TOSHIBA

TOSHIBA Original CMOS 8-Bit Microcontroller

TLCS-900/L1 Series

TMP91CW12

TOSHIBA CORPORATION

Semiconductor Company

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, "Points of Note and Restrictions".
Especially, take care below cautions.

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{NMI}$, INT0 to INT4, INTRTC), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

CMOS 16-Bit Microcontroller

TMP91CW12F

1. Outline and Features

TMP91CW12 is a high-speed 16-bit microcontroller designed for the control of various mid-to-large scale equipment.

TMP91CW12F comes in a 100-pin flat package.

Listed below are the features.

(1) High-speed 16-bit CPU (900/L1 CPU)
- Instruction mnemonics are upward-compatible with TLCS-90/900
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
- Micro DMA: Four-channels (1.0μs/2 bytes at 16MHz)

(2) Minimum instruction execution time : 250ns (at 16MHz)

(3) Built-in RAM: 4 Kbytes
Built-in ROM: 128 Kbytes

(4) External memory expansion
- Expandable up to 16 Mbytes (Shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus
  ··· Dynamic data bus sizing

(5) 8-bit timers: 8 channels

(6) 16-bit timer/event counter: 2 channels

030619EBP1

- The information contained herein is subject to change without notice.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law and regulations.
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.

Purchase of TOSHIBA I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

(7) General-purpose serial interface: 2 channels
- UART/Synchronous mode: 2 channels
- IrDA Ver.1.0 (115.2 kbps) mode selectable: 1 channel

(8) Serial bus interface: 1 channel
- I^2C bus mode/clock synchronous mode selectable

(9) 10-bit AD converter: 8 channels

(10) Watchdog timer

(11) Timer for real-time clock (RTC)

(12) Chip select/wait controller: 4 blocks

(13) Interrupts: 45 interrupts
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 26 internal interrupts:] Seven selectable priority levels
- 10 external interrupts:]

(14) Input/output ports: 81 pins

(15) Standby mode
- Three HALT modes: Programmable-IDLE2, IDLE1, STOP

(16) Triple clock controller
- Clock doubler (DFM) circuit is inside
- Clock gear function: Selectable a high-frequency clock fc/1 to fc/16
- RTC (fs = 32.768 kHz)

(17) Operating voltage
- V_{CC} = 2.7 to 5.5 V (fc max = 16 MHz)
- V_{CC} = 4.5 to 5.5 V (fc max = 25 MHz)

(18) Package
- 100-pin QFP: P-LQFP100-1414-0.50C

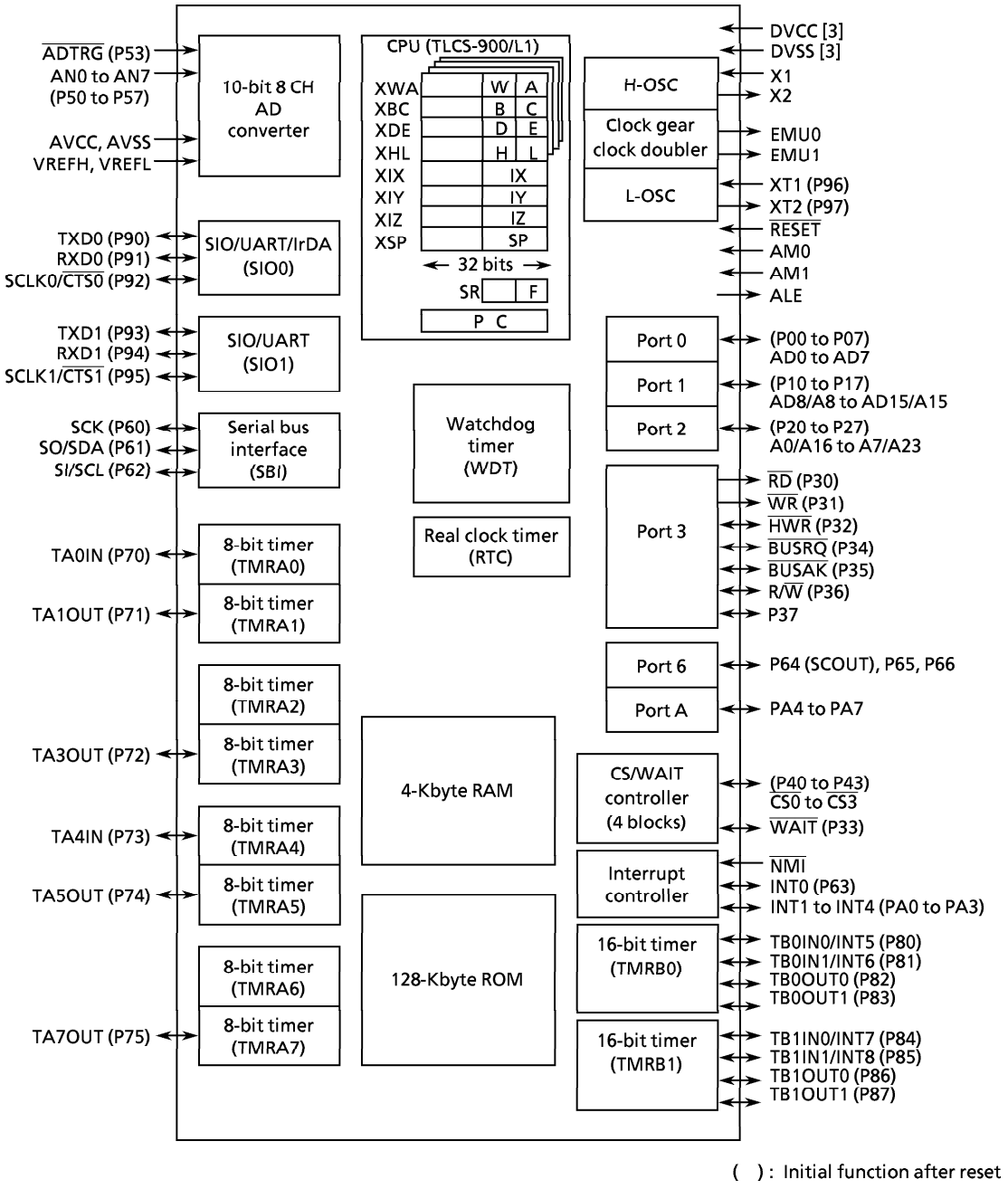


Figure 1.1 TMP91CW12 Block Diagram

2. Pin Assignment and Pin Functions

This section shows the TMP91CW12F pin assignment, and the names and an outline of the functions of the input/output pins.

2.1 Pin Assignment Diagram

Figure 2.1.1 is a pin assignment diagram for TMP91CW12F.

