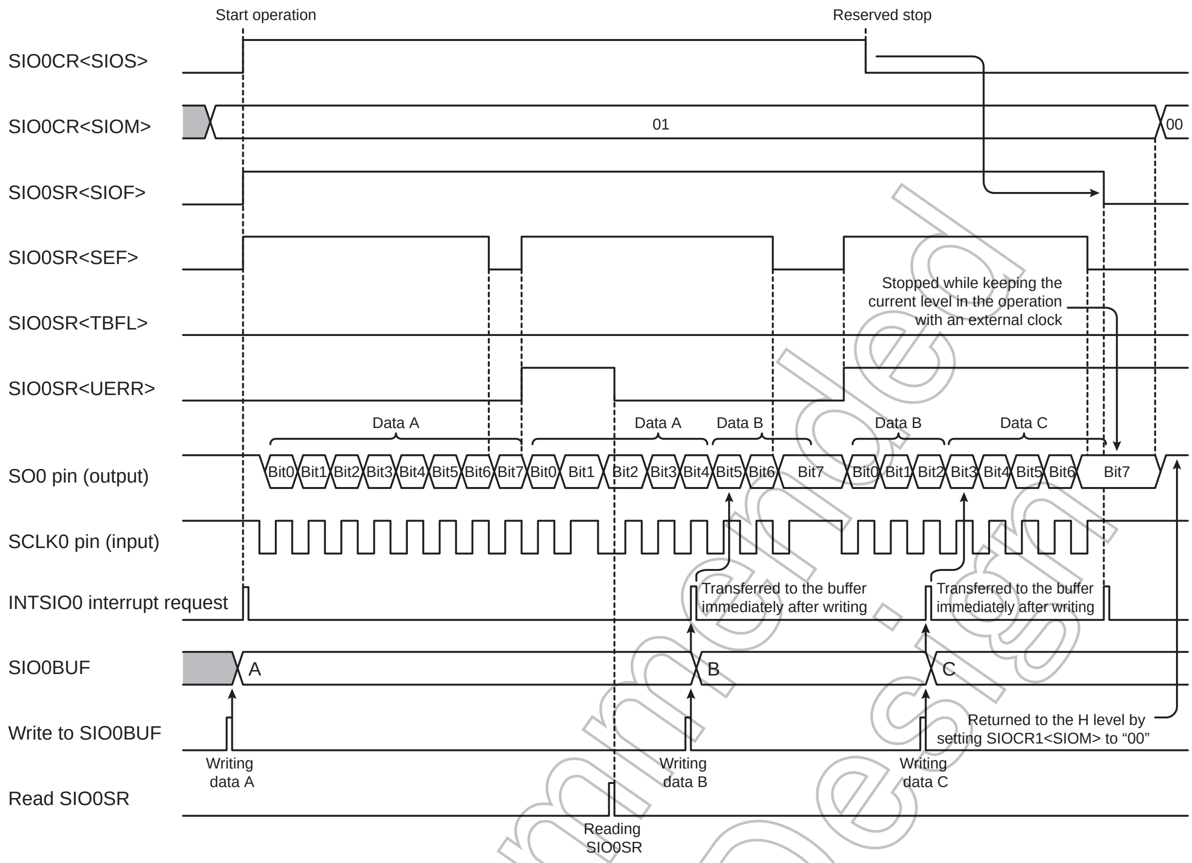


Figure 17-8 8-bit Transmit Mode (External Clock and Occurrence of Transmit Underrun Error)

17.5.2 8-bit Receive Mode

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

17.5.2.1 Setting

As in the case of the transmit mode, before starting the receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

Reception is started by setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

17.5.2.2 Starting the receive operation

Reception is started by setting SIO0CR<SIOS> to "1". External serial data is taken into the shift register from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1".

17.5.2.3 Operation on completion of reception

When the data reception is completed, the data is transferred from the shift register to SIO0BUF and an INTSIO0 interrupt request is generated. The receive completion flag SIO0SR<REND> is set to "1".

In the operation with the internal clock, the serial clock output is stopped until the receive data is read from SIO0BUF (automatic wait). At this time, SIO0SR<SEF> is set to "0". By reading the receive data from SIO0BUF, SIO0SR<SEF> is set to "1", the serial clock output is restarted and the receive operation continues.

In the operation with an external clock, data can be continuously received without reading the received data from SIO0BUF. In this case, data must be read from SIO0BUF before the subsequent data has been fully received. If the subsequent data is received completely before reading data from SIO0BUF, the overrun error flag SIO0SR<OERR> is set to "1". When an overrun error has occurred, set SIO0CR<SIOM> to "00" to abort the receive operation. The data received at the occurrence of an overrun error is discarded, and SIO0BUF holds the data value received before the occurrence of the overrun error.

SIO0SR<REND> is cleared to "0" by reading data from SIO0BUF. SIO0SR<OERR> is cleared by reading SIO0SR.

17.5.2.4 Stopping the receive operation

Set SIO0CR<SIOS> to "0" to stop the receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data has been completely received (reserved stop). At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF and SEF> are cleared to "0". Other SIO0SR registers keep their values.

The receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. If the internal clock is selected, the SCLK0 pin returns to the initial level.

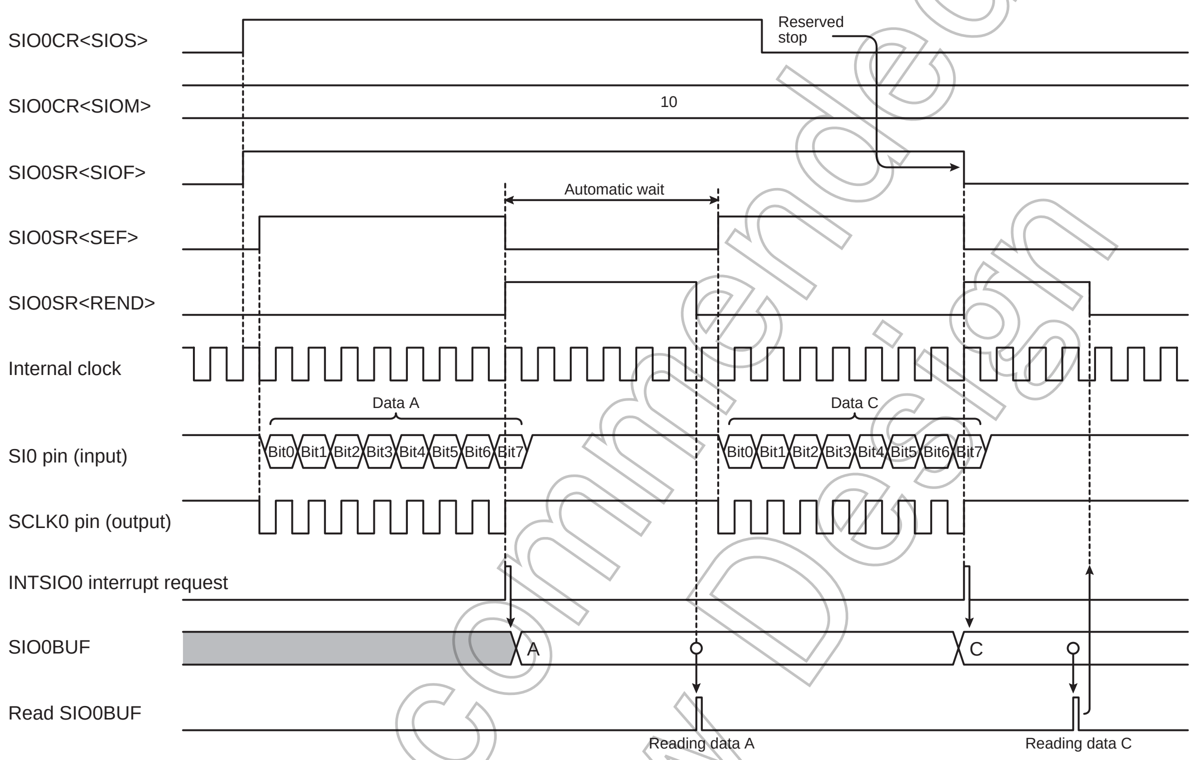


Figure 17-9 8-bit Receive Mode (Internal Clock and Reserved Stop)

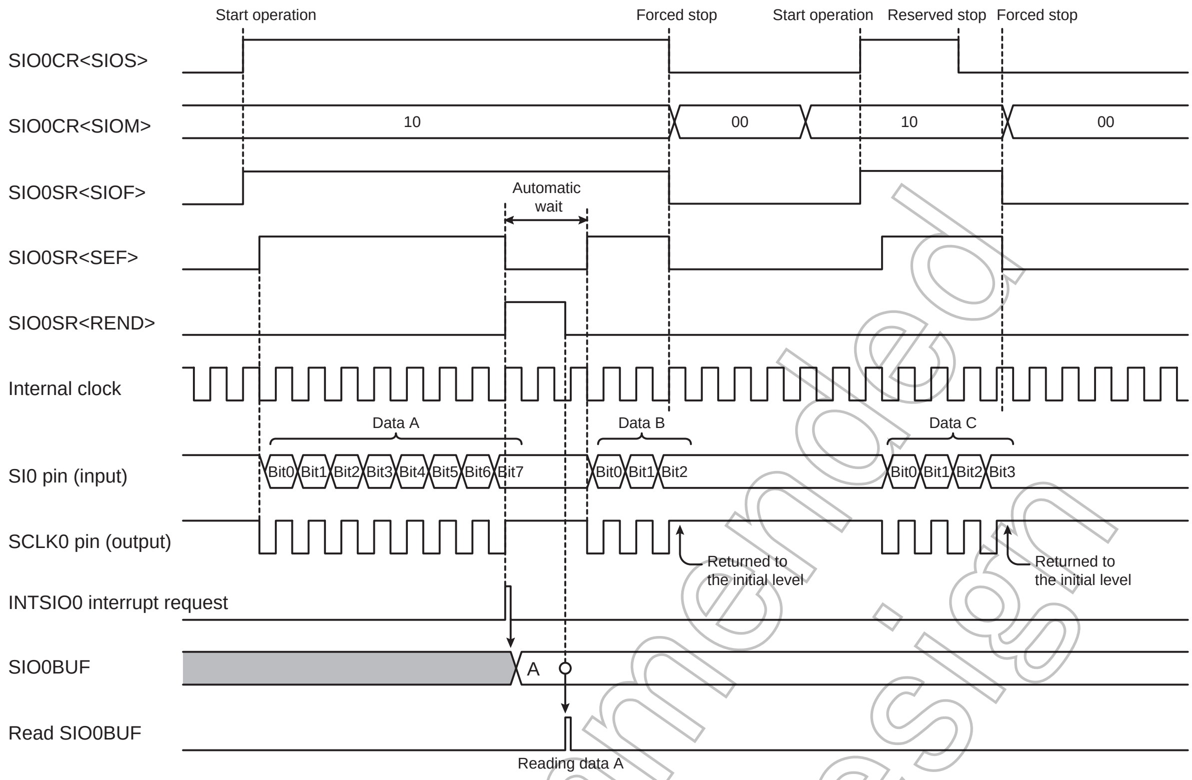


Figure 17-10 8-bit Receive Mode (Internal Clock and Forced Stop)

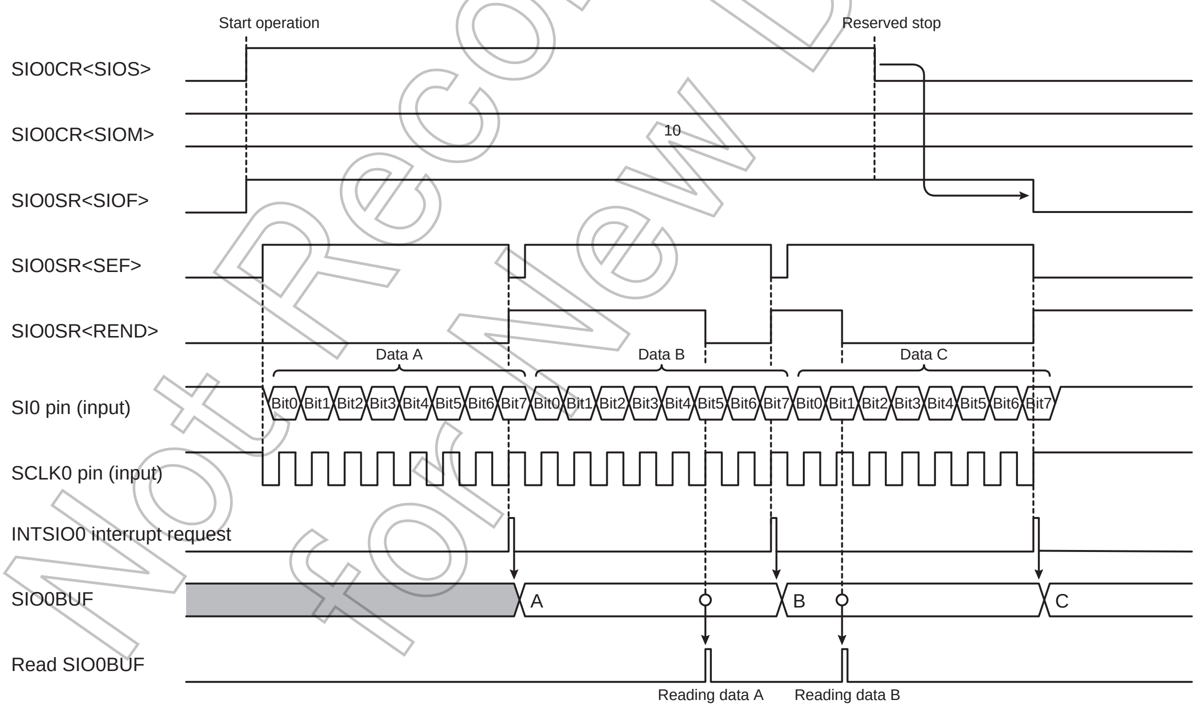


Figure 17-11 8-bit Receive Mode (External Clock and Reserved Stop)

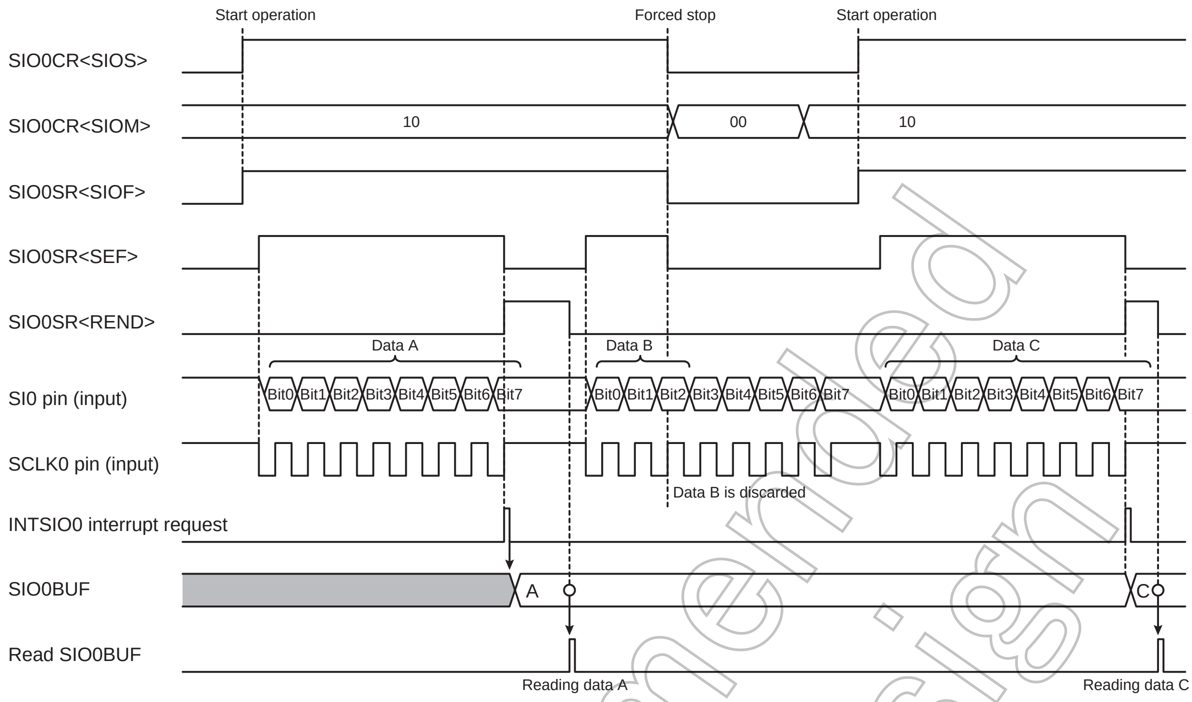


Figure 17-12 8-bit Receive Mode (External Clock and Forced Stop)

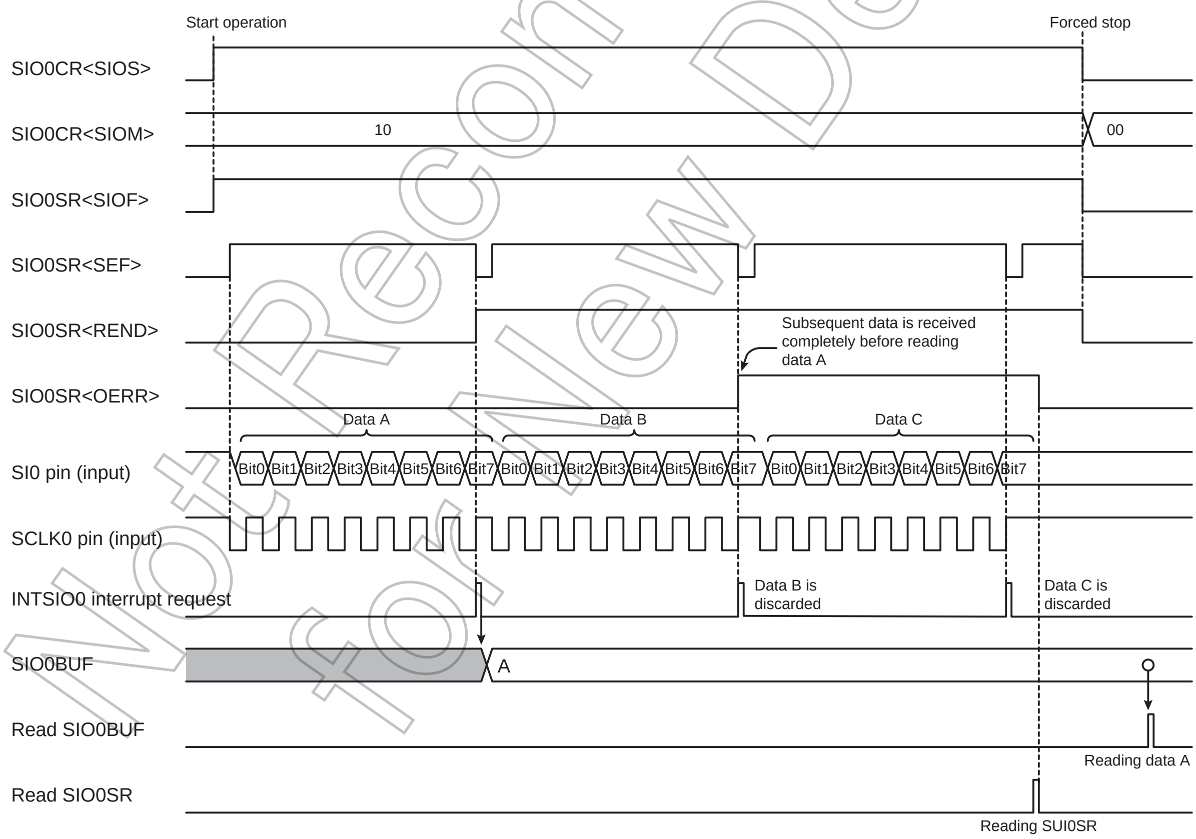


Figure 17-13 8-bit Receive Mode (External Clock and Occurrence of Overrun Error)

17.5.3 8-bit transmit/receive mode

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

17.5.3.1 Setting

Before starting the transmit/receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

The transmit/receive operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIOCR<SIOS> is valid.