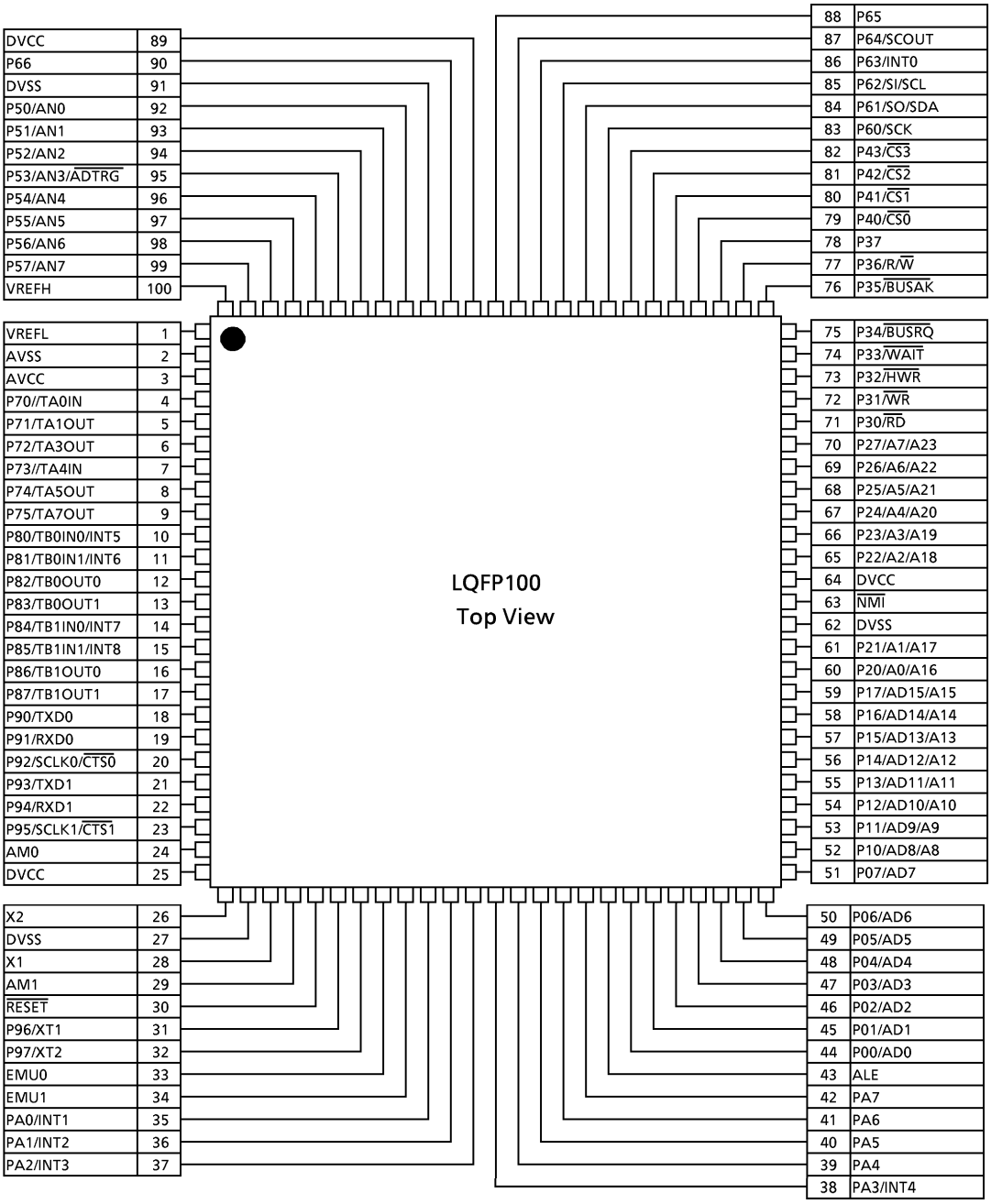


Figure 2.1.1 Pin Assignment Diagram (100-pin LQFP)

2.2 Pin Names and Functions

The names of the input/output pins and their functions are described below.
Table 2.2.1 Pin Names and Functions.

Table 2.2.1 Pin Names and Functions (1/4)

Pin Name	Number of Pins	I/O	Functions
P00 to P07 AD0 to AD7	8	I/O Tri-state	Port 0: I/O port that allows I/O to be selected at the bit level Address and data (Lower): Bits 0 to 7 for address and data bus
P10 to P17 AD8 to AD15 A8 to A15	8	I/O Tri-state Output	Port 1: I/O port that allows I/O to be selected at the bit level Address and data (Upper): Bits 8 to 15 for address and data bus Address: Bits 8 to 15 for address bus
P20 to P27 A0 to A7 A16 to A23	8	I/O Output Output	Port 2: I/O port that allows to be selected at the bit level Address: Bits 0 to 7 for address bus Address: Bits 16 to 23 for address bus
P30 $\overline{RD}$	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 $\overline{WR}$	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 to 7
P32 $\overline{HWR}$	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 to 15
P33 $\overline{WAIT}$	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait ((1 + N) WAIT mode)
P34 $\overline{BUSRQ}$	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request bus release.
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O port (with pull-up resistor) Bus acknowledge: Signal used to acknowledge bus release.
P36 R/$\overline{W}$	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0 represents write cycle.
P37	1	I/O	Port 37: I/O port (with pull-up resistor)
P40 $\overline{CS0}$	1	I/O Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area.
P41 $\overline{CS1}$	1	I/O Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area.
P42 $\overline{CS2}$	1	I/O Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area.
P43 $\overline{CS3}$	1	I/O Output	Port 43: I/O port (with pull-up resistor) Chip select 3: Outputs 0 if address is within specified address area.
P50 to P57 AN0 to AN7 $\overline{ADTRG}$	8	Input Input Input	Port 5: Pin used to input port Analog input: Pin used to input to AD converter AD trigger: Signal used to request AD start.

Note: This device's built-in memory or built-in I/O cannot be accessed by an external DMA controller, using the $\overline{BUSRQ}$ and $\overline{BUSAK}$ signals.

Table 2.2.1 Pin Names and Functions (2/4)

Pin Name	Number of Pins	I/O	Functions
P60 SCK	1	I/O I/O	Port 60: I/O port Serial bus interface clock at SIO mode.
P61 SO SDA	1	I/O Output I/O	Port 61: I/O port Serial bus interface output data at SIO mode. Serial bus interface data at I^2C bus mode.
P62 SI SCL	1	I/O Input I/O	Port 62: I/O port Serial bus interface input data at SIO mode. Serial bus interface clock at I^2C bus mode.
P63 INT0	1	I/O Input	Port 63: I/O port Interrupt request pin 0: Interrupt request pin with programmable level/rising edge/falling edge
P64 SCOUT	1	I/O Output	Port 64: I/O port System clock output: Output f_{FPH} or fs clock
P65	1	I/O	Port 65: I/O port
P66	1	I/O	Port 66: I/O port
P70 TA0IN	1	I/O Input	Port 70: I/O port Timer A0 input
P71 TA1OUT	1	I/O Output	Port 71: I/O port Timer A1 output
P72 TA3OUT	1	I/O Output	Port 72: I/O port Timer A3 output
P73 TA4IN	1	I/O Input	Port 73: I/O port Timer A4 input
P74 TA5OUT	1	I/O Output	Port 74: I/O port Timer A5 output
P75 TA7OUT	1	I/O Output	Port 75: I/O port Timer A7 output
P80 TB0IN0 INT5	1	I/O Input Input	Port 80: I/O port Timer B0 input 0 Interrupt request pin 5: Interrupt request pin with programmable rising edge/falling edge
P81 TB0IN1 INT6	1	I/O Input Input	Port 81: I/O port Timer B0 input 1 Interrupt request pin 6: Interrupt request pin with rising edge
P82 TB0OUT0	1	I/O Output	Port 82: I/O port Timer B0 output 0
P83 TB0OUT1	1	I/O Output	Port 83: I/O port Timer B0 output 1

Table 2.2.1 Pin Names and Functions (3/4)