17.5.3.2 Starting the transmit/receive operation

The transmit/receive operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and the serial data is transmitted from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>. At the same time, the serial data is received from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

The transmit data becomes undefined if the transmit/receive operation is started without writing any transmit data to SIO0BUF.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of data is received.

17.5.3.3 Transmit buffer and shift operation

If any data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

If any data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

17.5.3.4 Operation on completion of transmission/reception

When the data transmission/reception is completed, SIO0SR<REND> is set to "1" and an INTSIO0 interrupt request is generated. The operation varies depending on the operating clock.

(1) When the internal clock is used

If SIO0SR<TBFL> is "1", it is cleared to "0" and the transmit/receive operation continues. If SIO0SR<REND> is already "1", SIO0SR<OERR> is set to "1".

If SIO0SR<TBFL> is "0", the transmit/receive operation is aborted. The SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the subsequent data is written to SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the clock and the transmit/receive operation is restarted. To confirm the receive data, read it from SIO0BUF before writing data to SIO0BUF.

(2) When an external clock is used

The transmit/receive operation continues. If the external serial clock is input without writing any data to SIO0BUF, the last data value set to SIO0BUF is re-transmitted. At this time, the transmit underrun error flag SIO0SR<UERR> is set to "1".

When the next 8-bit data is received completely before SIO0BUF is read, or in the state of SIO0SR<REND>="1", SIO0SR<OERR> is set to "1".

17.5.3.5 Stopping the transmit/receive operation

Set SIO0CR<SIOS> to "0" to stop the transmit/receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data is received completely. At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "H" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit/receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

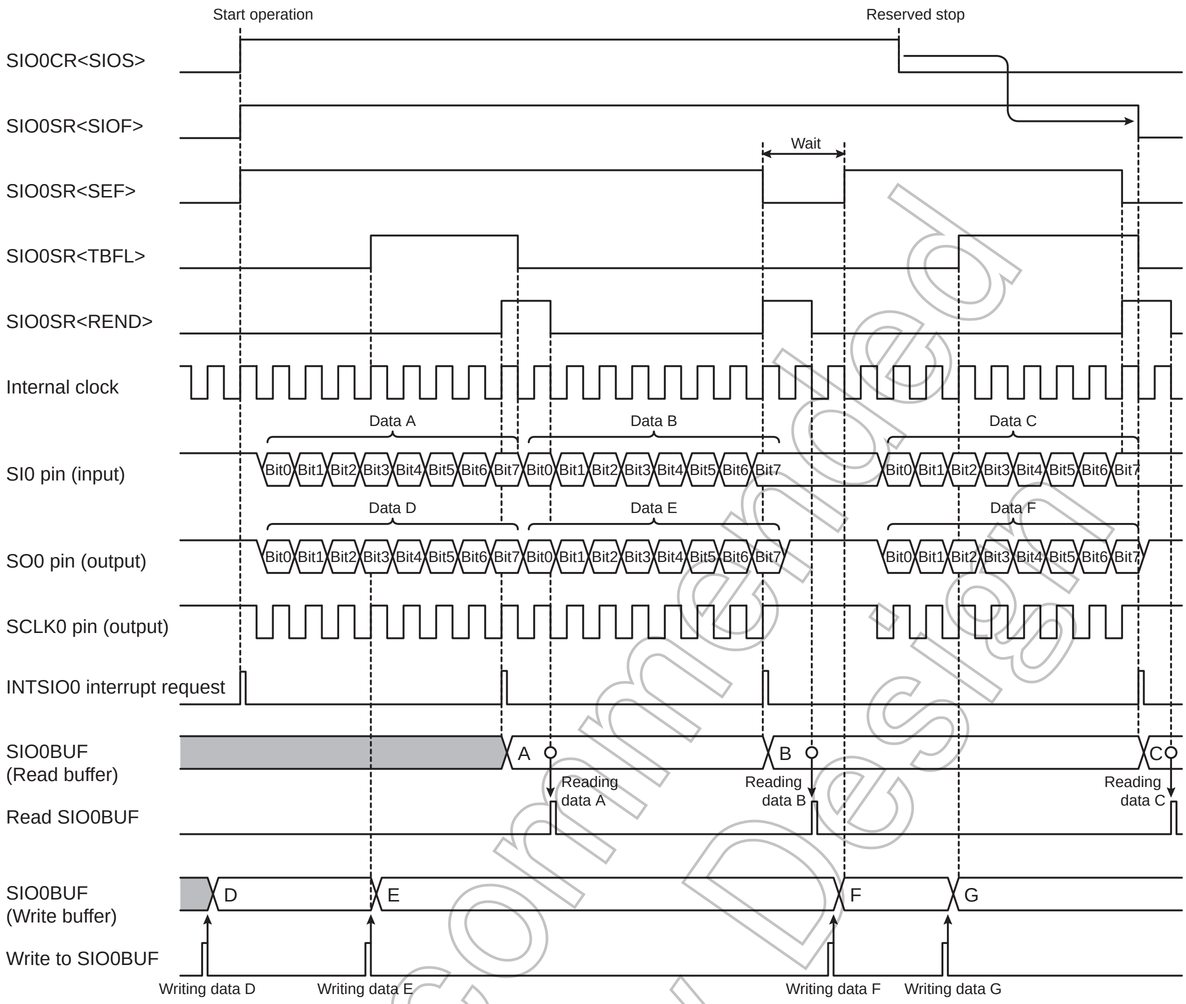


Figure 17-14 8-bit Transmit/Receive Mode (Internal Clock and Reserved Stop)

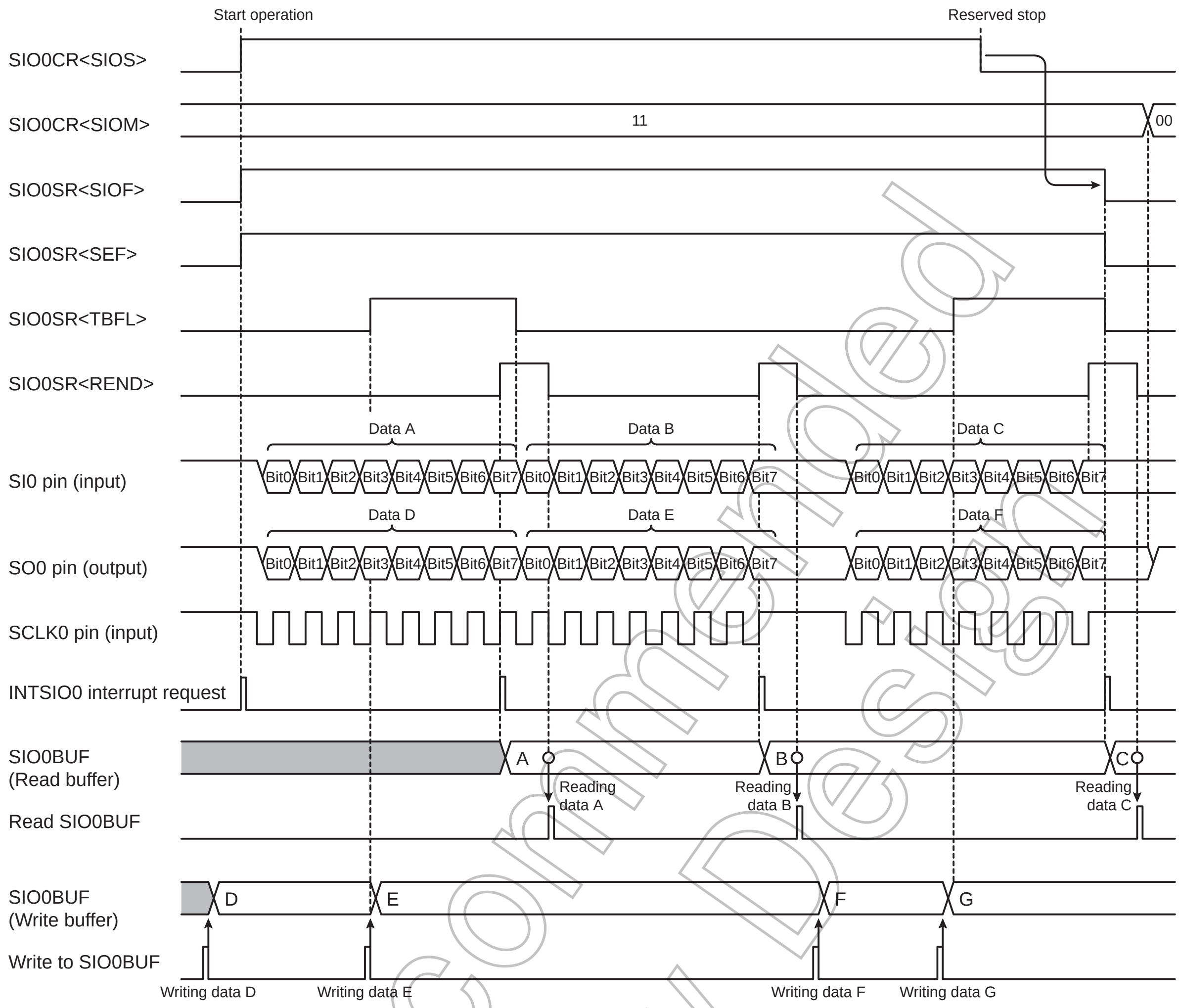


Figure 17-15 8-bit Transmit/Receive Mode (External Clock and Reserved Stop)

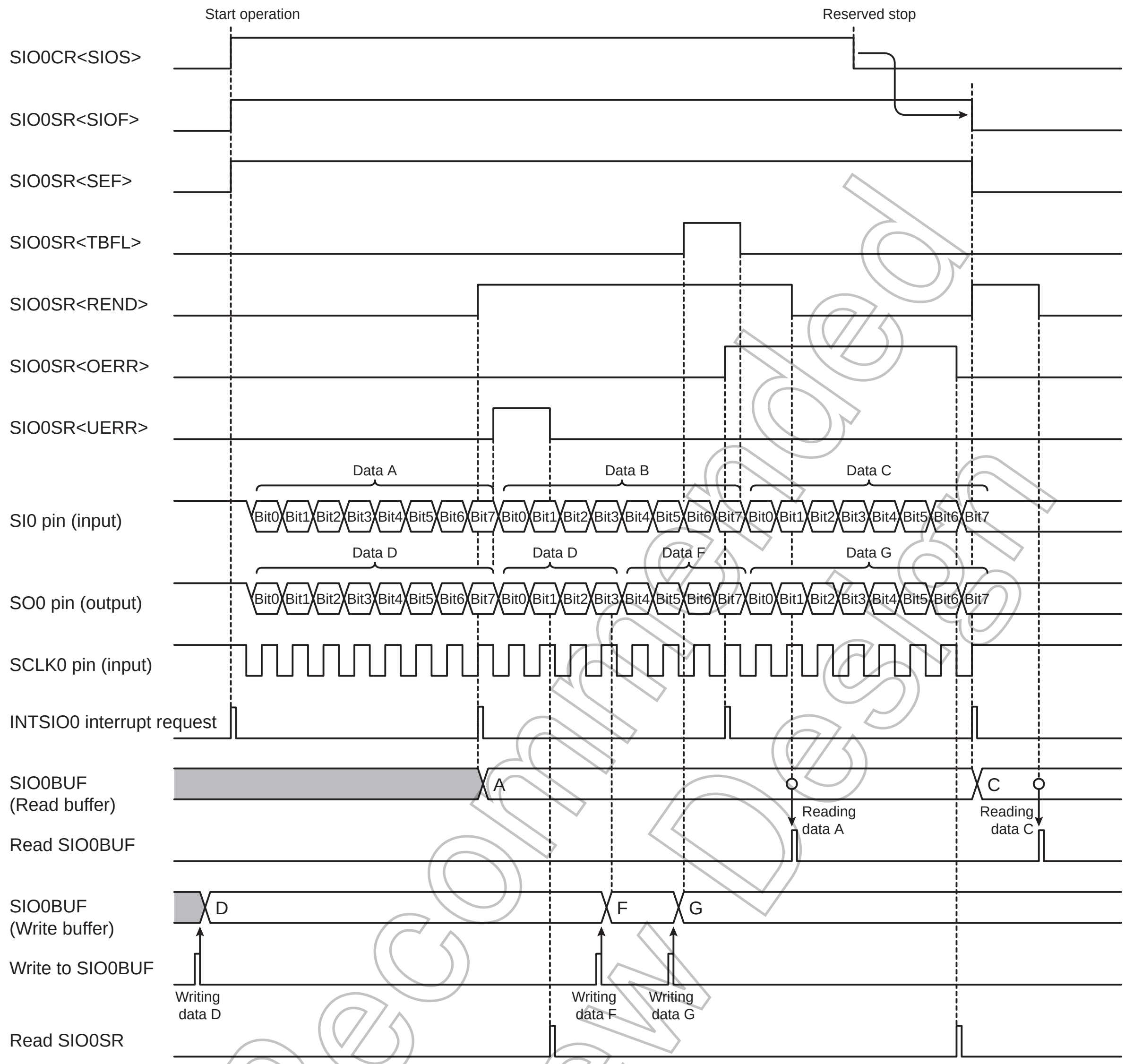


Figure 17-16 8-bit Transmit/Receive Mode (External Clock, Occurrence of Transmit Underrun Error and Occurrence of Overrun Error)

17.6 AC Characteristics

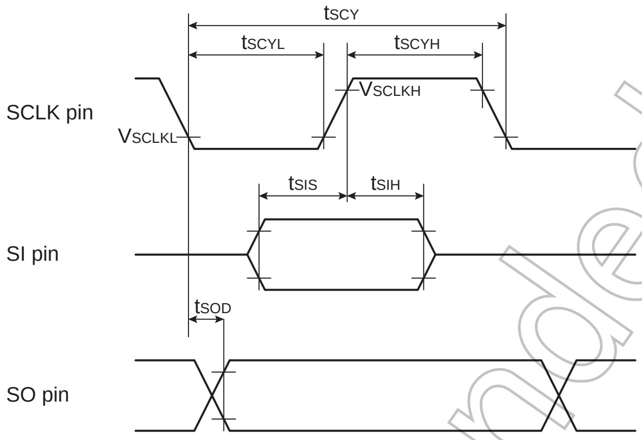
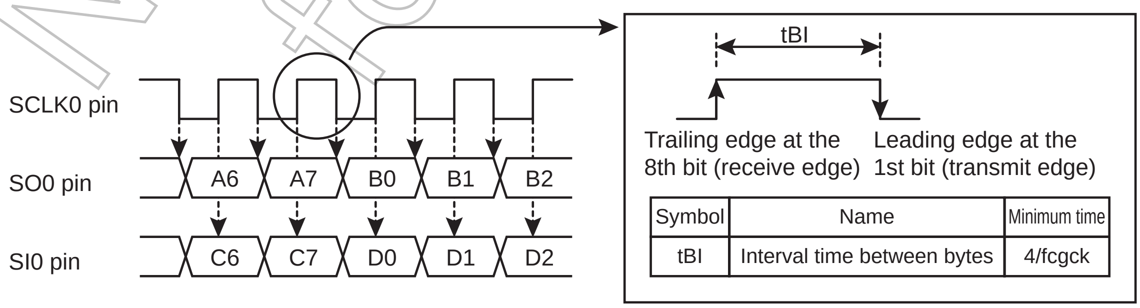


Figure 17-17 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 V - 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
SCLK cycle time	t_{SCY}	Internal clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	ns
SCLK "L" pulse width	t_{SCYL}		1 / fcgck − 25	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck − 15	-	-	
SI input setup time	t_{SIS}		60	-	-	
SI input hold time	t_{SIH}		35	-	-	
SO output delay time	t_{SOD}		−50	-	50	
SCLK cycle time	t_{SCY}	External clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	
SCLK "L" pulse width	t_{SCYL}		1 / fcgck	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck	-	-	
SI input setup time	t_{SIS}		50	-	-	
SI input hold time	t_{SIH}		50	-	-	
SO output delay time	t_{SOD}		0	-	60	
SCLK low-level input voltage	t_{SCLKL}		0	-	V_{DD} × 0.30	V
SCLK high-level input voltage	t_{SCLKH}		V_{DD} × 0.70	-	V_{DD}	



Symbol	Name	Minimum time
tBI	Interval time between bytes	4/fcgck

Figure 17-18 Interval time between bytes

17.7 Revision History

Rev	Description
RA001	"Table 17-3 Transfer Baud Rate" Revised table (Add some fcgck condition).
	"17.6 AC Characteristics" Revised table (Add some fcgck condition).

18. Serial Bus Interface (SBI)

The TMP89CM42 contains 1 channels of serial bus interface (SBI).

The serial bus interface supports serial communication conforming to the I^2C bus standards. It has clock synchronization and arbitration functions, and supports the multi-master in which multiple masters are connected on a bus. It also supports the unique free data format.

18.1 Communication Format

18.1.1 I^2C bus

The I^2C bus is connected to devices via the SDA0 and SCL0 pins and can communicate with multiple devices.

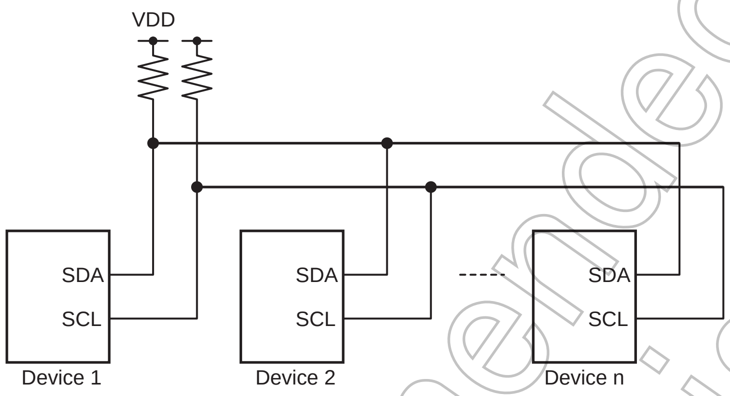


Figure 18-1 Device Connections

Communications are implemented between a master and slave.

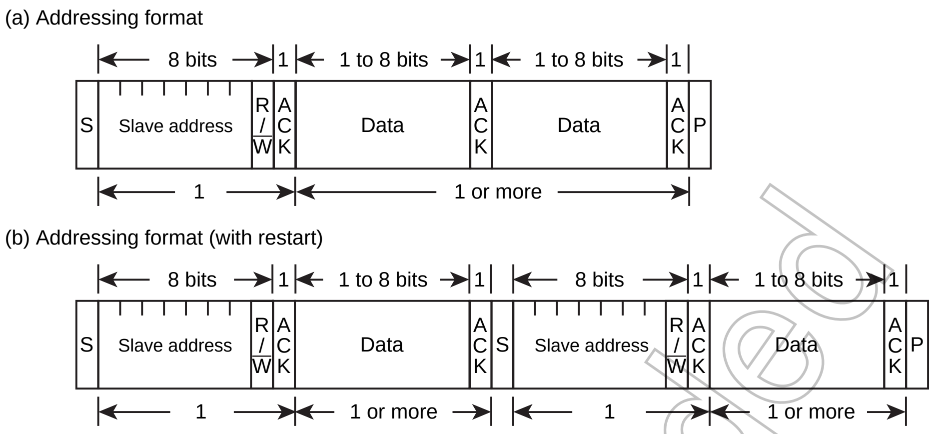
The master transmits the start condition, the slave addresses, the direction bit and the stop condition to the slave(s) connected to the bus, and transmits and receives data.

The slave detects these conditions transmitted from the master by the hardware, and transmits and receives data.

The data format of the I^2C bus that can communicate via the serial bus interface is shown in Figure 18-2.

The serial bus interface does not support the following functions among those specified by the I^2C bus standards:

1. Start byte
2. 10-bit addressing
3. SDA and SCL pins falling edge slope control



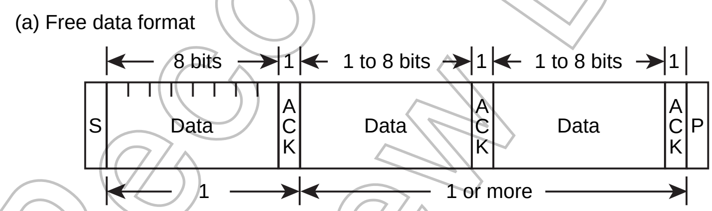
S : Start condition
R/W : Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 18-2 Data Format of I²C Bus

18.1.2 Free data format

The free data format is for communication between a master and slave.

In the free data format, the slave address and the direction bit are processed as data.



S : Start condition
R/W : Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 18-3 Free Data Format

18.2 Configuration

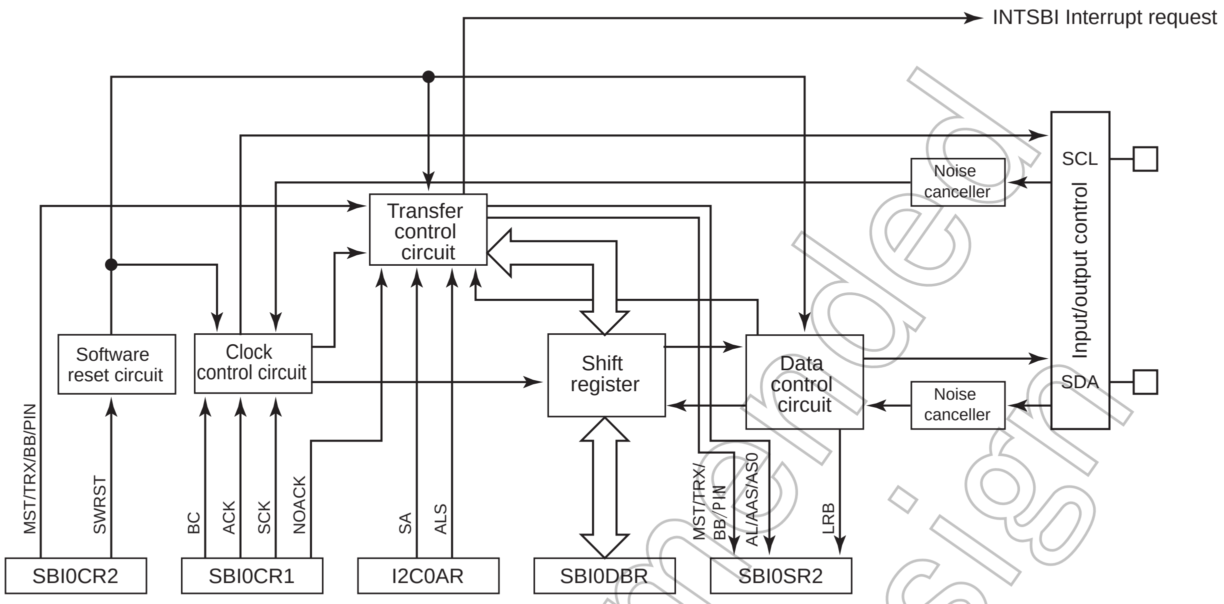


Figure 18-4 Serial Bus Interface 0 (SBI0)

18.3 Control

The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface status register 2 (SBI0SR2)
- Serial bus interface data buffer register (SBI0DBR)
- I^2C bus address register (I2C0AR)

In addition, the serial bus interface has low power consumption registers that save power when the serial bus interface is not being used.

Low power consumption register 1

POFFCR1 (0x0F75)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	SBI0EN	-	-	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0	Disable
		1	Enable
UART1EN	UART1 control	0	Disable
		1	Enable
UART0EN	UART0 control	0	Disable
		1	Enable

Note 1: When SBI0EN is cleared to "0", the clock supply to the serial bus interface is stopped. At this time, the data written to the serial bus interface control registers is invalid. When the serial bus interface is used, set SBI0EN to "1" and then write the data to the serial bus interface control registers.

Serial bus interface control register 1

SBI0CR1 (0x0022)

	7	6	5	4	3	2	1	0
Bit Symbol	BC			ACK	NOACK	SCK		
Read/Write	R/W			R/W	R/W	R/W		
After reset	0	0	0	0	0	0	0	0

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock oscillation circuit clock

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Note 5: When fcgck is 4MHz, SCK should be not set to 0y000, 0y001 or 0y010 because it is not possible to satisfy the bus specification of fast mode.

Serial bus interface control register 2

SBI0CR2 (0x0023)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	SBIM	-	SWRST	
Read/Write	W	W	W	W	W	R	W	
After reset	0	0	0	1	0	0	0	

Note 1: When SBI0CR2<SBIM> is "0", no value can be written to SBI0CR2 except SBI0CR2<SBIM>. Before writing values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

Note 2: Don't change the contents of the registers, except SBI0CR2<SWRST>, when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: Make sure that the port is in a high state before switching the port mode to the serial bus interface mode. Make sure that the bus is free before switching the serial bus interface mode to the port mode.

Note 4: SBI0CR2 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 5: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 6: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface status register 2

SBI0SR2 (0x0023)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	1	0	0	0	*

Note 1: * : Unstable

Note 2: When SBI0CR2<SBIM> becomes "0", SBI0SR is initialized.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

I²C bus address register

I2C0AR (0x0024)

	7	6	5	4	3	2	1	0
Bit Symbol	SA							ALS
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1: Don't set I2C0AR<SA> to "0x00". If it is set to "0x00", the slave address is deemed to be matched when the I²C bus standard start byte ("0x01") is received in the slave mode.

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface data buffer register

SBI0DBR (0x0025)

	7	6	5	4	3	2	1	0
Bit Symbol	SBI0DBR							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

Note 1: Write the transmit data beginning with the most significant bit (bit 7).

Note 2: SBI0DBR has individual writing and reading buffers, and written data cannot be read out. Therefore, SBI0DBR must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 3: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 4: To set SBI0CR2<PIN> to "1" by writing the dummy data to SBI0DBR, write 0x00. Writing any data other than 0x00 causes an improper value in the subsequently received data.

Note 5: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

18.4 Functions

18.4.1 Low Power Consumption Function

The serial bus interface has a low power consumption register (POFFCR1) that saves power when the serial bus interface is not being used.

Setting POFFCR1<SBI0EN> to "0" disables the basic clock supply to the serial bus interface to save power. Note that this makes the serial bus interface unusable. Setting POFFCR1<SBI0EN> to "1" enables the basic clock supply to the serial bus interface and makes external interrupts usable.

After reset, POFFCR1<SBI0EN> is initialized to "0", and this makes the serial bus interface unusable. When using the serial bus interface for the first time, be sure to set POFFCR1<SBI0EN> to "1" in the initial setting of the program (before the serial bus interface control registers are operated).

Do not change POFFCR1<SBI0EN> to "0" during the serial bus interface operation, otherwise serial bus interface may operate unexpectedly.

18.4.2 Selecting the slave address match detection and the GENERAL CALL detection

SBI0CR1<NOACK> enables and disables the slave address match detection and the GENERAL CALL detection in the slave mode.

Clearing SBI0CR1<NOACK> to "0" enables the slave address match detection and the GENERAL CALL detection.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. The slave addresses and "GENERAL CALL" sent from the master are ignored. No acknowledgement is returned and no interrupt request is generated.

In the master mode, SBI0CR1<NOACK> is ignored and has no influence on the operation.

Note: If SBI0CR1<NOACK> is cleared to "0" during data transfer in the slave mode, it remains at "1" and returns an acknowledge signal of data transfer.

18.4.3 Selecting the number of clocks for data transfer and selecting the acknowledgement or non-acknowledgment mode

1-word data transfer consists of data and an acknowledge signal. When the data transfer is finished, an interrupt request is generated.

SBI0CR1<BC> is used to select the number of bits of data to be transmitted/received subsequently.

The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

The master device generates the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode. The slave device counts the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

The master device does not generate the clocks for an acknowledge signal. The slave device does not count the clocks for an acknowledge signal.

18.4.3.1 Number of clocks for data transfer

The number of clocks for data transfer is set by using SBI0CR1<BC> and SBI0CR1<ACK>.

The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

In the acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, generates the clocks for an acknowledge signal, and generates an interrupt request.

The slave device counts the clocks that correspond to the data bits, counts the clocks for an acknowledge signal, and generates an interrupt request.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

In the non-acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, and generates an interrupt request.

The slave device counts the clocks that correspond to the data bits, and generates an interrupt request.

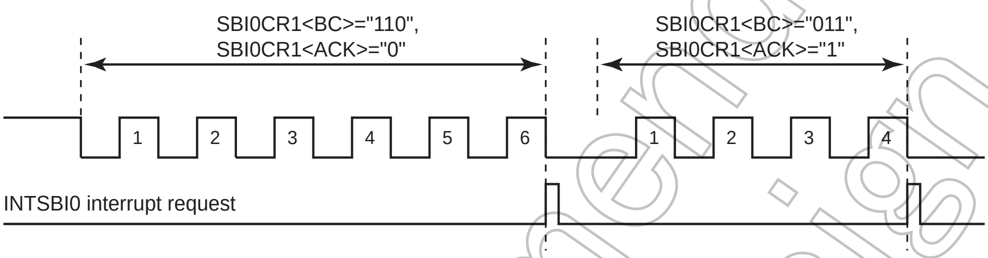


Figure 18-5 Number of Clocks for Data Transfer and SBI0CR1<BC> and SBI0CR1<ACK>

The relationship between the number of clocks for data transfer and SBI0CR1<BC> and SBI0CR1<ACK> is shown in Table 18-1.

Table 18-1 Relationship between the Number of Clocks for Data Transfer and SBI0CR1<BC> and SBI0CR1<ACK>

BC	ACK=0 (Non-acknowledgment mode)		ACK=1 (Acknowledgment mode)	
	Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

BC is cleared to "000" by the start condition.

Therefore, the slave address and the direction bit are always transferred in 8-bit units. In other cases, BC keeps the set value.

Note: SBI0CR1<ACK> must be set before transmitting or receiving a slave address. When SBI0CR1<ACK> is cleared, the slave address match detection and the direction bit detection are not executed properly.

18.4.3.2 Output of an acknowledge signal

In the acknowledgment mode, the SDA0 pin changes as follows during the period of the clocks for an acknowledge signal.

- In the master mode

In the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal. In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

- In the slave mode

When a match between the received slave address and the slave address set to I2C0AR<SA> is detected or when a GENERAL CALL is received, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

During the data transfer after the slave address match is detected or a "GENERAL CALL" is received in the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal.

In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated. Table 18-2 shows the states of the SCL0 and SDA0 pins in the acknowledgment mode.

Note: In the non-acknowledgment mode, the clocks for an acknowledge signal are not generated or counted, and thus no acknowledge signal is output.

Table 18-2 States of the SCL0 and SDA0 Pins in the Acknowledgment Mode

Mode	Pin	Condition	Transmitter	Receiver
Master	SCL0	-	Add the clocks for an acknowledge signal.	Add the clocks for an acknowledge signal
	SDA0	-	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin
Slave	SCL0	-	Count the clocks for an acknowledge signal	Count the clocks for an acknowledge signal
	SDA0	When the slave address match is detected or a "GENERAL CALL" is received	-	Output the low level as an acknowledge signal to the pin
		During transfer after the slave address match is detected or a "GENERAL CALL" is received	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin

18.4.4 Serial clock

18.4.4.1 Clock source

SBI0CR1<SCK> is used to set the HIGH and LOW periods of the serial clock to be output in the master mode.

SCK	t_{HIGH}(m/fcgck)	t_{LOW}(n/fcgck)
	m	n
000:	9	12
001:	11	14
010:	15	18
011:	23	26
100:	39	42
101:	71	74
110:	135	138
111:	263	266

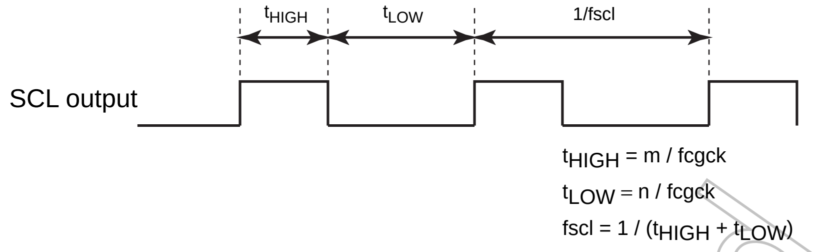


Figure 18-6 SCL Output

Note: There are cases where the HIGH period differs from t_{HIGH} selected at SBI0CR1<SCK> when the rising edge of the SCL pin becomes blunt due to the load capacity of the bus.

In the master mode, the hold time when the start condition is generated is t_{HIGH} [s] and the setup time when the stop condition is generated is t_{HIGH} [s].

When SBI0CR2<PIN> is set to "1" in the slave mode, the time that elapses before the release of the SCL pin is t_{LOW} [s].

In both the master and slave modes, the high level period must be 3/fcgck[s] or longer and the low level period must be 5/fcgck[s] or longer for the externally input clock, regardless of the SBI0CR1<SCK> setting.

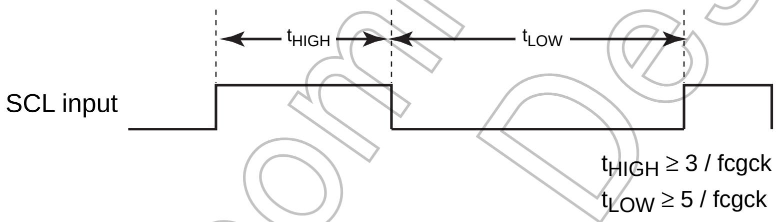


Figure 18-7 SCL Input

18.4.4.2 Clock synchronization

In the I²C bus, due to the structure of the pin, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate the clock pulse of another master device which generates a high-level clock pulse. Therefore, the master outputting the high level must detect this to correspond to it.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

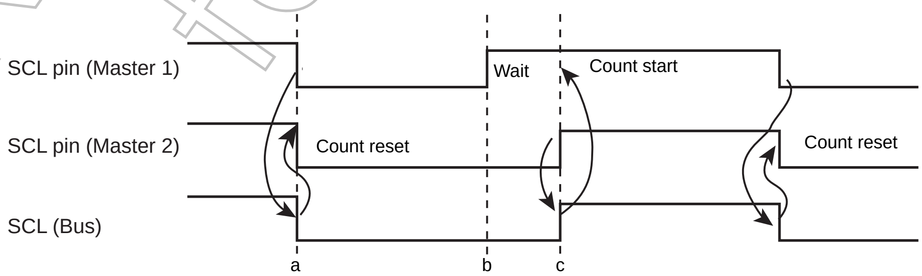


Figure 18-8 Example of Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is determined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

18.4.5 Master/slave selection

To set a master device, SBI0CR2<MST> should be set to "1".

To set a slave device, SBI0CR2<MST> should be cleared to "0". When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<MST> is cleared to "0" by the hardware.

18.4.6 Transmitter/receiver selection

To set the device as a transmitter, SBI0CR2<TRX> should be set to "1". To set the device as a receiver, SBI0CR2<TRX> should be cleared to "0".

For the I^{2}C bus data transfer in the slave mode, SBI0CR2<TRX> is set to "1" by the hardware if the direction bit (R/$\overline{W}$) sent from the master device is "1", and is cleared to "0" if the bit is "0".

In the master mode, after an acknowledge signal is returned from the slave device, SBI0CR2<TRX> is cleared to "0" by hardware if a transmitted direction bit is "1", and is set to "1" by hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<TRX> is cleared to "0" by the hardware. Table 18-3 shows SBI0CR2<TRX> changing conditions in each mode and SBI0CR2<TRX> value after changing.