Pin Name	Number of Pins	I/O	Functions
P84 TB1IN0 INT7	1	I/O Input Input	Port 84: I/O port Timer B1 input 0 Interrupt request pin 7: Interrupt request pin with programmable rising edge/falling edge
P85 TB1IN1 INT8	1	I/O Input Input	Port 85: I/O port Timer B1 input 1 Interrupt request pin 8: Interrupt request pin with rising edge
P86 TB1OUT0	1	I/O Output	Port 86: I/O port Timer B1 output 0
P87 TB1OUT1	1	I/O Output	Port 87: I/O port Timer B1 output 1
P90 TXD0	1	I/O Output	Port 90: I/O port Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port Serial receive data 0
P92 SCLK0 $\overline{CTS0}$	1	I/O I/O Input	Port 92: I/O port Serial clock I/O 0 Serial data send enable 0 (Clear to Send)
P93 TXD1	1	I/O Output	Port 93: I/O port Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial receive data 1
P95 SCLK1 $\overline{CTS1}$	1	I/O I/O Input	Port 95: I/O port (with pull-up resistor) Serial clock I/O 1 Serial data send enable 1 (Clear to Send)
P96 XT1	1	I/O Input	Port 96: I/O port (Open Drain Output) Low-frequency oscillator connecting pin
P97 XT2	1	I/O Output	Port 97: I/O port (Open Drain Output) Low-frequency oscillator connecting pin
PA0 to PA3 INT1 to INT4	4	I/O Input	Port A0 to A3: I/O port Interrupt request pin 1 to 4: Interrupt request pin with programmable rising edge/falling edge
PA4 to PA7	4	I/O	Port A4 to A7: I/O port
ALE	1	Output	Address latch enable (Can be disabled for reducing noise.).
$\overline{NMI}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge or both edges.
AM0/AM1	2	Input	Address mode: The Vcc pin should be connected.
EMU0/EMU1	2	Output	Test pin: Open pins.
$\overline{RESET}$	1	Input	Reset: Initializes TMP91CW12. (with pull-up resistor)

Table 2.2.1 Pin Names and Functions (4/4)

Pin Name	Number of Pins	I/O	Functions
VREFH	1	Input	Pin for reference voltage input to AD converter (H)
VREFL	1	Input	Pin for reference voltage input to AD converter (L)
X1/X2	2	I/O	High-frequency oscillator connecting pin
AVCC	1		Power supply pin for AD converter
AVSS	1		GND pin for AD converter (0 V)
DVCC	3		Power supply pin (All VCC pins should be connected with the power supply pin.)
DVSS	3		GND pin (0 V) (All VSS pins should be connected with GND (0 V).)

Note: All pins that have built-in pull-up resistors (other than the $\overline{\text{RESET}}$ pin) can be disconnected from the built-in pull-up resistor by software.

3. Operation

The following describes block by block the functions and basic operation of TMP91CW12.
Notes and restrictions for each block are outlined in 7. "Points to Note and Restrictions" at the end of this manual.

3.1 CPU

TMP91CW12 incorporates a high-performance 16-bit CPU (900/L1 CPU). For CPU operation, see the "TLCS-900/L1 CPU".
The following describes the unique functions of the CPU used in TMP91CW12; these functions are not covered in the TLCS-900/L1 CPU section.

3.1.1 Reset

When resetting the TMP91CW12 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then set the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks (80 μs at 4 MHz).
Thus, when turn on the switch, be set to the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks.
Clock gear is initialized 1/16 mode by reset operation. It means that the system clock mode f_{SYS} is set to fc/32 (= fc/16 × 1/2).
When the reset is accepted, the CPU:

- Sets as follows the program counter (PC) in accordance with the reset vector stored at address FFFF00H - FFFF02H:
  PC (7:0) ← Value at FFFF00H address
  PC (15:8) ← Value at FFFF01H address
  PC (23:16) ← Value at FFFF02H address
- Sets the stack pointer (XSP) to 100H.
- Sets bits <IFF2:0> of the status register (SR) to 111 (Sets the interrupt level mask register to level 7).
- Sets the <MAX> bit of the status register to 1 (MAX mode).
  (Note: As this product does not support a MIN mode, don't write 0 to <MAX>.)
- Clears bits <RFP2:0> of the status register to 000 (Sets the register bank to 0).

When reset is released, the CPU starts executing instructions in accordance with the program counter settings. CPU internal registers not mentioned above do not change when the reset is released.
When the reset is accepted, the CPU sets internal I/O, ports, and other pins as follows.

- Initializes the internal I/O registers.
- Sets the port pins, including the pins that also act as internal I/O, to general-purpose input or output port mode.
- Sets the ALE pin to High-Z.
  Figure 3.1.1 is a reset timing of the TMP91CW12.

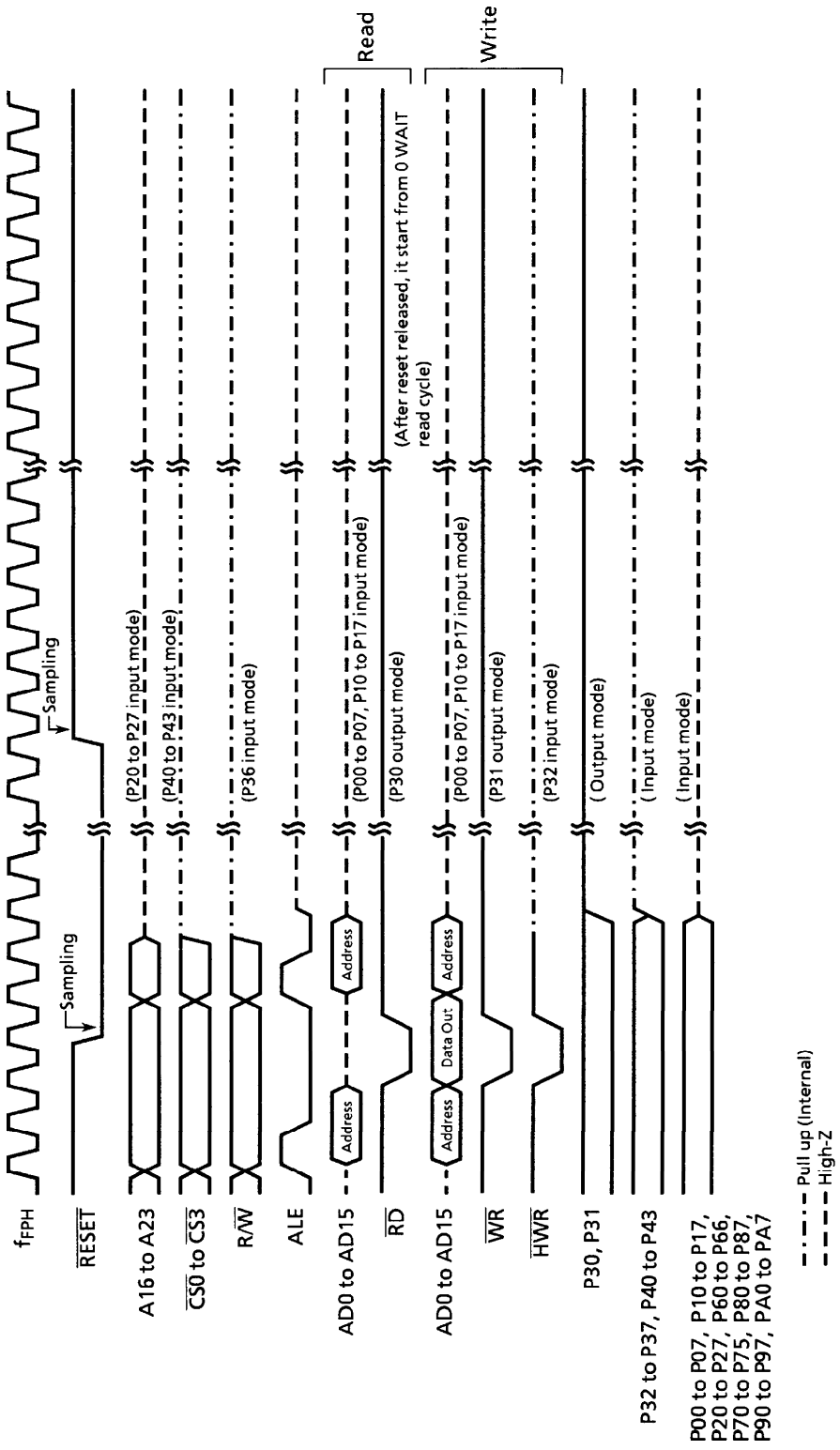


Figure 3.1.1 TMP91CW12 Reset Timing Example

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91CW12.

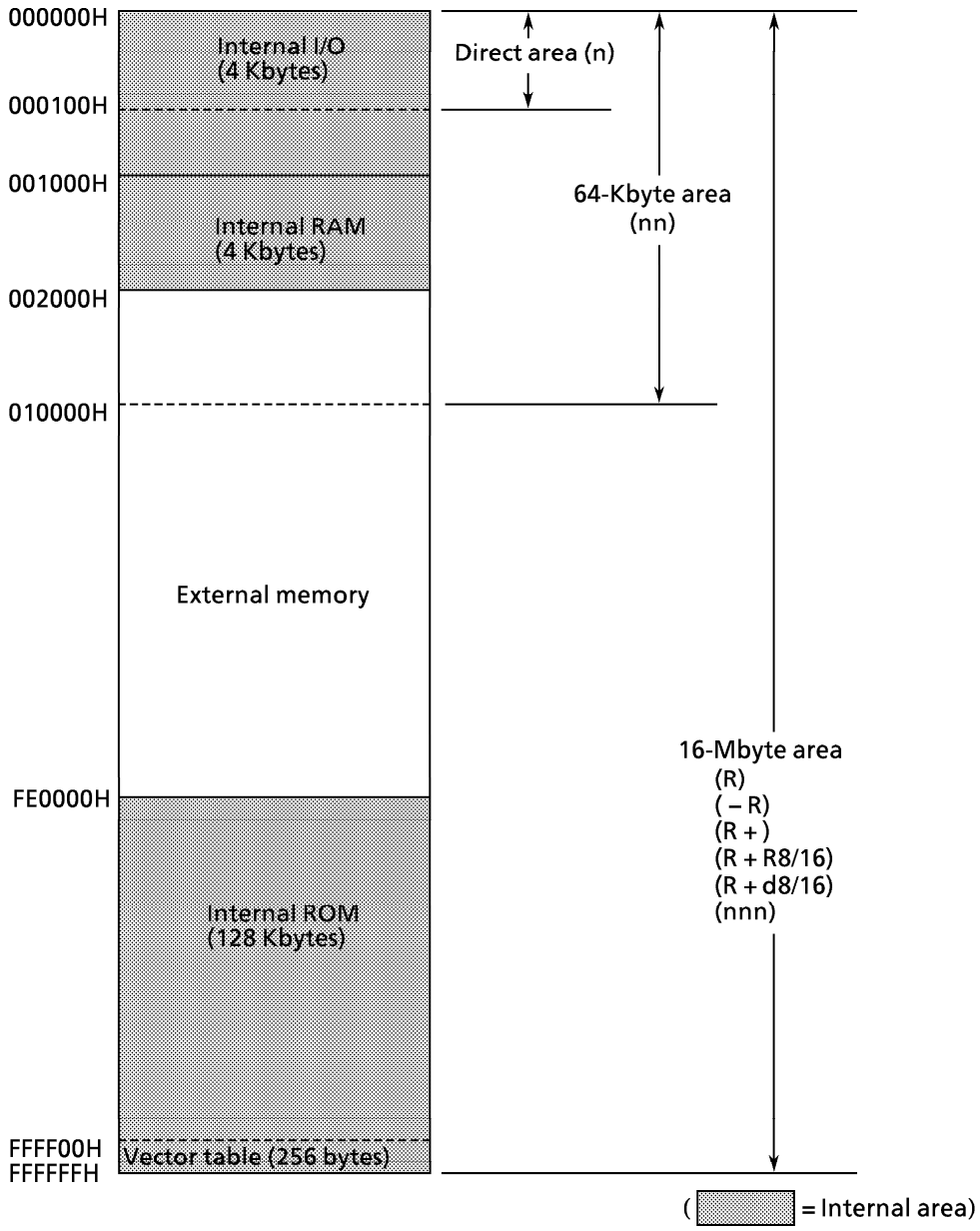
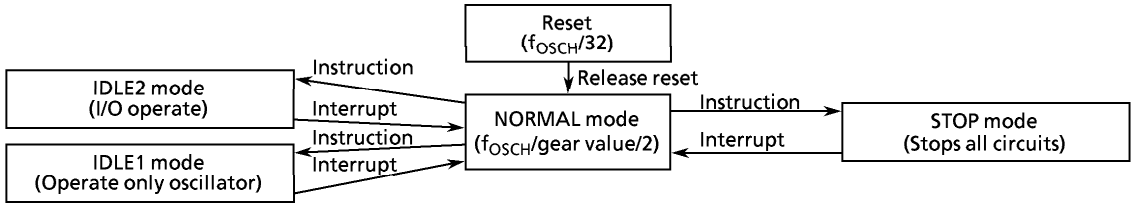


Figure 3.2.1 Memory Map

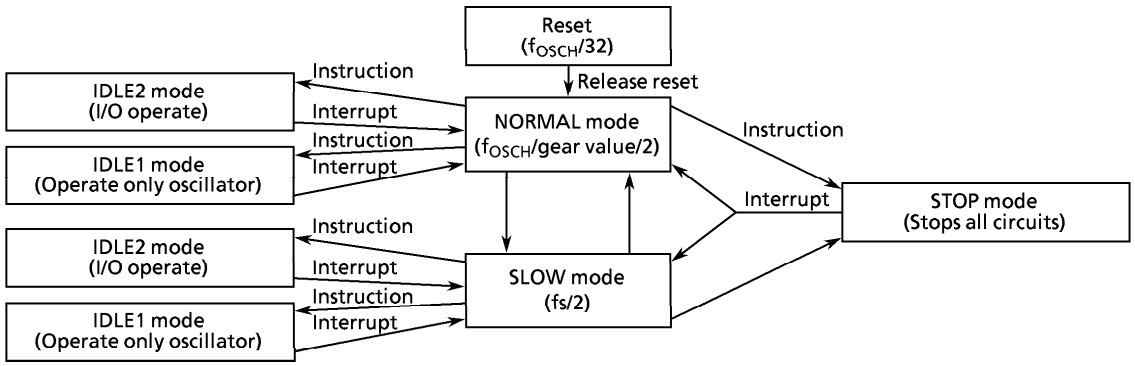
3.3 Triple Clock, Standby Function

TMP91CW12 contains (1) Clock gear, (2) Clock doubler (DFM), (3) Standby controller, and (4) Noise reducing circuit. They are used for low-power, low-noise system.