Note: When SBI0CR1<NOACK> is "1", the slave address match detection and the GENERAL CALL detection are disabled, and thus SBI0CR2<TRX> remains unchanged.

Table 18-3 SBI0CR1<TRX> Operation in Each Mode

Mode	Direction bit	Changing condition	TRX after changing
Slave mode	"0"	A received slave address is the same as the value set to I2C0AR<SA>	"0"
	"1"		"1"
Master mode	"0"	ACK signal is returned	"1"
	"1"		"0"

When the serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating the start condition. SBI0CR2<TRX> is not changed by the hardware.

18.4.7 Start/stop condition generation

When SBI0SR2<BB> is "0", a slave address and a direction bit which are set to the SBI0DBR are output on a bus after generating a start condition by writing "1" to SBI0CR2 <MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>. It is necessary to set SBI0CR1<ACK> to "1" before generating the start condition.

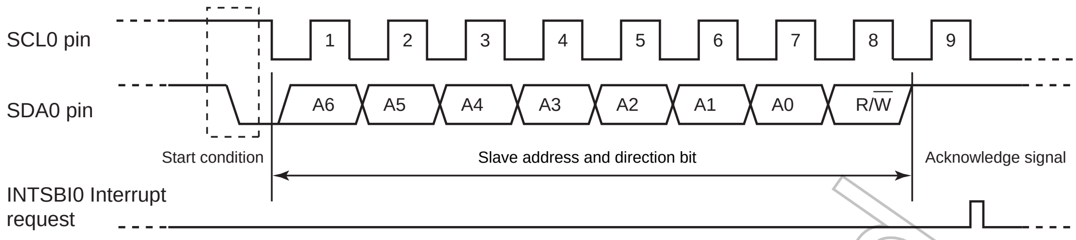


Figure 18-9 Generating the Start Condition and a Slave Address

When SBI0CR2<BB> is "1", the sequence of generating the stop condition on the bus is started by writing "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and writing "0" to SBI0CR2<BB>.

When a stop condition is generated. The SCL line on a bus is pulled down to the low level by another device, a stop condition is generated after releasing the SCL line.

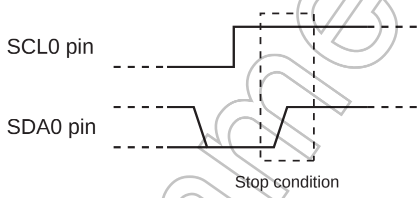


Figure 18-10 Stop Condition Generation

The bus condition can be indicated by reading the contents of SBI0SR2<BB>. SBI0SR2<BB> is set to "1" when the start condition on the bus is detected (Bus Busy State) and is cleared to "0" when the stop condition is detected (Bus Free State).

18.4.8 Interrupt service request and release

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by SBI0CR1<BC> and SBI0CR1<ACK> is complete, a serial bus interface interrupt request (INTSBI0) is generated.

In the slave mode, a serial bus interface interrupt request (INTSBI0) is generated when the above and following conditions are satisfied:

- At the end of the acknowledge signal when the received slave address matches to the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of the slave address or receiving of "GENERAL CALL"

When a serial bus interface interrupt request occurs, SBI0CR2<PIN> is cleared to "0". During the time that SBI0CR2<PIN> is "0", the SCL0 pin is pulled down to the low level.

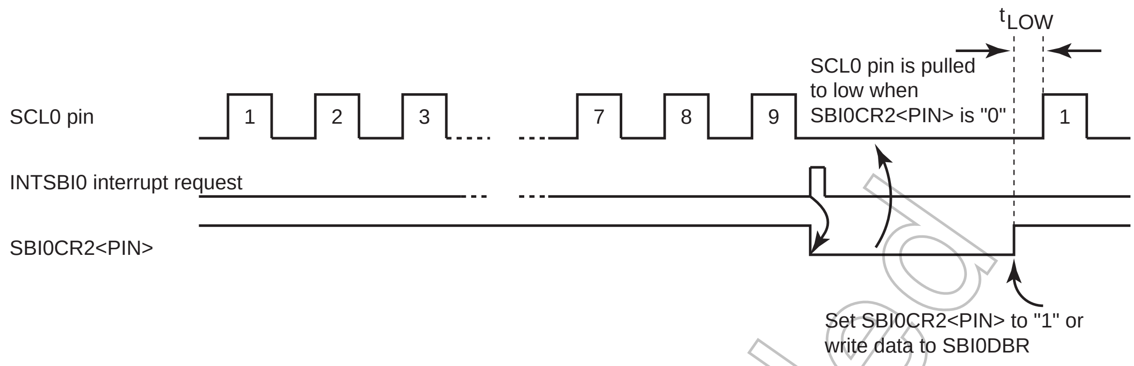


Figure 18-11 SBI0CR2<PIN> and SCL0 Pin

Writing data to SBI0DBR sets SBI0CR2<PIN> to "1". The time from SBI0CR2<PIN> being set to "1" until the SBI0 pin is released takes t_{LOW}.

Although SBI0CR2<PIN> can be set to "1" by the software, SBI0CR2<PIN> can not be cleared to "0" by the software.

18.4.9 Setting of serial bus interface mode

SBI0CR2<SBIM> is used to set serial bus interface mode.

Setting SBI0CR2<SBIM> to "1" selects the serial bus interface mode. Setting it to "0" selects the port mode.

Set SBI0CR2<SBIM> to "1" in order to set serial bus interface mode. Before setting of serial bus interface mode, confirm serial bus interface pins in a high level, and then, write "1" to SBI0CR2<SBIM>.

And switch a port mode after confirming that a bus is free and set SBI0CR2<SBIM> to "0".

Note: When SBI0CR2<SBIM> is "0", no data can be written to SBI0CR2 except SBI0CR2<SBIM>. Before setting values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

18.4.10 Software reset

The serial bus interface circuit has a software reset function that initializes the serial bus interface circuit. If the serial bus interface circuit locks up, for example, due to noise, it can be initialized by using this function.

A software reset is generated by writing "10" and then "01" to SBI0CR2<SWRST>.

After a software reset is generated, the serial bus interface circuit is initialized and all the bits of SBI0CR2 register, except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR<SA> and SBI0SR2 registers, are initialized.

18.4.11 Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point "a". After that, when Master 1 outputs "1" and Master 2 outputs "0", since the SDA line of a bus is wired AND, the SDA line is pulled down to the low level by Master 2. When the SCL line of a bus is pulled-up at point "b", the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

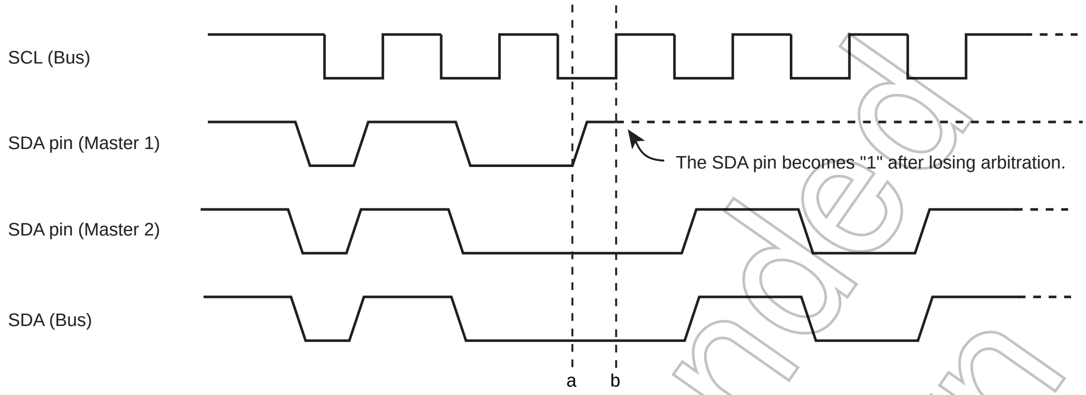


Figure 18-12 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and SBI0SR2<AL> is set to "1".

When SBI0SR2<AL> is set to "1", SBI0CR2<MST> and SBI0CR2<TRX> are cleared to "0" and the mode is switched to a slave receiver mode. Thus, the serial bus interface circuit stops output of clock pulses during data transfer after the SBI0SR2<AL> is set to "1". After the data transfer is completed, SBICR2<PIN> is cleared to "0" and the SCL pin is pulled down to the low level.

SBI0SR2<AL> is cleared to "0" by writing data to the SBI0DBR, reading data from the SBI0DBR or writing data to the SBI0CR2.

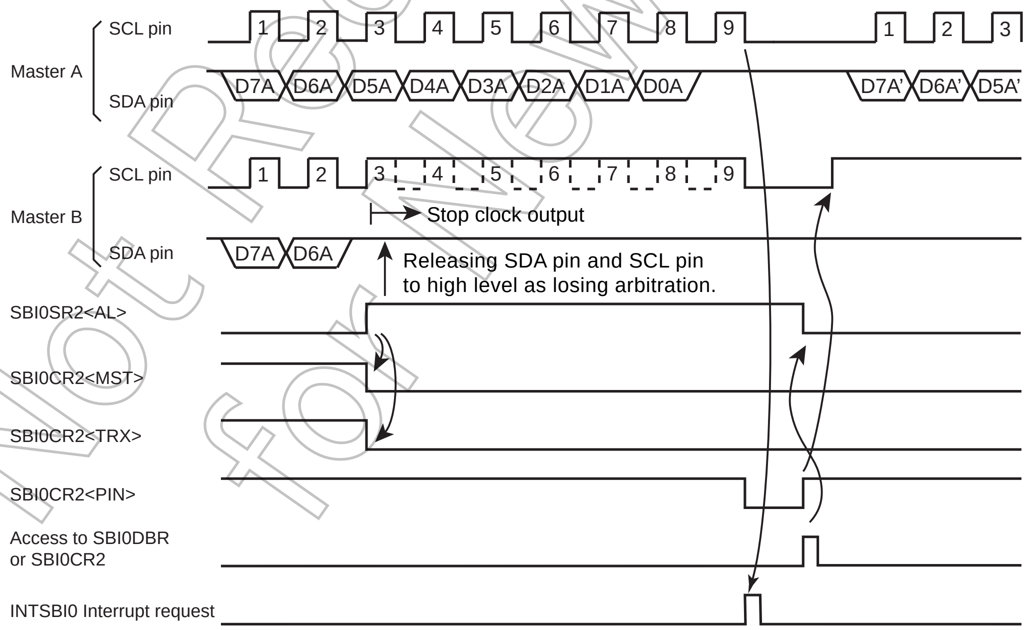


Figure 18-13 Example When Master B is a Serial Bus Interface Circuit

18.4.12 Slave address match detection monitor

In the slave mode, SBI0SR2<AAS> is set to "1" when the received data is "GENERAL CALL" or the received data matches the slave address setting by I2C0AR<SA> with SBI0CR1<NOACK> set at "0" and the I^{2}C bus mode is active (I2C0AR<ALS>="0").

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AAS> remains at "0" even if a "GENERAL CALL" is received or the same slave address as the I2C0AR<SA> set value is received.

When a serial bus interface circuit operates in the free data format (I2C0AR<ALS>= "1"), SBI0SR2<AAS> is set to "1" after receiving the first 1-word of data. SBI0SR2<AAS> is cleared to "0" by writing data to the SBI0DBR or reading data from the SBI0DBR.

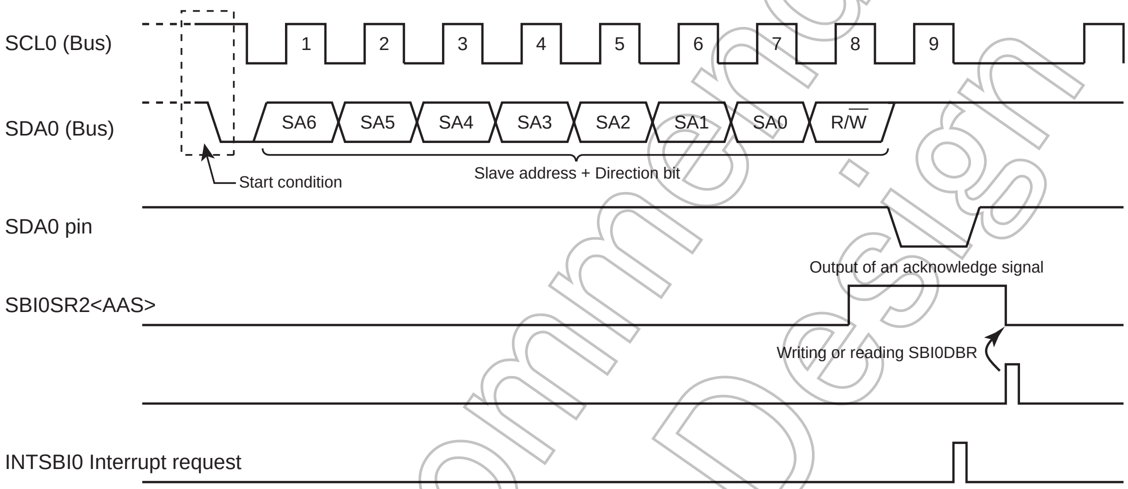


Figure 18-14 Changes in the Slave Address Match Detection Monitor

18.4.13 GENERAL CALL detection monitor

SBI0SR2<AD0> is set to "1" when SBI0CR1<NOACK> is "0" and GENERAL CALL (all 8-bit received data is "0" immediately after a start condition) in a slave mode.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AD0> remains at "0" even if a "GENERAL CALL" is received.

SBI0SR2<AD0> is cleared to "0" when a start or stop condition is detected on a bus.

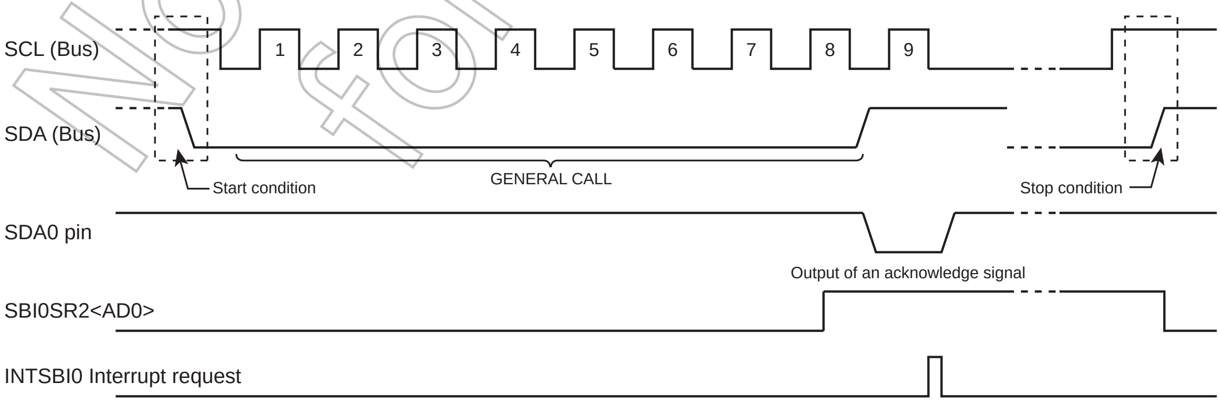


Figure 18-15 Changes in the GENERAL CALL Detection Monitor

18.4.14 Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to SBI0SR2<LRB>.

In the acknowledge mode, immediately after an interrupt request is generated, an acknowledge signal is read by reading the contents of SBI0SR2<LRB>.

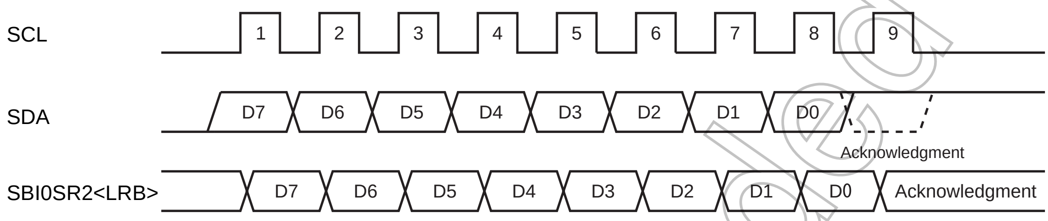


Figure 18-16 Changes in the Last Received Bit Monitor

18.4.15 Slave address and address recognition mode specification

When the serial bus interface circuit is used in the I^2C bus mode, clear I2C0AR<ALS> to "0", and set I2C0AR<SA> to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set I2C0AR<ALS> to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after the start condition.

18.5 Data Transfer of I^{2}C Bus

18.5.1 Device initialization

Set POFFCR1<SBI0EN> to "1".

After confirming that the serial bus interface pin is high level, set SBI0CR2<SBIM> to "1" to select the serial bus interface mode.

Set SBI0CR1<ACK> to "1", SBI0CR1<NOACK> to "0" and SBI0CR1<BC> to "000" to count the number of clocks for an acknowledge signal, to enable the slave address match detection and the GENERAL CALL detection, and set the data length to 8 bits. Set T_{HIGH} and T_{LOW} at SBI0CR1<SCK>.

Set a slave address at I2C0AR<SA> and set I2C0AR<ALS> to "0" to select the I^{2}C bus mode.

Finally, set SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0", SBI0CR2<PIN> to "1" and SBI0CR2<SWRST> to "00" for specifying the default setting to a slave receiver mode.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

Example :Initialize a device

```
CHK_PORT:  LD    A, (P2PRD)          ; Checks whether the serial bus interface pin is at the high level
           AND   A, 0x18
           CMP   A, 0x18
           JR    NZ, CHK_PORT
           LD    (SBI0CR2), 0x18     ; Selects the serial bus interface mode
           LD    (SBI0CR1), 0x16     ; Selects the acknowledgment mode and sets SBI0CR1<SCK> to "110"
           LD    (I2C0AR), 0xa0      ; Sets the slave address to 1010000 and selects the I2C bus mode
           LD    (SBI0CR2), 0x18     ; Selects the slave receiver mode
```

18.5.2 Start condition and slave address generation

Confirm a bus free status (SBI0SR2<BB>="0").

Set SBI0CR1<ACK> to "1" and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

By writing "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBI0DBR are output. The time from generating the START condition until the falling SBI0 pin takes t_{HIGH}.

An interrupt request occurs at the 9th falling edge of a SCL clock cycle, and SBI0CR2<PIN> is cleared to "0". The SCL0 pin is pulled down to the low level while SBI0CR2<PIN> is "0". When an interrupt request occurs, SBI0CR2<TRX> changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: Do not write a slave address to the SBI0DBR while data is transferred. If data is written to the SBI0DBR, data to be output may be destroyed.