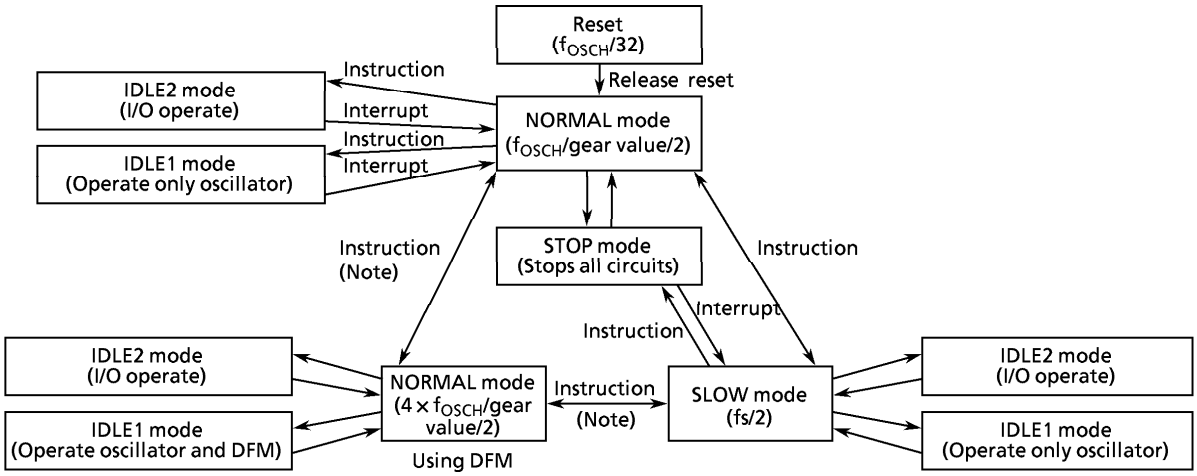
The clock operating mode is classified to (a) Single clock mode (only X1, X2 pin), (b) Dual clock mode (X1, X2, XT1, XT2 pin), and (c) Triple clock mode (X1, X2, XT1, XT2 pin and DFM).
Figure 3.3.1 shows a transition figure.



(a) Single clock mode transition figure



(b) Dual clock mode transition figure



(c) Triple clock mode transision diagram

Note 1: It's prohibited to control DFM in SLOW mode when shifting from SLOW mode to NORMAL mode with use of DFM. (DFM Start up/stop/change write to DFMCR0<ACT1:0>resister)

Note 2: If you shift from NORMAL mode with use of DFM to NORMAL mode, the instruction should be separated into two procedures as below. Change CPU clock → Stop DFM circuit.

Note 3: It's prohibited to shift from NORMAL mode with use of DFM to STOP mode directly. You should set NORMAL mode once, and then shift to STOP mode. (You should stop high-frequency oscillator after you stop DFM.)

Figure 3.3.1 System Clock Block Diagram

The clock frequency input from the X1 and X2 pins is called fc and the clock frequency input from the XT1 and XT2 pins is called fs. The clock frequency selected by SYSCR1<SYSCK> is called the system clock f_{FPH}. The system clock f_{SYS} is defined as the divided clock of f_{FPH}, and one cycle of f_{SYS} is regret to as one state.

3.3.1 Block Diagram of System Clock

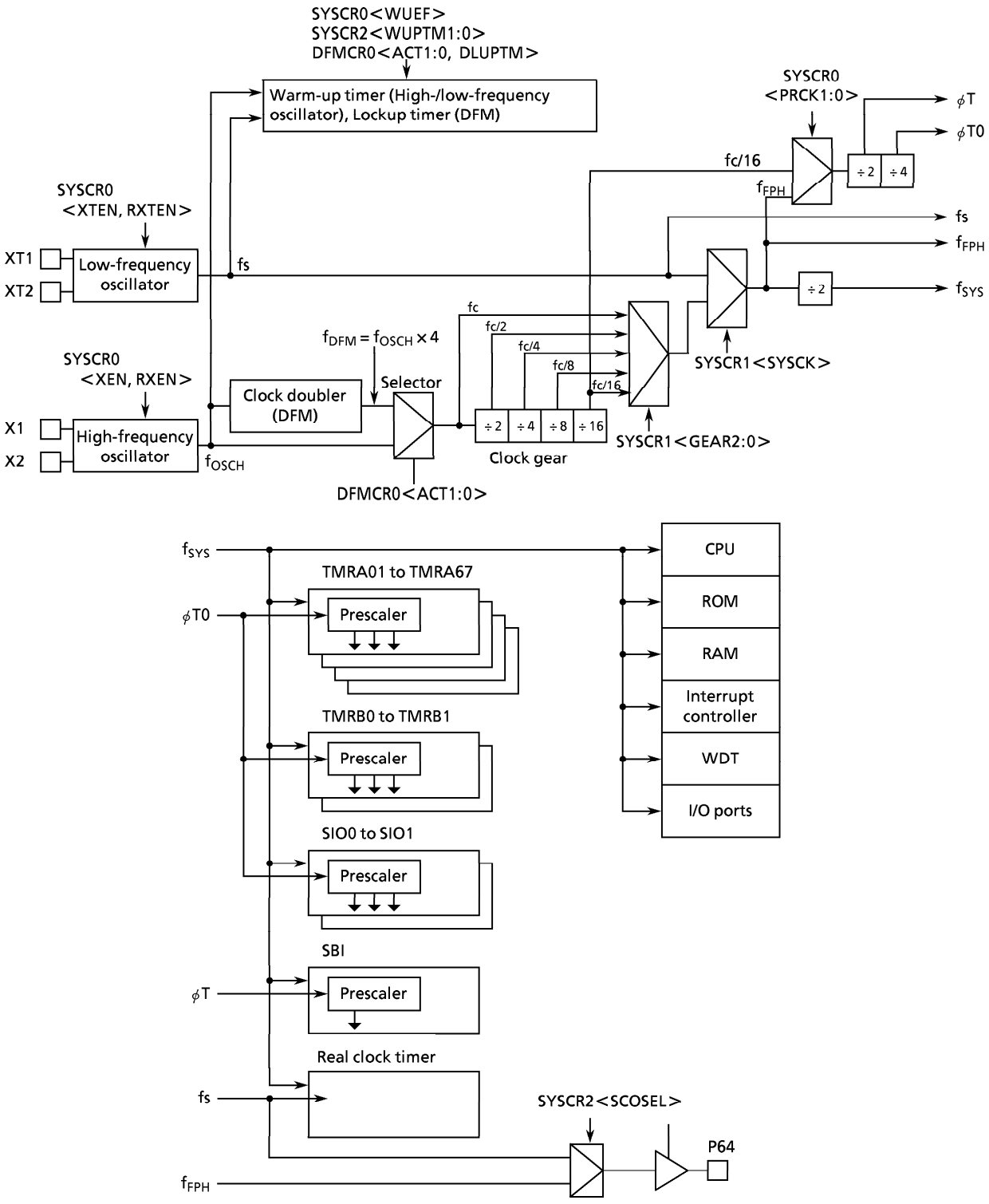


Figure 3.3.2 Block Diagram of System Clock

3.3.2 SFR

SYSCR0 (00E0H)