Note 2: The bus free state must be confirmed by software within 98.0 μs (the shortest transmitting time according to the standard mode I^{2}C bus standard) or 23.7μs (the shortest transmitting time according to the fast mode I^{2}C bus standard) after setting of the slave address to be output. Only when the bus free state is confirmed, set "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> to generate the start conditions. If the writing of slave address and setting of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> doesn't finish within 98.0μs or 23.7μs, the other masters may start the transferring and the slave address data written in SBI0DBR may be broken.

Example :Generate the start condition

```
CHK_BB:   TEST   (SBI0SR2).BB        ; Confirms that the bus is free
          JR     F, CHK_BB
          LD     (SBI0DBR), 0xcb     ; The transmission slave address 0x65 and the direction bit "1"
          LD     (SBI0CR2), 0xf8     ; Write "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

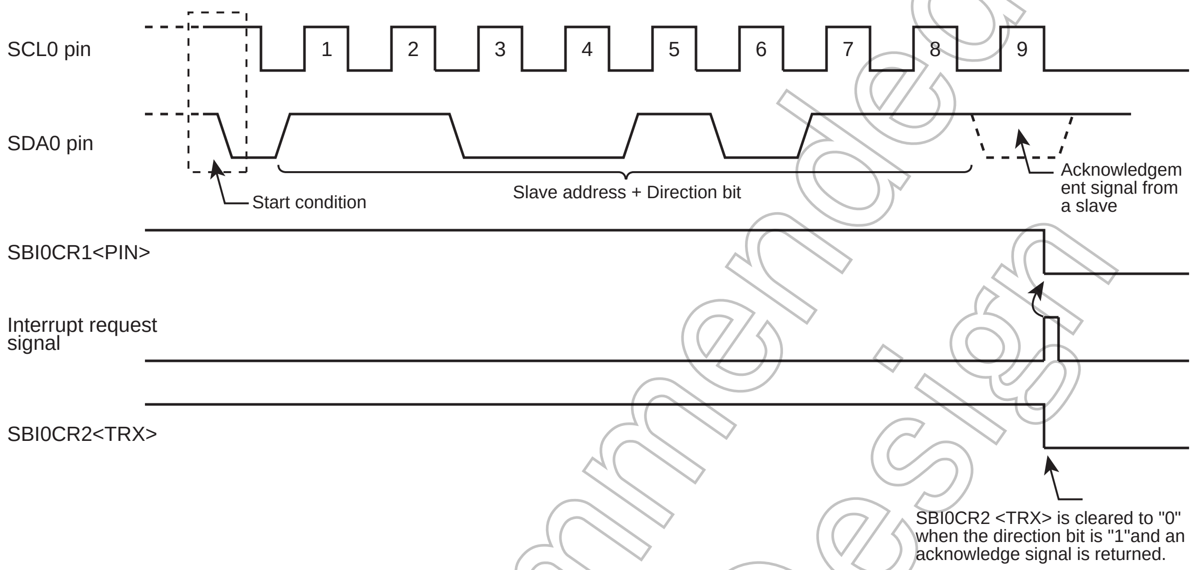


Figure 18-17 Generating the Start Condition and the Slave Address

18.5.3 1-word data transfer

Check SBI0SR2<MST> by the interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

18.5.3.1 When SBI0SR2<MST> is "1" (Master mode)

Check SBI0SR2<TRX> and determine whether the mode is a transmitter or receiver.

(1) When SBI0SR2<TRX> is "1" (Transmitter mode)

Check SBI0SR2<LRB>. When SBI0SR2<LRB> is "1", a receiver does not request data. Implement the process to generate a stop condition (described later) and terminate data transfer.

When SBI0SR2<LRB> is "0", the receiver requests subsequent data. When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again, set SBI0CR1<ACK> to "1", and write the transmitted data to SBI0DBR.

After writing the data, SBI0CR2<PIN> becomes "1", a serial clock pulse is generated for transferring the subsequent 1-word data from the SCL0 pin, and then the 1-word data is transmitted from the SDA0 pin.

After the data is transmitted, an interrupt request occurs. SBI0CR2<PIN> become "0" and the SCL0 pin is set to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the SBI0SR2<LRB> checking above.

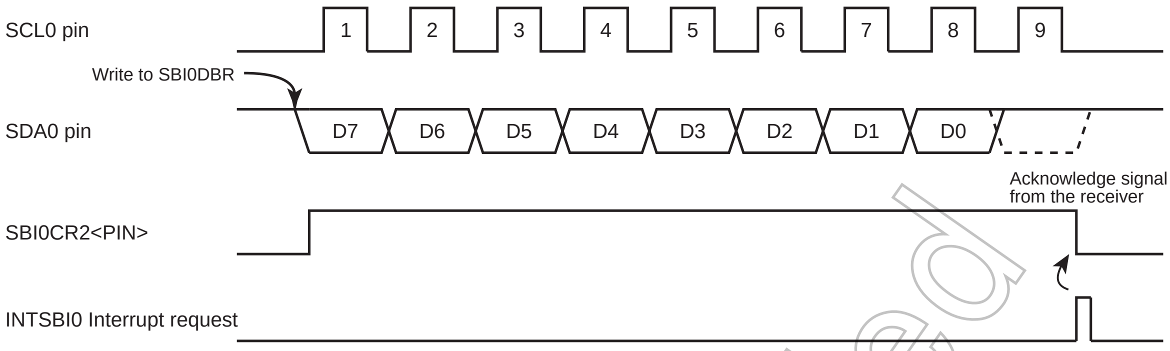


Figure 18-18 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"

(2) When SBI0SR2<TRX> is "0" (Receiver mode)

When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again. Set SBI0CR1< ACK> to "1" and read the received data from the SBI0DBR (Reading data is undefined immediately after a slave address is sent).

After the data is read, SBI0CR2<PIN> becomes "1" by writing the dummy data (0x00) to the SBI0DBR. The serial bus interface circuit outputs a serial clock pulse to the SCL0 pin to transfer the subsequent 1-word data and sets the SDA0 pin to "0" at the acknowledge signal timing.

An interrupt request occurs and SBI0CR2<PIN> becomes "0". Then a serial bus interface circuit outputs a clock pulse for 1-word data transfer and the acknowledge signal by writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" after reading the received data.

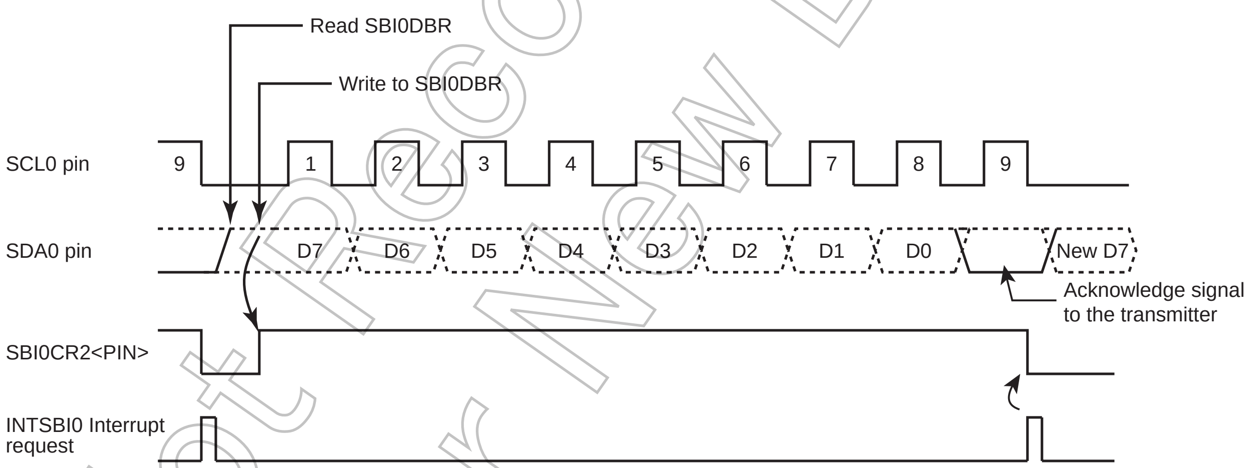


Figure 18-19 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"

To make the transmitter terminate transmission, execute following procedure before receiving a last data.

1. Read the received data.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "000".
3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-word data in which no clock is generated for an acknowledge signal by setting SBI0CR2<PIN> to "1". Next, execute following procedure.

1. Read the received data.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "001".

3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-bit data by setting SBI0CR1<PIN> to "1".

In this case, since the master device is a receiver, the SDA line on a bus keeps the high level. The transmitter receives the high-level signal as a negative acknowledge signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

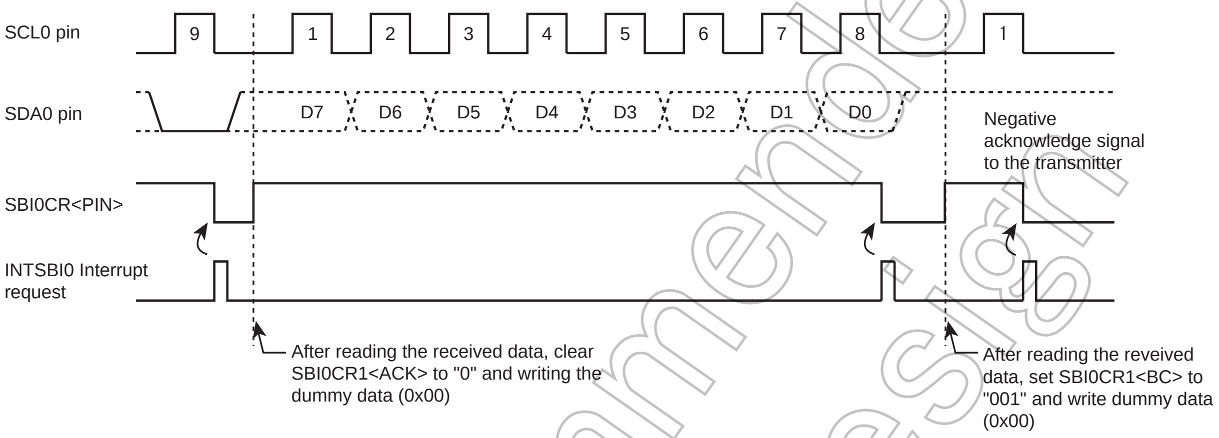


Figure 18-20 Termination of Data Transfer in the Master Receiver Mode

18.5.3.2 When SBI0SR2<MST> is "0" (Slave mode)

In the slave mode, a serial bus interface circuit operates either in the normal slave mode or in the slave mode after losing arbitration.

In the slave mode, the conditions of generating the serial bus interface interrupt request (INTSBI0) are follows:

- At the end of the acknowledge signal when the received slave address matches the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

The serial bus interface circuit changes to the slave mode if arbitration is lost in the master mode. And an interrupt request occurs when the word data transfer terminates after losing arbitration. The generation of the interrupt request and the behavior of SBI0CR2<PIN> after losing arbitration are shown in Table 18-4.

Table 18-4 The Behavior of an interrupt request and SBI0CR2<PIN> After Losing Arbitration

	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as Master Transmitter
interrupt request	An interrupt request is generated at the termination of word-data transfer.	
SBI0CR2<PIN>	SBI0CR2<PIN> is cleared to "0".	

When an interrupt request occurs, SBI0CR2<PIN> is reset to "0", and the SCL0 pin is set to the low level. Either writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" releases the SCL0 pin after taking t_{LOW}.

Check SBI0SR2<AL>, SBI0SR2<TRX>, SBI0SR2<AAS> and SBI0SR2<AD0> and implement processes according to conditions listed in Table 18-5.

Table 18-5 Operation in the Slave Mode

SBI0SR2<TRX>	SBI0SR2<AL>	SBI0SR2<AAS>	SBI0SR2<AD0>	Conditions	Process
1	1	1	0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to the SBI0DBR.
	0	1	0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, the serial bus interface circuit finishes the transmission of 1-word data	Check SBI0SR2<LRB>. If it is set to "1", set SBI0CR2<PIN> to "1" since the receiver does not request subsequent data. Then, clear SBI0CR2<TRX> to "0" to release the bus. If SBI0SR2<LRB> is set to "0", set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to SBI0DBR since the receiver requests subsequent data.
0	1	1	1/0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "0" or receives a "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	0	The serial bus interface circuit loses arbitration when transmitting a slave address or data, and terminates transferring the word data.	The serial bus interface circuit is changed to the slave mode. Write the dummy data (0x00) to the SBI0DBR to clear SBI0SR2<AL> to "0" and set SBI0CR2<PIN> to "1".
	0	1	1/0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "0" or receives "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	1/0	In the slave receiver mode, the serial bus interface circuit terminates the receipt of 1-word data.	Set the number of bits in 1-word to SBI0CR1<BC>, read the received data from the SBI0DBR and write the dummy data (0x00).

Note: In the slave mode, if the slave address set in I2C0AR<SA> is "0x00", a START Byte "0x01" in I2C bus standard is received, the device detects slave address match and SBI0CR2<TRX> is set to "1". Do not set I2C0AR<SA> to "0x00".

18.5.4 Stop condition generation

When SBI0CR2<BB> is "1", a sequence of generating a stop condition is started by setting "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and clearing SBI0CR2<BB> to "0". Do not modify the contents of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> until a stop condition is generated on a bus.

When a SCL line on a bus is pulled down by other devices, a serial bus interface circuit generates a stop condition after a SCL line is released.

The time from the releasing SCL line until the generating the STOP condition takes t_{HIGH}.

Example :Generate the stop condition

```
          LD      (SBI0CR2), 0xD8    ; Sets SBI0CR2<MST>, <TRX> and <PIN> to "1" and SBI0CR2<BB> to "0"
CHK_BB:   TEST    (SBI0SR2).BB       ;Waits until the bus is set free
          JR      T, CHK_BB
```

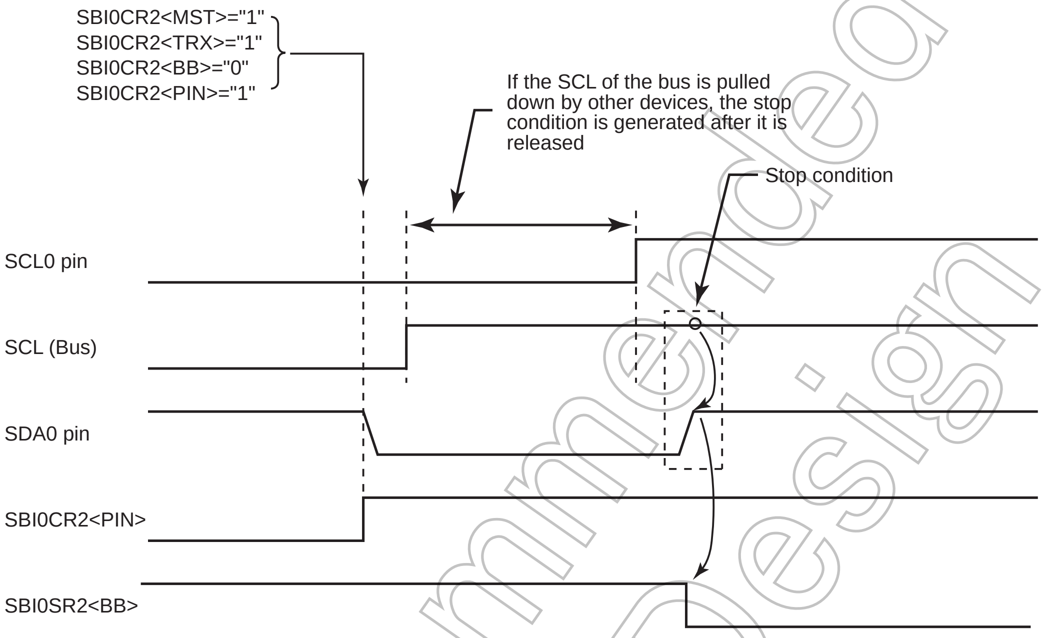


Figure 18-21 Stop Condition Generation

18.5.5 Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart the serial bus interface circuit.

Clear SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0" and set SBI0CR2 <PIN> to "1". The SDA0 pin retains the high level and the SCL0 pin is released.

Since this is not a stop condition, the bus is assumed to be in a busy state from other devices.

Check SBI0SR2<BB> until it becomes "0" to check that the SCL0 pin of the serial bus interface circuit is released.

Check SBI0SR2<LRB> until it becomes "1" to check that the SCL line on the bus is not pulled down to the low level by other devices.

After confirming that the bus stays in a free state, generate a start condition in the procedure "18.5.2 Start condition and slave address generation".

In order to meet the setup time at a restart, take at least 4.7μs of waiting time by the software in the standard mode I^2C bus standard or at least 0.6μs of waiting time in the fast mode I^2C bus standard from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When the master is in the receiver mode, it is necessary to stop the data transmission from the slave device before the STOP condition is generated. To stop the transmission, the master device make the slave device receiving a negative acknowledge. Therefore, SBI0SR2<LRB> is "1" before generating the Restart and it can not be confirmed that SCL line is not pulled down by other devices. Please confirm the SCL line state by reading the port.

Example :Generate a restart

```
          LD     (SBI0CR2), 0x18   ; Sets SBI0CR2<MST>, <TRX> and <BB> to "0" and SBI0CR2<PIN> to "1"
CHK_BB:   TEST   (SBI0SR2).BB      ; Waits until SBI0SR2<BB> becomes "0"
          JR     T, CHK_BB
CHK_LRB:  TEST   (SBI0SR2).LRB     ; Waits until SBI0SR2<LRB> becomes "1"
          JR     F, CHK_LRB
                 .
                 .                 ; Wait time process by the software
                 .
          LD     (SBI0CR2), 0xf8   ; Sets SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

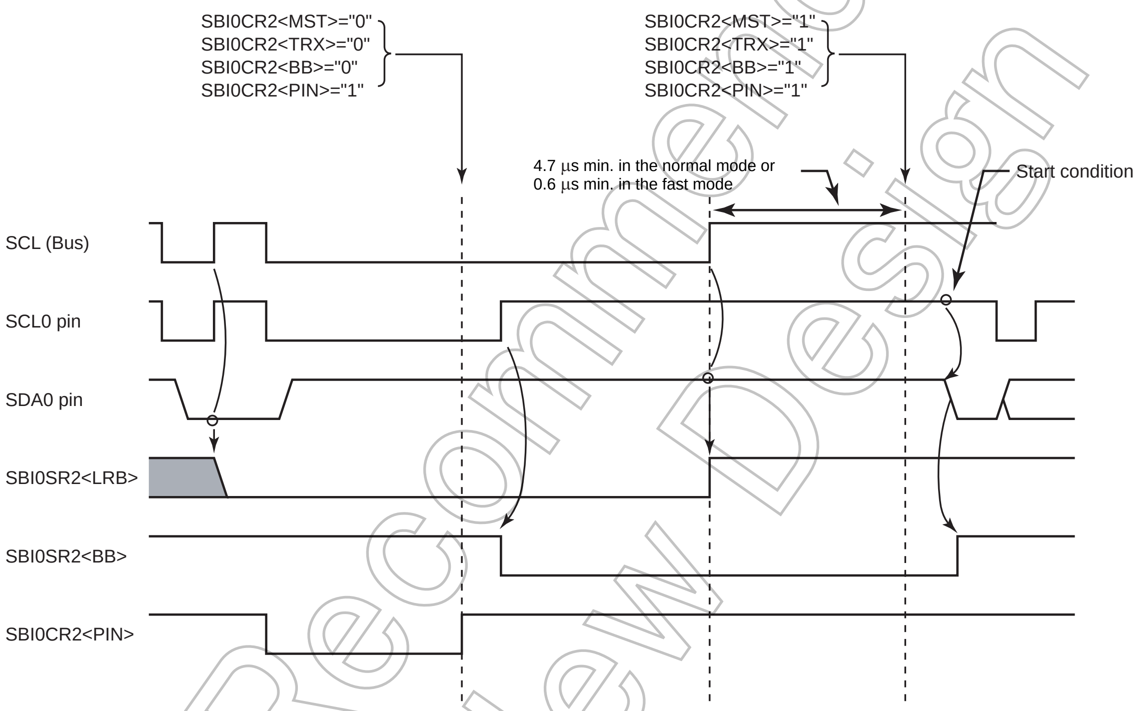


Figure 18-22 Timing Diagram When Restarting

18.6 AC Specifications

The AC specifications are as listed below.