	7	6	5	4	3	2	1	0
Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation	High-frequency oscillator (fc) after released STOP mode 0: Stop 1: Oscillation	Low-frequency oscillator (fs) after released STOP mode 0: Stop 1: Oscillation	Slect clock after released STOP mode 0: fc 1: fs	Warm-up timer 0 write: Don't care 1 write: Start timer 0 read: End warm up 1 read: Not end warm up	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	

SYSCR1 (00E1H)

	7	6	5	4	3	2	1	0
Bit symbol					SYSCK	GEAR2	GEAR1	GEAR0
Read/Write					R/W			
After reset					0	1	0	0
Function					Select system clock 0: fc 1: fs	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		

SYSCR2 (00E2H)

	7	6	5	4	3	2	1	0
Bit symbol		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
Read/Write		R/W	R/W	R/W	R/W	R/W		R/W
After reset		0	1	0	1	1		0
Function		0: fs 1: f_{FPH}	Warm-up timer 00: Reserved 01: 2^8/inputted frequency 10: 2^{14} 11: 2^{16}		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			Pin state control in STOP mode 0: I/O off 1: Remains the state before halt

Note 1: SYSCR1<Bit 7:4>, SYSCR2<Bit 7, 1>are read as undefined-value.

Note 2: In case of using built-in SBI circuit, set SYSCR0<PRCK1:0> to 00.

Figure 3.3.3 SFR for System Clock

DFMCR0 (00E8H)

	7	6	5	4	3	2	1	0
Bit symbol	ACT1	ACT0	DLUPFG	DLUPTM				
Read/Write	R/W	R/W	R	R/W				
After reset	0	0	0	0				
Function	DFM LUP Select f_{FPH} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lockup status flag 0: End 1: Not end	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				

Figure 3.3.4 SFR for DFM

Limitation point on the use of DFM

1. It's prohibited to execute DFM enable/disable control in the SLOW mode(fs) (write to DFMCR0<ACT1:0>="10"). You should control DFM in the NORMAL mode.
2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0>="10"), you shouldn't execute that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above executions should be separated into two procedures as showing below.

   LD (DFMCR0),C0H; Change the clock f_{DFM} to f_{OSCH}

   LD (DFMCR0),00H; DFM stop
3. If you stop high-frequency oscillator during using DFM (DFMCR0<ACT1:0>="10"), you should stop DFM before you stop high-frequency oscillator.

Please refer to 3.3.5 "Clock Doubler (DFM)" for the details.

EMCCR0 (00E3H)

	7	6	5	4	3	2	1	0
Bit symbol	PROTECT	–	–	–	ALEEN	EXTIN	DRVOSCH	DRVOSCL
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	1	0	0	0	1	1
Function	Protect flag 0: OFF 1: ON	Write 0	Write 1	Write 0	ALE pin output control 0: High-Z output 1: ALE output	1 : External clock	fc oscillator driving ability 1: Normal 0: Weak	fs oscillator driving ability 1: Normal 0: Weak

EMCCR1 (00E4H)

	7	6	5	4	3	2	1	0
Bit symbol	The protect is OFF by writing 1FH. The protect is ON by writing except 1FH.							
Read/Write								
After reset								
Function								

Figure 3.3.5 SFR for Noise Reducing

3.3.3 System Clock Controller

The system clock controller generates system clock (f_{SYS}) for CPU core and internal I/O. It contains two oscillation circuits and clock gear circuit for high frequency (fc). The register SYSCR1<SYSCK> changes system clock to either fc or fs, SYSCR0<XEN>, <XTEN> controls enable/disable each oscillator, SYSCR1<GEAR2:0> changes high frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16), and these functions can reduce the power consumption of the equipment in which the device is installed.
The system clock (f_{SYS}) is set to fc/32 (fc/16×1/2) because of <XEN>=1, <XTEN>=0, <SYSCK>=0, <GEAR2:0>=100 by resetting.
For example, f_{SYS} is set to 0.5 MHz by resetting the case of 16 MHz oscillator is connected to X1, X2 pins.

(1) Switching NORMAL to SLOW mode

When the resonator is connected to X1, X2, or XT1, XT2 pin, the warm-up timer is used to change the operation frequency after getting stabilized oscillation. The warm-up time can be selected by SYSCR2<WUPTM1:0>. This starting and ending of warm-up timer are performed like the following example 1, 2 by program.
Table 3.3.1 shows the warm-up time.

Note 1: The case of using oscillator (Not resonator) with stabilized oscillation, a warm-up timer is not need.

Note 2: The warm-up timer is operated by a oscillation clock. Therefore, warm-up time has an error.

Note 3: Note on using low-frequency oscillation circuit

To connect the low-frequency resonator to port 96, 97, it is necessary to set the following to reduce the power consumption.
(connecting with resonators)
P9CR<P96C, P97C>=11, P9<P96:97>= 00
(connecting with oscillators)
P9CR<P96C, P97C>=11, P9<P96:97>= 10

Table 3.3.1 Warm-up Time

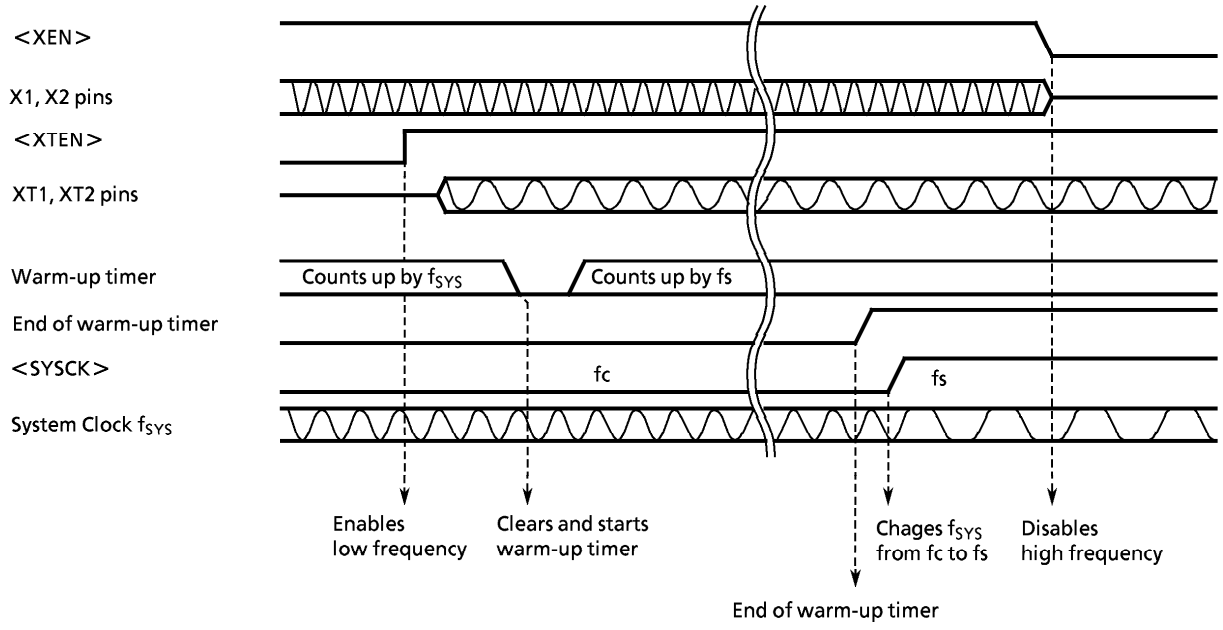
Warm-up Time SYSCR2<WUPTM1:0>	Change to NORMAL	Change to SLOW
01 (2^8/frequency)	16 (μs)	7.8 (ms)
10 (2^{14}/frequency)	1.024 (ms)	500 (ms)
11 (2^{16}/frequency)	4.096 (ms)	2000 (ms)

at f_{OSCH} = 16 MHz,
fs = 32.768 kHz

(Setting example 1)

The case of changing from high frequency (fc) to low frequency (fs).