The operating mode (fast or standard) mode should be selected suitable for frequency of fcgck. For these operating mode, refer to the following table.

Table 18-6 AC Specifications (Circuit Output Timing)

Parameter	Symbol	Standard mode		Fast mode		Unit
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	fcgck / (m+n)	0	fcgck / (m+n)	kHz
Hold time (re)start condition. This period is followed by generation of the first clock pulse.	$t_{HD;STA}$	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (output)	t_{LOW}	n / fcgck	-	n / fcgck	-	μs
High-level period of SCL clock (output)	t_{HIGH}	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (input)	t_{LOW}	5 / fcgck	-	5 / fcgck	-	μs
High-level period of SCL clock (input)	t_{HIGH}	3 / fcgck	-	3 / fcgck	-	μs
Restart condition setup time	$t_{SU;STA}$	Depends on the software	-	Depends on the software	-	μs
Data hold time	$t_{HD;DAT}$	0	5 / fcgck	0	5 / fcgck	μs
Data setup time	$t_{SU;DAT}$	250	-	100	-	ns
Rising time of SDA and SCL signals	t_r	-	1000	-	300	ns
Falling time of SDA and SCL signals	t_f	-	300	-	300	ns
Stop condition setup time	$t_{SU;STO}$	m / fcgck	-	m / fcgck	-	μs
Bus free time between the stop condition and the start condition	t_{BUF}	Depends on the software	-	Depends on the software	-	μs
Time before rising of SCL after SBICR2<PIN> is changed from "0" to "1"	$t_{SU;SCL}$	n / fcgck	-	n / fcgck	-	μs

Note: For m and n, refer to"18.4.4.1 Clock source".

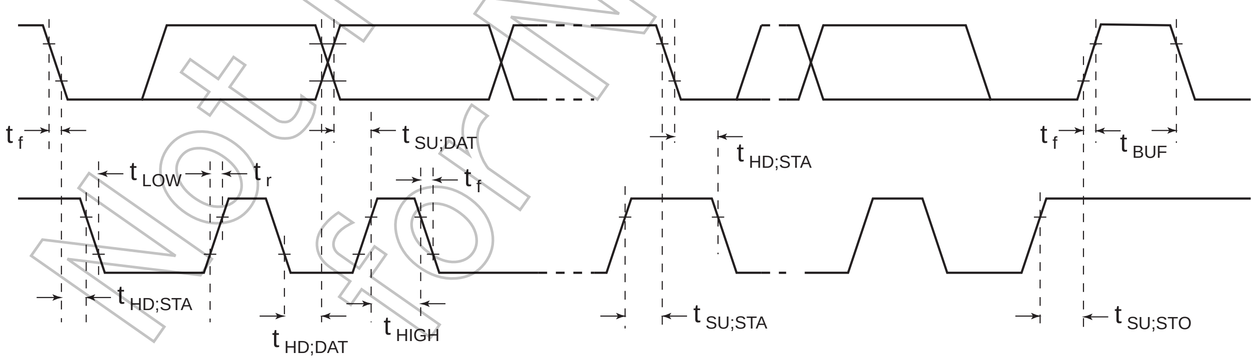


Figure 18-23 Definition of Timing (No. 1)

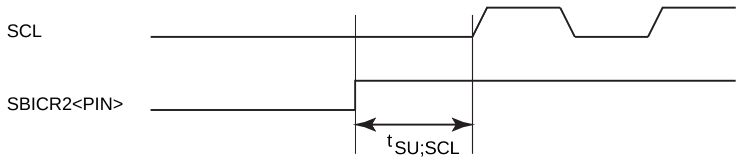


Figure 18-24 Definition of Timing (No. 2)

18.7 Revision History

Rev	Description
RA001	" Serial bus interface control register 1" Revised SCK description. Added Note5.
	"18.6 AC Specifications" Revised fcgck description.
	"Table 18-8 AC Specifications (Circuit Output Timing)" Revised value of "SCL clock frequency".
	Revised from "normal mode" to "standard mode".
RA002	"18.5.1 Device initialization" Revised example of program.

19. Key-on Wakeup (KWU)

The key-on wakeup is a function for releasing the STOP mode at the $\overline{\text{STOP}}$ pin or at pins KWI7 through KWI0.

19.1 Configuration

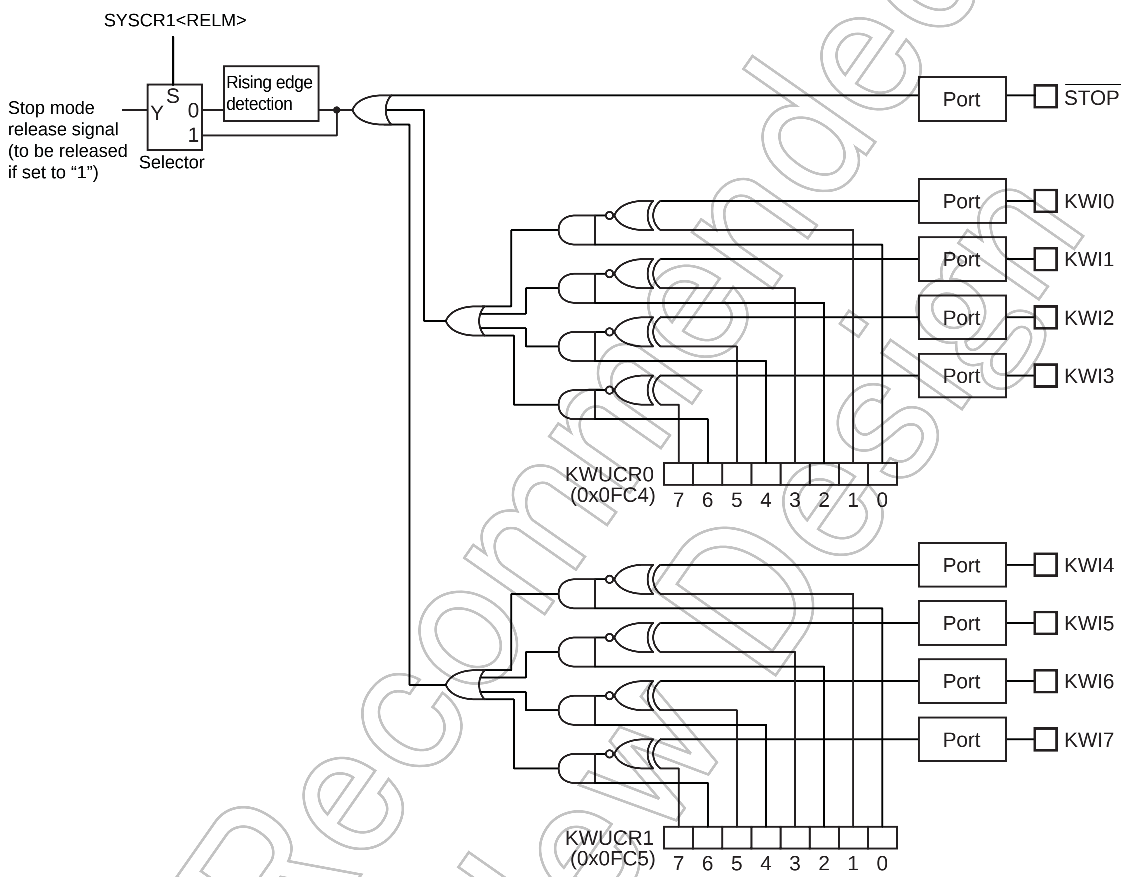


Figure 19-1 Key-on Wakeup Circuit

19.2 Control

Key-on wakeup control registers (KWUCR0 and KWUCR1) can be configured to designate the key-on wakeup pins (KWI7 through KWI0) as STOP mode release pins and to specify the STOP mode release levels of each of these designated pins.

Key-on wakeup control register 0

KWUCR0 (0x0FC4)		7	6	5	4	3	2	1	0
	Bit Symbol	KW3LE	KW3EN	KW2LE	KW2EN	KW1LE	KW1EN	KW0LE	KW0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW3LE	STOP mode release level of KWI3 pin	0: Low level 1: High level
KW3EN	Input enable/disable control of KWI3 pin	0: Disable 1: Enable
KW2LE	STOP mode release level of KWI2 pin	0: Low level 1: High level
KW2EN	Input enable/disable control of KWI2 pin	0: Disable 1: Enable
KW1LE	STOP mode release level of KWI1	0: Low level 1: High level
KW1EN	Input enable/disable control of KWI1 pin	0: Disable 1: Enable
KW0LE	STOP mode release level of KWI0 pin	0: Low level 1: High level
KW0EN	Input enable/disable control of KWI0 pin	0: Disable 1: Enable

Key-on wakeup control register 1

KWUCR1 (0x0FC5)		7	6	5	4	3	2	1	0
	Bit Symbol	KW7LE	KW7EN	KW6LE	KW6EN	KW5LE	KW5EN	KW4LE	KW4EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW7LE	STOP mode release level of KWI7 pin	0: Low level 1: High level
KW7EN	Input enable/disable control of KWI7 pin	0: Disable 1: Enable
KW6LE	STOP mode release level of KWI6 pin	0: Low level 1: High level
KW6EN	Input enable/disable control of KWI6 pin	0: Disable 1: Enable
KW5LE	STOP mode release level of KWI5 pin	0: Low level 1: High level
KW5EN	Input enable/disable control of KWI5 pin	0: Disable 1: Enable
KW4LE	STOP mode release level of KWI4 pin	0: Low level 1: High level
KW4EN	Input enable/disable control of KWI4 pin	0: Disable 1: Enable

19.3 Functions

By using the key-on wakeup function, the STOP mode can be released at a STOP pin or at KWIm pin (m: 0 through 7). After resetting, the STOP pin is the only STOP mode release pin. To designate the KWIm pin as a STOP mode release pin, therefore, it is necessary to configure the key-on wakeup control register (KWUCRn) (n: 0 or 1). Because the STOP pin lacks a function for disabling inputs, it can be designated as a pin for receiving a STOP mode release signal, irrespective of whether the key-on wakeup function is used or not.

- Setting KWUCRn and P4PU registers

  To designate a key-on wakeup pin (KWIm) as a STOP mode release pin, set KWUCRn<KWmEN> to "1". After KWIm pin is set to "1" at KWUCRn<KWmEN>, a specific STOP mode release level can be specified for this pin at KWUCRn<KWmLE>. If KWUCRn<KWmLE> is set to "0", STOP mode is released when an input is at a low level. If it is set to "1", STOP mode is released when an input is at a high level. For example, if you want to release STOP mode by inputting a high-level signal into a KWI0 pin, set KWUCR0<KW0EN> to "1", " and KWUCR0<KW0LE> to "1".

  Each KWIm pin can be connected to internal pull-up resistors. Before connecting to internal pull-up resistors, the corresponding bits in the pull-up control register (P4PU) at port P4 must be set to "1".

- Starting STOP mode

  To start the STOP mode, set SYSCR1<RELM> to "1" (level release mode), and SYSCR1<STOP> to "1".

  To use the key-on wakeup function, do not set SYSCR1<RELM> to "0" (edge release mode). If the key-on wakeup function is used in edge release mode, STOP mode cannot be released, although a rising edge is input into the STOP pin. This is because the KWIm pin enabling inputs to be received is at a release level after the STOP mode starts.

- Releasing STOP mode

  To release STOP mode, input a high-level signal into the STOP pin or input a specific release level into the KWIm pin for which receipt of inputs is enabled. If you want to release STOP mode at the KWIm pin, rather than the STOP pin, continue inputting a low-level signal into the STOP pin throughout the period from when the STOP mode is started to when it is released.

  If the STOP pin or KWIm pin is already at a release level when the STOP mode starts, the following instruction will be executed without starting the STOP mode (with no warm-up performed).

Note 1: If an analog voltage is applied to KWIm pin for which receipt of inputs is enabled by the key-on wakeup control register (KWUCRn) setting, a penetration current will flow. Therefore, in this case, the analog voltage should be not applied to this pin.

Table 19-1 STOP Mode Release Level (edge)

Pin name	Release level (edge)		
	SYSCR1<RELM>="1" (level release mode)		SYSCR1<RELM>="0" (edge release mode)
	KWUCRn<KWmLE>="0"	KWUCRn<KWmLE>="1"	
STOP	"H" level		Rising edge
KWIm	"L" level	"H" level	Don't use

Example :A case in which STOP mode is started with the release level of the $\overline{\text{STOP}}$ pin set to a high level and the release level of KWI0 set to a low level (connected to an internal pull-up resistor of the KWI0 pin)

```
DI                                      ; IMF←0
SET    (P4PU).0                         ; KWI0 (P40) connected to a pull-up resistor
LD     (KWUCR0), 0y00000001             ; the KWI0 pin is set to enable inputs, and its release level is set
                                        ; to a low level.
LD     (SYSCR1), 0y10100000             ; Starting in level release mode
```

20. 10-bit AD Converter (ADC)

The TMP89CM42 has a 10-bit successive approximation type AD converter.

20.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 20-1.

It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDRL and ADCDRH, a DA converter, a sample-hold circuit, a comparator, a successive comparison circuit, etc.

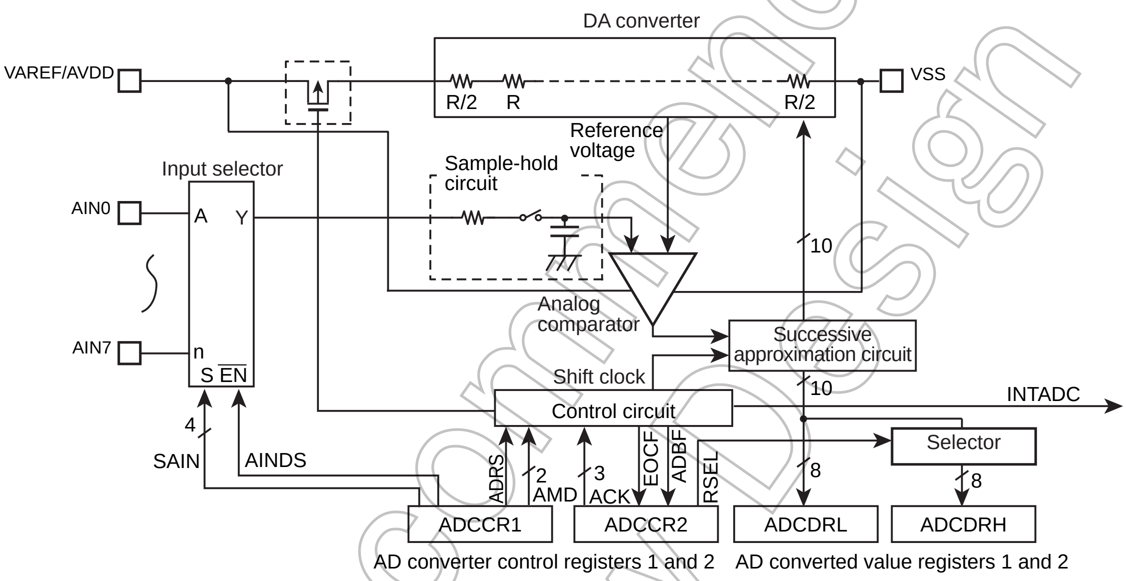


Figure 20-1 10-bit AD Converter

Note 1: Before using the AD converter, set an appropriate value to the I/O port register which is also used as an analog input port. For details, see the section on "I/O ports".

Note 2: The DA converter current (IREF) is automatically cut off at times other than during AD conversion.

20.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

   This register selects an analog channel in which to perform AD conversion, selects an AD conversion operation mode, and controls the start of the AD converter.

2. AD converter control register 2 (ADCCR2)

   This register selects the AD conversion time, and monitors the operating status of the AD converter.

3. AD converted value registers (ADCDRH and ADCDRL)

   These registers store the digital values generated by the AD converter.

AD converter control register 1

ADCCR1 (0x0034)	7	6	5	4	3	2	1	0
Bit Symbol	ADRS	AMD		AINEN	SAIN			
Read/Write	R/W	R/W		R/W	R/W			
After reset	0	0	0	0	0	0	0	0

ADRS	AD conversion start	0: 1:	- AD conversion start
AMD	AD operating mode	00: 01: 10: 11:	AD operation disable, forcibly stop AD operation Single mode Reserved Repeat mode
AINEN	Analog input control	0: 1:	Analog input disable Analog input enable
SAIN	Analog input channel select	0000: 0001: 0010: 0011: 0100: 0101: 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111:	AIN0 AIN1 AIN2 AIN3 AIN4 AIN5 AIN6 AIN7 Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved

Note 1: Do not perform the following operations on the ADCCR1 register while AD conversion is being executed (ADCCR2<ADBF>="1").
- Changing SAIN
- Setting AINEN to "0"
- Changing AMD (except a forced stop by setting AMD to "00")
- Setting ADRS to "1"

Note 2: If you want to disable all analog input channels, set AINEN to "0".

Note 3: Although analog input pins are also used as input/output ports, it is recommended for the purpose of maintaining the accuracy of AD conversion that you do not execute input/output instructions during AD conversion. Additionally, do not input widely varying signals into the ports adjacent to analog input pins.

Note 4: When STOP, IDLE0 or SLOW mode is started, ADRS, AMD and AINEN are initialized to "0". If you use the AD converter after returning to NORMAL mode, you must reconfigure ADRS, AMD and AINEN.

Note 5: After the start of AD conversion, ADRS is automatically cleared to "0" ("0" is read).

AD converter control register 2

ADCCR2 (0x0035)	7	6	5	4	3	2	1	0
Bit Symbol	EOCF	ADBF	-	-	"0"	ACK		
Read/Write	R	R	R	R	W	R/W		
After reset	0	0	0	0	0	0	0	0

EOCF	AD conversion end flag	0: 1:	Before conversion or during conversion Conversion end
ADBF	AD conversion BUSY flag	0: 1:	AD conversion being halted AD conversion being executed
ACK	AD conversion time select (examples of AD conversion time are shown in the table below)	000: 001: 010: 011: 100: 101: 110: 111:	39/fcgck 78/fcgck 156/fcgck 312/fcgck 624/fcgck 1248/fcgck Reserved Reserved

Note 1: Make sure that you make the ACK setting when AD conversion is in a halt condition (ADCCR2<ADBF>="0").

Note 2: Make sure that you write "0" to bit 3 of ADCCR2.

Note 3: If STOP, IDLE0 or SLOW mode is started, EOCF and ADBF are initialized to "0".

Note 4: If the AD converted value register (ADCDRH) is read, EOCF is cleared to "0". It is also cleared to "0" if AD conversion is started (ADCCR1<ADRS>="1") without reading ADCDRH after completing AD conversion in single mode.

Note 5: If an instruction to read ADCCR2 is executed, 0 is read from bits 3 through 5.

Table 20-1 ACK Settings and Conversion Times Relative to Frequencies

		Frequency (fcgck)								
ACK setting	Conversion time	10MHz	8MHz	5MHz	4MHz	2.5MHz	2MHz	1MHz	0.5MHz	0.25 MHz
000	39/fcgck	-	-	-	-	15.6 μs	19.5 μs	39.0 μs	78.0 μs	156.0 μs
001	78/fcgck	-	-	15.6 μs	19.5 μs	31.2 μs	39.0 μs	78.0 μs	156.0 μs	-
010	156/fcgck	15.6 μs	19.5 μs	31.2 μs	39.0 μs	62.4 μs	78.0 μs	156.0 μs	-	-
011	312/fcgck	31.2 μs	39.0 μs	62.4 μs	78.0 μs	124.8 μs	156.0 μs	-	-	-
100	624/fcgck	62.4 μs	78.0 μs	124.8 μs	156.0 μs	-	-	-	-	-
101	1248/fcgck	124.8 μs	156.0 μs	-	-	-	-	-	-	-
11*	Reserved									

Note 1: Spaces indicated by "-" in the above table mean that it is prohibited to establish conversion times in these spaces.
fcgck: High Frequency oscillation clock [Hz]

Note 2: Above conversion times do not include the time shown below.
- Time from when ADCCR1<ADRS> is set to 1 to when AD conversion is started
- Time from when AD conversion is finished to when a converted value is stored in ADCDRL and ADCDRH.
If ACK = 00*, the longest conversion time is 10/fcgck (s). If ACK = 01*, it is 32/fcgck (s). If ACK = 10*, it is 128/fcgck(s).

Note 3: The conversion time must be longer than the following time by analog reference voltage (VAREF).
- VAREF = 4.5 to 5.5 V 15.6 μs or longer
- VAREF = 2.7 to 5.5 V 31.2 μs or longer
- VAREF = 2.2 to 5.5 V 124.8 μs or longer

AD converted value register (lower side)

ADCDRL (0x0036)		7	6	5	4	3	2	1	0
	Bit Symbol	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

AD converted value register (upper side)

ADCDRH (0x0037)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	-	AD09	AD08
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

Note 1: A read of ADCDRL or ADCDRH must be read after the INTADC interrupt is generated or after ADCCR2<EOCF> becomes "1".

Note 2: In single mode, do not read ADCDRL or ADCDRH during AD conversion (ADCCR2<ADBF>="1"). (If AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the INTADC interrupt request is canceled, and the conversion result is lost.)

Note 3: If STOP, IDLE0 or SLOW mode is started, ADCDRL and ADCDRH are initialized to "0".

Note 4: If ADCCR1<AMD> is set to "00", ADCDRL and ADCDRH are initialized to "0".

Note 5: If an instruction to read ADCDRH is executed, "0" is read from bits 7 through 2.

Note 6: If AD conversion is finished in repeat mode in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value register. In this case, the INTADC interrupt request is canceled, and the conversion result is lost.

20.3 Functions

The 10-bit AD converter operates in either single mode in which AD conversion is performed only once or repeat mode in which AD conversion is performed repeatedly.

20.3.1 Single mode

In single mode, the voltage at a designated analog input pin is AD converted only once.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "01" allows AD conversion to start. ADCCR1<ADRS> is automatically cleared after the start of AD conversion. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" if AD conversion is finished or if AD conversion is forced to stop.

After AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. The AD converted value registers (ADCDRL and ADCDRH) should be usually read according to the INTADC interrupt processing routine. If the upper side (ADCDRH) of the AD converted value register is read, ADCCR2<EOCF> is cleared to "0".

Note: Do not perform the following operations on the ADCCR1 register when AD conversion is being executed (ADCCR2<ADBF>="1"). If the following operations are performed, there is the possibility that AD conversion may not be executed properly.

- Changing the ADCCR1<SAIN> setting
- Setting ADCCR1<AINEN> to "0"
- Changing the ADCCR1<AMD> setting (except a forced stop by setting AMD to "00")
- Setting ADCCR1<ADRS> to "1"

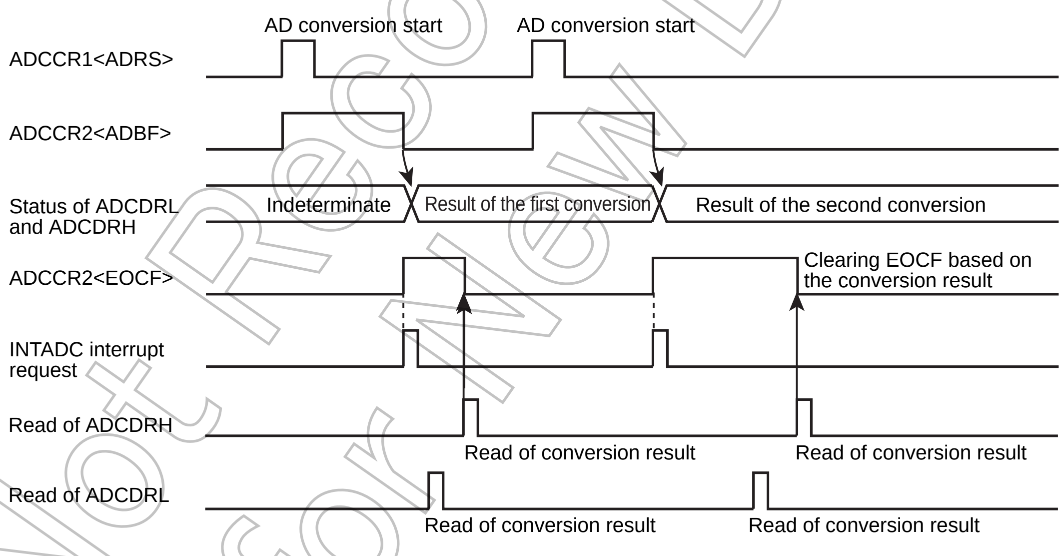


Figure 20-2 Single Mode

20.3.2 Repeat mode

In repeat mode, the voltage at an analog input pin designated at ADCCR1<SAIN> is AD converted repeatedly.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "11" allows AD conversion to start. After the start of AD conversion, ADCCR1<ADRS> is automatically cleared. After the first AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. After this interrupt is generated, the second (next) AD conversion starts immediately.

The AD converted value registers (ADCDRL and ADDRH) should be read before the next AD conversion is finished. If the next AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value registers (ADCDRL and ADCDRH). In this case, the INTADC interrupt request is not generated, and the conversion result is lost. (See Figure 20-3.)

To stop AD conversion, write "00" (AD operation disable) to ADCCR1<AMD>. As "00" is written to ADCCR1<AMD>, AD conversion stops immediately. In this case, the converted value is not stored in the AD converted value register. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" if "00" is written to AMD.

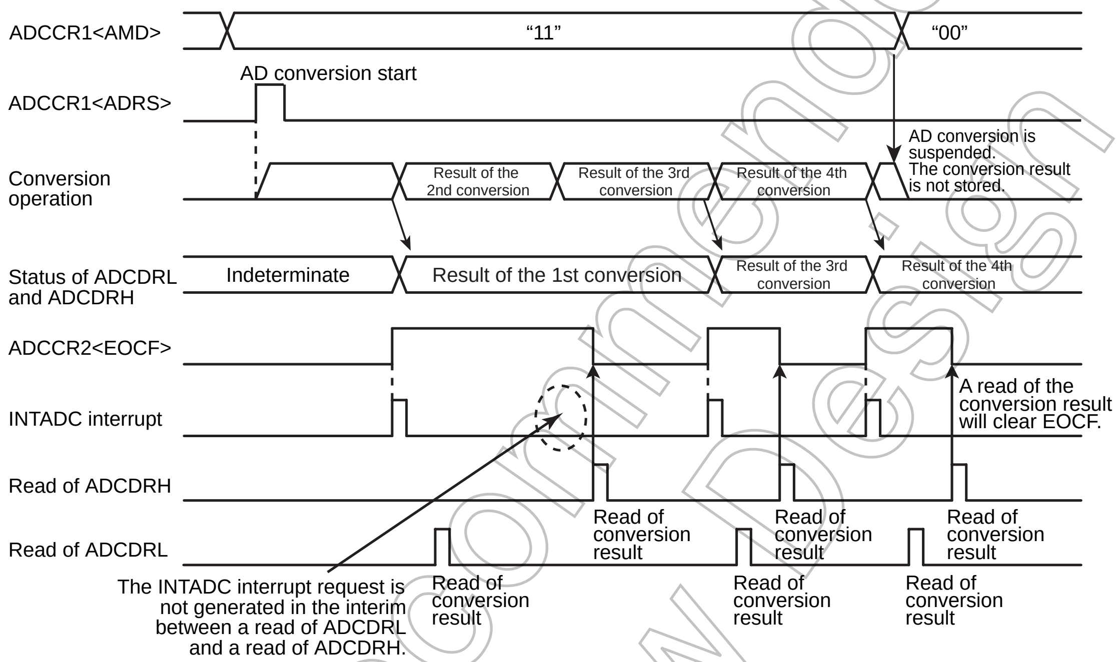


Figure 20-3 Repeat Mode

20.3.3 AD operation disable and forced stop of AD operation

If you want to force the AD converter to stop when AD conversion is ongoing in single mode or if you want to stop the AD converter when AD conversion is ongoing in repeat mode, set ADCCR1<AMD> to "00".

If ADCCR1<AMD> is set to "00", registers ADCCR2<EOCF>, ADCCR2<ADBF>, ADCDRL, and ADCDRH are initialized to "0".

20.4 Register Setting

1. Set the AD converter control register 1 (ADCCR1) as described below:
   - From the AD input channel select (SAIN), select the channel in which AD conversion is to be performed.
   - Set the analog input control (AINEN) to "Analog input enable".
   - At AMD, specify the AD operating mode (single or repeat mode).
2. Set the AD converter control register 2 (ADCCR2) as described below:
   - At the AD conversion time (ACK), specify the AD conversion time. For information on how to specify the conversion time, refer to the AD converter control register 2 and Table 20-1.
3. After the above two steps are completed, set "1" on the AD conversion start (ADRS) of the AD converter control register 1 (ADCCR1), and AD conversion starts immediately if single mode is selected.
4. As AD conversion is finished, the AD conversion end flag (EOCF) of the AD converter control register 2 (ADCCR2) is set to "1", the AD conversion result is stored in the AD converted value registers (ADCDRH and ADCDRL), and the INTADC interrupt request is generated.
5. After the conversion result is read from the AD converted value register (ADCDRH), EOCF is cleared to "0". EOCF will also be cleared to "0" if AD conversion is performed once again before reading the AD converted value register (ADCDRH). In this case, the previous conversion result is retained until AD conversion is finished.

Example: After selecting the conversion time 15.6 μs at 10 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, store the conversion result in the HL register. The operation mode is single mode.

```
                 : (Port setting)                          ;Before setting AD converter registers, make an appropriate port
                                                           ;register setting.(For further details, refer to the section that describes
                                                           ;I/O ports.)
                 LD      (ADCCR1), 0y00110011              ;Select AIN3 and operation mode
                 LD      (ADCCR2), 0y00000010              ;Select conversion time (156/fcgck)
                 SET     (ADCCR1). 7                       ;ADRS = 1 (AD conversion start)
SLOOP :          TEST    (ADCCR2). 7                       ;EOCF = 1 ?
                 J       T, SLOOP
                 LD      HL, (ADCDRL)                      ;Read result data
```

20.5 Starting STOP/IDLE0/SLOW Modes

If STOP/IDLE0/SLOW mode is started, registers ADCCR1<ADRS, AMD, AINEN>, ADCCR2<EOCF, ADBF>, ADCDRL and ADCDRH are initialized to "0". If any of these modes is started during AD conversion, AD conversion is suspended, and the AD converter stops (registers are likewise initialized). When restored from STOP/ IDLE0/ SLOW mode, AD conversion is not automatically restarted. Therefore, registers must be reconfigured as necessary.

If STOP/IDLE0/SLOW mode is started during AD conversion, analog reference voltage is automatically disconnected and, therefore, there is no possibility of current flowing into the analog reference voltage.

20.6 Analog Input Voltage and AD Conversion Result

Analog input voltages correspond to AD-converted, 10-bit digital values, as shown in Figure 20-4.

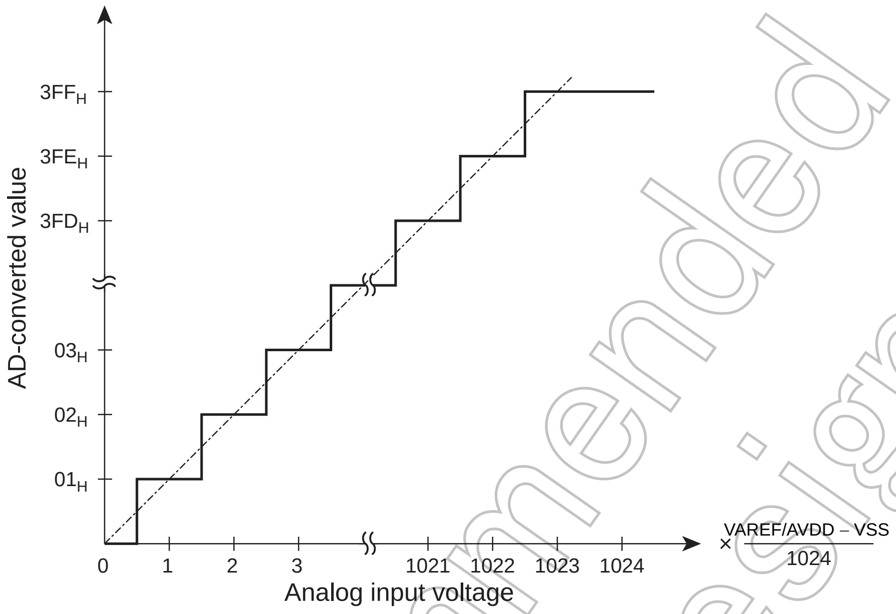


Figure 20-4 Relationships between Analog Input Voltages and AD-converted Values (typical values)

20.7 Precautions about the AD Converter

20.7.1 Analog input pin voltage range

Analog input pins (AIN0 through AIN7) should be used at voltages from VAREF to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain, and converted values on other pins will also be affected.

20.7.2 Analog input pins used as input/output ports

Analog input pins (AIN0 to AIN7) are also used as input/output ports. In using one of analog input pins (ports) to execute AD conversion, input/output instructions at all other pins (ports) must not be executed. If they are executed, there is the possibility that the accuracy of AD conversion may deteriorate. This also applies to pins other than analog input pins; if one pin receives inputs or generates outputs, noise may occur and its adjacent pins may be affected by that noise.

20.7.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 20-5. The higher the output impedance of the analog input source, the more susceptible it becomes to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. It is recommended that a capacitor be attached externally.

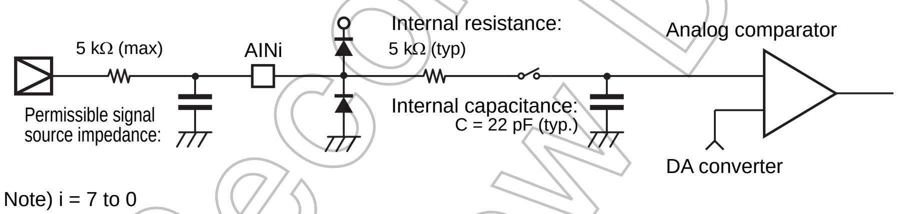


Figure 20-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

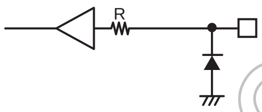
20.8 Revision History

Rev	Description
RA002	"20.4 Register Setting" Revised ADCCR2 value and comment of example program.

21. Input/Output Circuit

21.1 Control Pins

The input/output circuitries of the TMP89CM42 control pins are shown below.

Control pin	I/O	Circuitry	Remarks
XIN XOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
XTIN XTOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
$\overline{\text{RESET}}$	Input	Refer to the P1 ports in the chapter of Input/Output Ports.	
MODE	Input		R = 100 Ω (typ.)