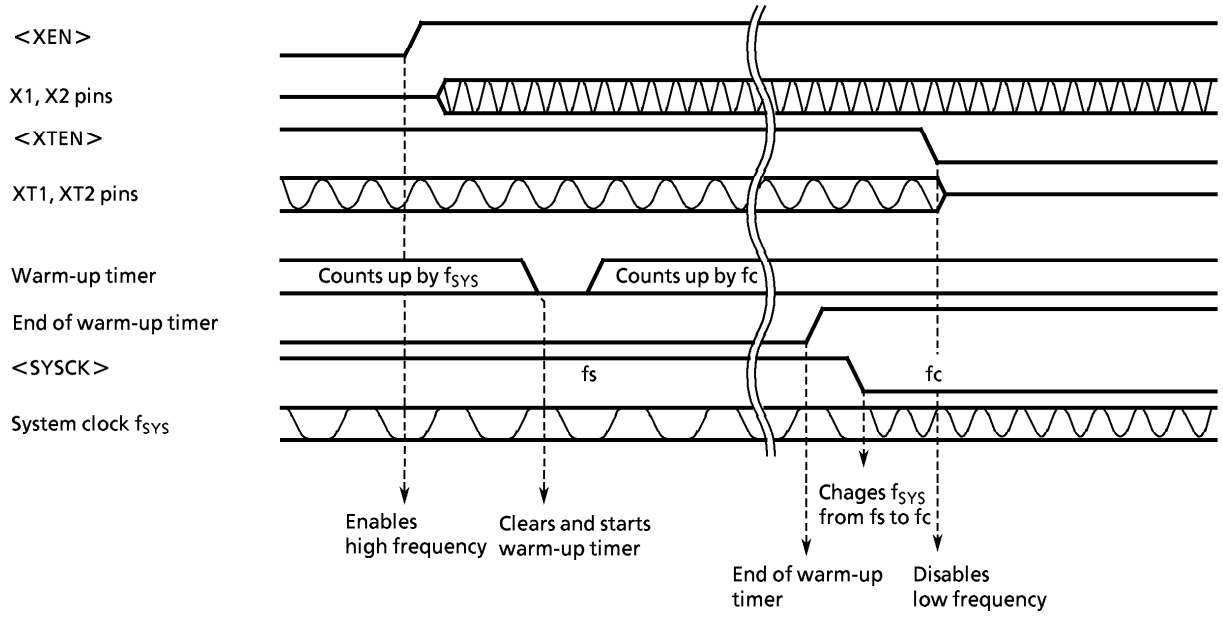
```
SYSCR0   EQU   00E0H
SYSCR1   EQU   00E1H
SYSCR2   EQU   00E2H
         LD    (SYSCR2), X-11--X-B   ; Sets warm-up time to 2^16/fs.
         SET   6, (SYSCR0)           ; Enables low-frequency oscillation
         SET   2, (SYSCR0)           ; Clears and starts warm-up timer.
WUP :    BIT   2, (SYSCR0)           ;
         JR    NZ, WUP               ;  } Detects end of warm-up timer.
         SET   3, (SYSCR1)           ; Changes fSYS from fc to fs.
         RES   7, (SYSCR0)           ; Disables high-frequency oscillation.
```



(Setting example 2)

The case of changing from low frequency (fs) to high frequency (fc).

```
SYSCR0    EQU   00E0H
SYSCR1    EQU   00E1H
SYSCR2    EQU   00E2H
          LD    (SYSCR2), X-10--X-B   ; Sets warm-up time to 2^14/fc.
          RES   7, (SYSCR0)           ; Enables high frequency (fc).
          SET   2, (SYSCR0)           ; Clears and starts warm-up timer.
WUP :     BIT   2, (SYSCR0)           ;
          JR    NZ, WUP               ;   } Detects end of warm-up timer.
          RES   3, (SYSCR1)           ; Changes fSYS from fs to fc.
          RES   6, (SYSCR0)           ; Disables low-frequency oscillation.
```



(2) Clock gear controller

When the high-frequency clock fc is selected at SYSCR1<SYSCK> = 0, the clock gear select register SYSCR1<GEAR2:0> sets f_{FPH} to either fc, fc/2, fc/4, fc/8, fc/16. Switching f_{FPH} with the clock gear reduces the power consumption.

(Setting example 3)

The case of changing gear value of high frequency

```
SYSCR1  EQU  00E1H
        LD   (SYSCR1), XXXX0000B        ; Changes fSYS to fc/2
```

X : Don't care

(High-speed clock gear changing)

To change the clock gear, write the register value to SYSCR1<GEAR2:0> register. It is necessary the warm-up time until changing after writing the register value.
There is the possibility that the instruction next to the clock gear changing instruction is executed by the clock gear before changing. To execute the instruction next to the clock gear switching instruction by the clock gear after changing, input the dummy instruction as follows (Instruction to execute the write cycle).

(Example)

```
SYSCR1  EQU  00E1H
        LD   (SYSCR1), XXXX0001B        ; Changes fSYS to fc/4.
        LD   (DUMMY), 00H               ; Dummy instruction
```

Instruction to be executed by the clock gear after changing

(3) Internal clock pin output function

P64/SCOUT pin outputs the internal clocks f_{FPH} or f_S.
The port 6 control register P6CR<P64C> = 1, P6FC<P64F> = 1 specifies the SCOUT output pin. The selection of output clock is set by SYSCR2<SCOSEL>.
Table 3.3.2 shows pin states in the respective operation modes which is under condition that P64/SCOUT pin is specified as SCOUT output.

Table 3.3.2 SCOUT Pin States in the Operation Modes

Operation Mode / SCOUT Select	NORMAL, SLOW	HALT Mode		
		IDLE2	IDLE1	STOP
<SCOSEL> = 0	Outputs fs clock.			Fixed to 0 or 1.
<SCOSEL> = 1	Outputs f_{FPH} clock.			