22. Electrical Characteristics

22.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0 V)

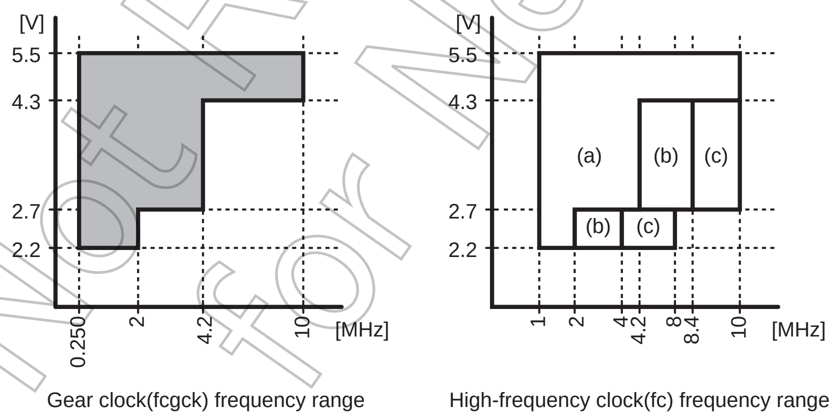
Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to 6.0	V
Input voltage	V_{IN1}	P0, P1, P2 (excluding P23 and P24), P4, P7, P8, P9, PB (tri-state port)	−0.3 to V_{DD} + 0.3	V
	V_{IN2}	P23, P24 (sink open drain port)	−0.3 to V_{DD} + 0.3	
	V_{IN3}	AIN0 to AIN7 (analog input voltage)	−0.3 to A_{VDD} + 0.3	
Output voltage	V_{OUT1}		−0.3 to V_{DD} + 0.3	V
Output current (per pin)	I_{OUT1}	P0, P1, P2 (excluding P23 and P24), P4, P7, P8, P9, PB (tri-state port)	−1.8	mA
	I_{OUT2}	P0, P1, P2, P4, P9 (pull-up resistor)	−0.4	
	I_{OUT3}	P0, P1, P2, P4, P74 to P77, P8, P9 (tri-state port)	3.2	
	I_{OUT4}	P70 to P73, PB (large current port)	30	
Output current (total)	ΣI_{OUT1}	P0, P1, P2 (excluding P23 and P24), P4, P7, P8, P9, PB (tri-state port)	−30	
	ΣI_{OUT2}	P0, P1, P2, P4, P9 (pull-up resistor)	−4	
	ΣI_{OUT3}	P0, P1, P2, P4, P74 to P77, P8, P9 (tri-state port)	60	
	ΣI_{OUT4}	P70 to P73, PB (large current port)	120	
Power dissipation (Topr = 85°C)	P_D		250	mW
Soldering temperature (time)	Tsld		260 (10 s)	°C
Storage temperature	Tstg		−55 to 125	
Operating temperature	Topr		−40 to 85	

22.2 Operating Conditions

The operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the operating conditions for the device are always adhered to.

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 10.0 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	2.7	5.5	V
			fc = 8.0 MHz		2.2		
			fcgck = 10.0 MHz		4.3		
			fcgck = 4.2 MHz		2.7		
			fcgck = 2.0 MHz		2.2		
			fs = 32.768 kHz	SLOW1, 2 modes SLEEP0, 1 modes			
			STOP mode				
Input high level	V_{IH1}	MODE pin	V_{DD} ≥ 4.5 V		V_{DD} × 0.70	V_{DD}	V
	V_{IH2}	Hysteresis input			V_{DD} × 0.75		
	V_{IH3}		V_{DD} < 4.5 V		V_{DD} × 0.90		
Input low level	V_{IL1}	MODE pin	V_{DD} ≥ 4.5 V		0	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input				V_{DD} × 0.25	
	V_{IL3}		V_{DD} < 4.5 V			V_{DD} × 0.10	
Clock frequency	fc	XIN, XOUT	V_{DD} = 2.2 to 5.5 V		1.0	8.0	MHz
			V_{DD} = 2.7 to 5.5 V		1.0	10.0	
	fcgck		V_{DD} = 2.2 to 5.5 V		0.25	2.0	
			V_{DD} = 2.7 to 5.5 V			4.2	
			V_{DD} = 4.3 to 5.5 V			10.0	
	fs	XTIN, XTOUT	V_{DD} = 2.2 to 5.5 V		30.0	34.0	kHz



(a) fc, fc/2 or fc/4 can be used as gear clock (fcgck).

(b) Only fc/2 or fc/4 can be used as gear clock (fcgck).

(c) Only fc/4 can be used as gear clock (fcgck).

Figure 22-1 Clock gear (fcgck) and High-frequency clock (fc)

22.3 DC Characteristics

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input	V_{DD} = 5.5 V V_{IN} = V_{MODE} = 5.5 V/0 V	–	0.9	–	V
Input current	I_{IN1}	MODE		–	–	±2	μA
	I_{IN2}	P0, P1, P2, P4, P5, P7, P8, P9, PB					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN2}	$\overline{RESET}$ pull-up	V_{DD} = 5.5 V, V_{IN} = V_{MODE} = 0 V	100	220	500	kΩ
	R_{IN3}	P0, P1, P2 (excluding P23 and P24), P4, P9 pull-up		30	50	100	
Output leakage current	I_{LO1}	P23, P24 (skin open drain port)	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
	I_{LO2}	P0, P1, P2 (excluding P23 and P24), P4, P5, P7, P8, P9, PB (tri-state port)	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	–	–	±2	
Output high voltage	V_{OH}	Except P23, P24, XOUT, XTOUT	V_{DD} = 4.5 V, I_{OH} = −0.7 mA	4.1	–	–	V
Output low voltage	V_{OL}	Except XOUT, XTOUT	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	
Output low current	I_{OL}	P70 to P73, PB (Large current port)	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	mA

Note 1: Typical values show those at Topr = 25°C and V_{DD} = 5.0 V.

Note 2: Input current I_{IN3} : The current through pull-up resistor is not included.

Note 3: V_{IN} : The input voltage on the pin except MODE pin, V_{MODE} : The input voltage on the MODE pin

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Supply current in NORMAL 1, 2 modes	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fcgck = 10.0 MHz fs = 32.768 kHz	–	8.3	10.5	mA
Supply current in IDLE0, 1, 2 modes				–	5.2	7.5	
Supply current in SLOW1 mode			V_{DD} = 3.0 V V_{IN} = 2.8 V/0.2 V V_{MODE}=2.8V/0.1V fs = 32.768 kHz	–	11	22	μA
Supply current in SLEEP1 mode				–	10	21	
Supply current in SLEEP0 mode				–	9	20	
Supply current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V V_{MODE}=5.3V/0.1V	–	8	20	

Note 1: Typical values shown are Topr = 25°C and V_{DD} = 5.0 V, unless otherwise specified.

Note 2: I_{DD} does not include I_{REF}. It is the electrical current in the state in which the peripheral circuitry has been operated.

Note 3: Each supply current in SLOW2 mode is equivalent to that in IDLE0, IDLE1 and IDLE2 modes.

22.4 AD Conversion Characteristics

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max		Unit
Analog reference voltage / Power supply voltage of analog control circuit	V_{AREF} / A_{VDD}		V_{DD}				V
Analog input voltage range	V_{AIN}		V_{SS}	–	V_{AREF}		V
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} / V_{AREF} = 5.5 V V_{SS} = 0.0 V	–	0.6	1.0		mA
					89CM42 89CH42	89FM42 89FH42	
Non-linearity error (Note4)		V_{DD} = A_{VDD} / V_{AREF} = 5.0 V V_{SS} = 0.0V	–	–	±4	±3	LSB
Zero point error (Note4)			–	–	±4	±3	LSB
Full scale error (Note4)			–	–	±4	±3	LSB
Total error (Note4)			–	–	±4	±3	LSB

(V_{SS} = 0.0 V, 2.7 V ≤ V_{DD} < 4.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max		Unit
Analog reference voltage / Power supply voltage of analog control circuit	V_{AREF} / A_{VDD}		V_{DD}				V
Analog input voltage range	V_{AIN}		V_{SS}	–	V_{AREF}		V
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} / V_{AREF} = 4.5 V V_{SS} = 0.0 V	–	0.5	0.8		mA
					89CM42 89CH42	89FM42 89FH42	
Non-linearity error (Note4)		V_{DD} = A_{VDD} / V_{AREF} = 2.7 V V_{SS} = 0.0V	–	–	±4	±3	LSB
Zero point error (Note4)			–	–	±4	±3	LSB
Full scale error (Note4)			–	–	±4	±3	LSB
Total error (Note4)			–	–	±4	±3	LSB

(V_{SS} = 0.0 V, 2.2 V ≤ V_{DD} < 2.7 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max		Unit
Analog reference voltage / Power supply voltage of analog control circuit	V_{AREF} / A_{VDD}		V_{DD}				V
Analog input voltage range	V_{AIN}		V_{SS}	–	V_{AREF}		V
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} / V_{AREF} = 2.7 V V_{SS} = 0.0 V	–	0.3	0.5		mA
					89CM42 89CH42	89FM42 89FH42	
Non-linearity error (Note4)		V_{DD} = A_{VDD} / V_{AREF} = 2.2 V, V_{SS} = 0.0 V	–	–	±5	±4	LSB
Zero point error (Note4)			–	–	±5	±4	LSB
Full scale error (Note4)			–	–	±5	±4	LSB
Total error (Note4)			–	–	±5	±4	LSB

Note 1: The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2: Conversion times differ with variation in the power supply voltage.

Note 3: The voltage to be input to the AIN input pin must be within the range V_{AREF} to V_{SS}. If a voltage outside this range is input, converted values will become indeterminate, and converted values of other channels will be affected.

Note 4: AD conversion characteristics differ between TMP89FM42/FH42 and TMP89CM42/CH42.

Note 5: If the AD converter is not used, fix the V_{AREF}/A_{VDD} pin to the V_{DD} level.

22.5 Power-on Reset Circuit Characteristics

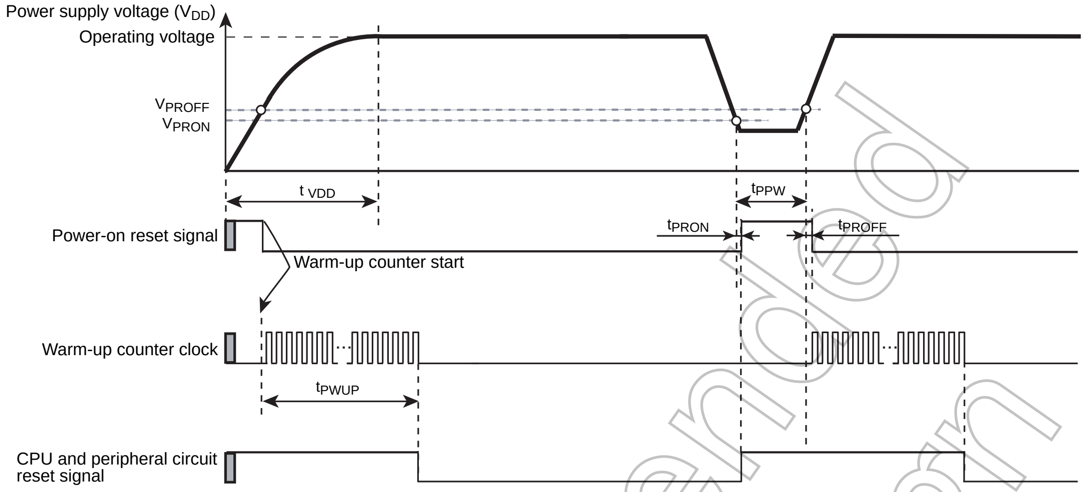


Figure 22-2 Power-on Reset Operation Timing

Note: Care must be taken in system designing since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (V_{DD}).

(V_{SS}=0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{PROFF}	Power-on reset releasing voltage[Note]	1.85	2.02	2.19	V
V_{PRON}	Power-on reset detecting voltage[Note]	1.70	1.85	2.00	
t_{PROFF}	Power-on reset releasing response time	−	0.01	0.1	ms
t_{PRON}	Power-on reset detecting response time	−	0.01	0.1	
t_{PRW}	Power-on reset minimum pulse width	1.0	−	−	
t_{PWUP}	Warming-up time after a reset is cleared	−	102×2^9/fc	−	s
t_{VDD}	Power supply rise time	−	−	5	ms

Note 1: Because the power-on reset releasing voltage and the power-on reset detecting voltage change relative to one another, the detected voltage will never become inverted.

Note 2: A clock output by an oscillating circuit is used as the input clock for a warming-up counter. Because the oscillation frequency does not stabilize until an oscillating circuit stabilizes, some errors may be included in the warming-up time.

Note 3: Boost the power supply voltage such that t_{VDD} becomes smaller that t_{PWUP}.

22.6 Voltage Detecting Circuit Characteristics

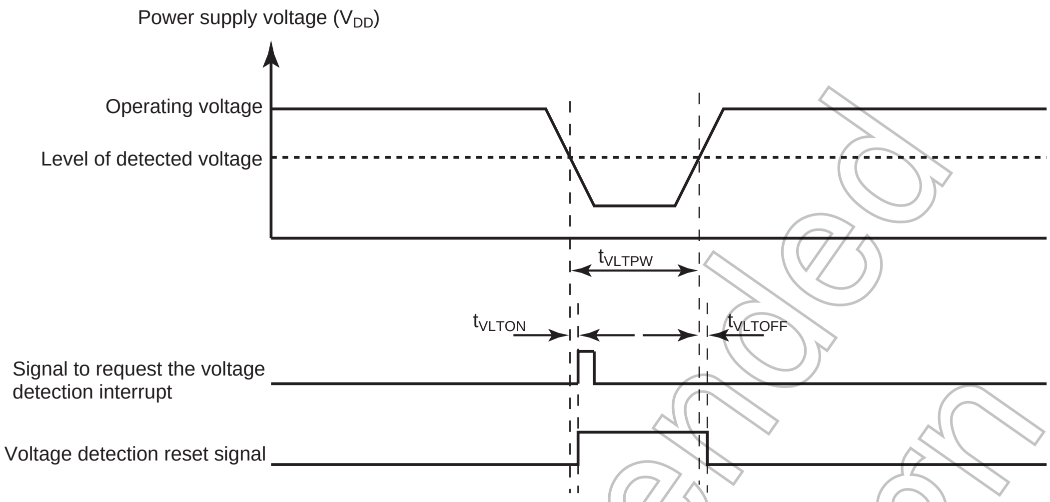


Figure 22-3 Operation Timing of the Voltage Detecting Circuit

Note: Care must be taken in system designing since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (V_{DD}).

(V_{SS} = 0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{VLTOFF}	Voltage detection releasing response time	–	0.01	0.1	ms
t_{VLTON}	Voltage detecting detection response time	–	0.01	0.1	
t_{VLTPW}	Voltage detecting minimum pulse width	1.0	–	–	

22.7 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.3 V to 5.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.100	−	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	−	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input). fc = 10.0 MHz	−	50.0	−	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	−	15.26	−	μs
Low-level clock pulse width	t_{WSL}					

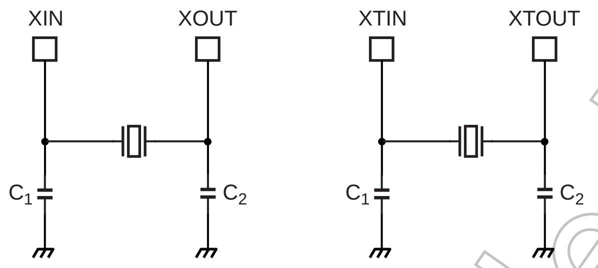
(V_{SS} = 0 V, V_{DD} = 2.7 V to 4.3 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.238	−	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	−	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input). fc = 10.0 MHz	−	50.0	−	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	−	15.26	−	μs
Low-level clock pulse width	t_{WSL}					

(V_{SS} = 0 V, V_{DD} = 2.2 V to 2.7 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.500	−	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	−	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input). fc = 8.0 MHz	−	62.5	−	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	−	15.26	−	μs
Low-level clock pulse width	t_{WSL}					

22.8 Oscillating Condition



(1) High-frequency oscillation (2) Low-frequency oscillation

Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: The product numbers and specifications of the resonators supplied by Murata Manufacturing Co., Ltd. are subject to change.

For up to date information, please refer to the following

http://www.murata.com

22.9 Handling Precaution

- The solderability test conditions are shown below.
  1. When using the Sn-37Pb solder bath

     Solder bath temperature = 230°C

     Dipping time = 5 seconds

     Number of times = once

     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu solder bath

     Solder bath temperature = 245°C

     Dipping time = 5 seconds

     Number of times = once

     R-type flux used

  The pass criteron of the above test is as follows: Solderability rate until forming ≥ 95%
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

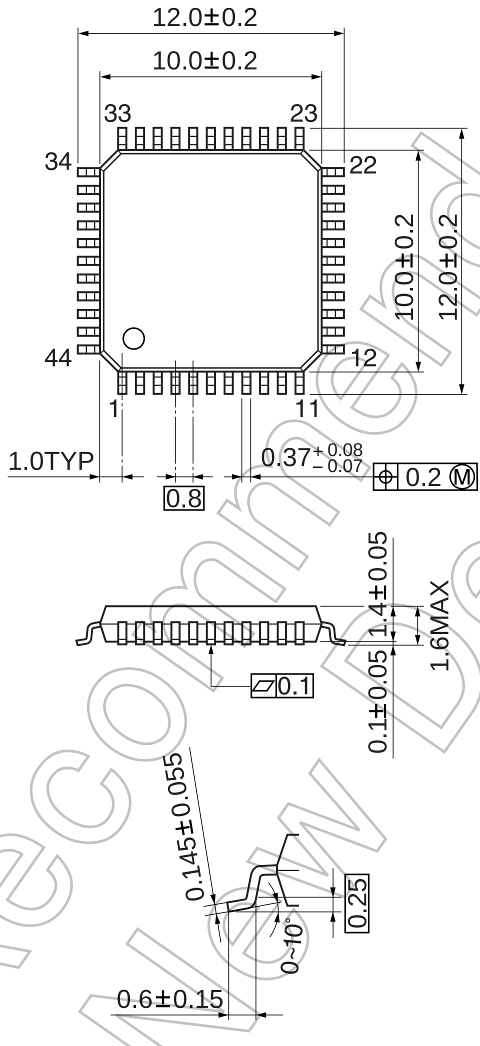
22.10 Revision History

Rev	Description
RA003	"22.4 AD Conversion Characteristics" Revised description for AD conversion error.
RA004	"22.5 Power-on Reset Circuit Characteristics" Revised spec of Power-on reset detecting voltage (V_{PRON}). "22.3 DC Characteristics" Added Note 3.
RA005	"22.9 Handling Precaution" Revised mark of lead-free.

23. Package Dimensions

LQFP44-P-1010-0.80B Rev 01

Unit: mm



RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively “TOSHIBA”), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively “Product”) without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA’s written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product’s quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the “TOSHIBA Semiconductor Reliability Handbook” and (b) the instructions for the application that Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS’ PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact (“Unintended Use”). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.