3.3.4 Prescaler Clock Controller

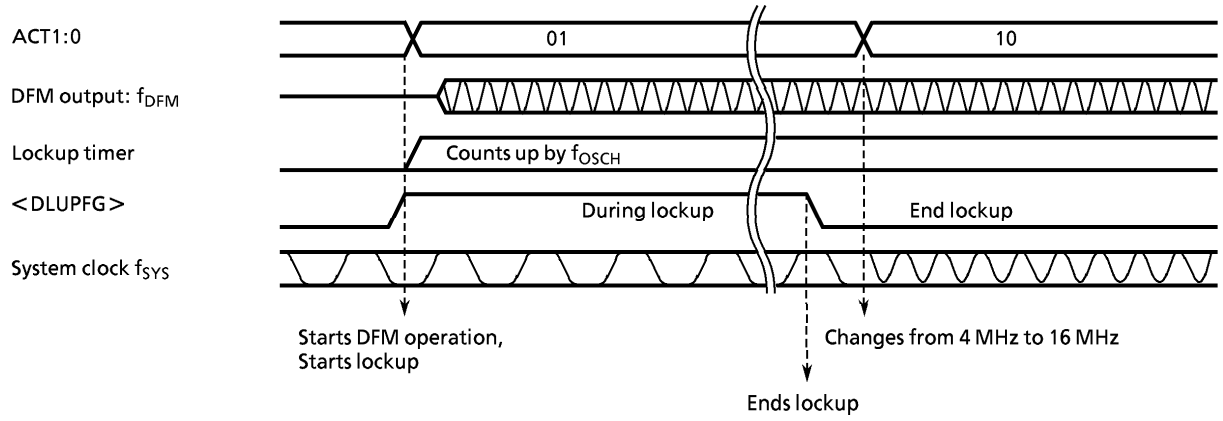
The internal I/O (TMRA01 to TMRA67, TMRB0, TMRB1, SIO0, SIO1) has the prescaler which divide the clock.
The ϕT0 clock input to prescaler is a clock divided by 4 which is selected either f_{FPH} or fc/16 by SYSCR0 <PRCK1:0> register.
In case of using built-in SBI circuit, set SYSCR0<PRCK1:0> to 00.

3.3.5 Clock Doubler (DFM)

DFM can output f_{DFM} clock which is 4-times for f_{OSCH}. It is available to use the low-frequency oscillator though the internal clock is high frequency.
DFM is initialized to stop status by reset, setting to DFMCR0 register is needed before use.
This circuit needs a stabilized time like an oscillator which is called lockup time.
The following is the example for using DFM.

```
DFMCR0   EQU   00E8H
         LD    (DFMCR0), 01X0XXXXB ;  Set lockup time to 2^12/4 MHz
                                   ;  Enables DFM operation and starts lockup.
LUP:     BIT   5, (DFMCR0)         ;  } Detects end of lockup
         JR    NZ, LUP             ;  }
         LD    (DFMCR0), 10X0XXXXB ;  Changes fc from 4 MHz to 16 MHz.
                                      (Changes fSYS from 2 MHz to 8 MHz.)
```

X: Don't care



Note: Input frequency limitation for DFM
The input frequency for DFM is recommended to use at following condition.

- f_{OSCH} = 2 to 4 MHz (V_{CC} = 3 V ± 10%)
- f_{OSCH} = 4 to 6.25 MHz (V_{CC} = 5 V ± 10%)

Limitation point on the use of DFM

1. It's prohibited to execute DFM enable/disable control in the SLOW mode(fs) (write to DFMCR0<ACT1:0> = "10"). You should control DFM in the NORMAL mode.
2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0> = "10"), you shouldn't execute the commands that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above executions should be separated into two procedures as showing below.

   LD (DFMCR0), C0H ; Change the clock f_{DFM} to f_{OSCH}

   LD (DFMCR0), 00H ; DFM stop

3. If you stop high-frequency oscillator during using DFM (DFMCR0<ACT1:0> = "10"), you should stop DFM before you stop high-frequency oscillator.

Examples of settings are below.

(1) Start up/change control

(OK) Low-frequency oscillator operation mode(fs) (High-frequency oscillator STOP) → High-frequency oscillator start up → High-frequency oscillator operation mode (f_{OSCH}) → DFM start up → DFM use mode (f_{DFM})

```
      LD    (SYSCR0), 11---1--B   ; High-frequency oscillator start-up/ Warm-up start
WUP:  BIT   2,(SYSCR0)            ; } Check for the flag of warm-up end
      JR    NZ, WUP               ; }
      LD    (SYSCR1), ----0---B   ; Change the system clock fs to fOSCH
      LD    (DFMCR0),01-0----B    ; DFM start up/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lockup end
      JR    NZ, LUP               ; }
      LD    (DFMCR0),10-0----B    ; Change the system clock fOSCH to fDFM
```

(OK) Low-frequency oscillator operation mode (fs) (High-frequency oscillator operate) → High-frequency oscillator operation mode (f_{OSCH}) → DFM start up → DFM use mode (f_{DFM})

```
      LD    (SYSCR1), ----0---B   ; Change the system clock fs to fOSCH
      LD    (DFMCR0), 01-0----B   ; DFM start up/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lockup end
      JR    NZ,LUP                ; }
      LD    (DFMCR0), 10-0----B   ; Change the system clock fOSCH to fDFM
```

(Error) Low-frequency oscillator operation mode(fs) (High-frequency oscillator STOP) → High-frequency oscillator start up → DFM start up → DFM use mode (f_{DFM})

```
      LD    (SYSCR0), 11---1--B   ; High-frequency oscillator start up/Warm-up start
WUP:  BIT   2, (SYSCR0)           ; } Check for the flag of warm-up end
      JR    NZ,WUP                ; }
      LD    (DFMCR0),01-0----B    ; DFM start up/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lockup end
      JR    NZ,LUP                ; }
      LD    (DFMCR0),10-0----B    ; Change the internal clock fOSCH to fDFM
      LD    (SYSCR1), ----0---B   ; Change the system clock fs to fDFM
```

(2) Change/stop control

(OK) DFM use mode (f_{DFM}) → High-frequency oscillator operation mode (f_{OSCH}) → DFM stop → Low-frequency oscillator operation mode (fs) → High-frequency oscillator stop

```
LD    (DFMCR0),11------B    ;  Change the system clock fDFM to fOSCH
LD    (DFMCR0),00------B    ;  DFM stop
LD    (SYSCR1), --  --1---B ;  Change the system clock fOSCH to fs
LD    (SYSCR0), 0-------B   ;  High-frequency oscillator stop
```

(Error) DFM use mode (f_{DFM}) → Low-frequency oscillator operation mode(fs) → DFM stop →High-frequency oscillator stop

```
LD    (SYSCR1), ----1---B   ;  Change the system clock fDFM to fs
LD    (DFMCR0),11------B    ;  Change the internal clock (fc) fDFM to fOSCH
LD    (DFMCR0),00------B    ;  DFM stop
LD    (SYSCR0), 0-------B   ;  High-frequency oscillator stop
```

(OK) DFM use mode (f_{DFM})→Set the STOP mode →High-frequency oscillator operation mode (f_{OSCH}) → DFM stop→ HALT (High-frequency oscillator stop)

```
LD    (SYSCR2), ----01--B   ;  Set the STOP mode
                            ;  (This command can execute before use of DFM)
LD    (DFMCR0),11------B    ;  Change the system clock fDFM to fOSCH
LD    (DFMCR0),00------B    ;  DFM stop
HALT                        ;  Shift to STOP mode
```

(Error) DFM use mode (f_{DFM})→Set the STOP mode→ HALT (High-frequency oscillator stop)

```
LD    (SYSCR2), ----01--B   ;  Set the STOP mode
                            ;  (This command can execute before use of DFM)
HALT                        ;  Shift to STOP mode
```

3.3.6 Noise Reducing Circuits

Noise reducing circuits are built-in to realize the following features.

(1) Reduce drivability for high-frequency oscillator
(2) Reduce drivability for low-frequency oscillator
(3) Single drive for high-frequency oscillator
(4) Disables output for ALE pin
(5) Protect register

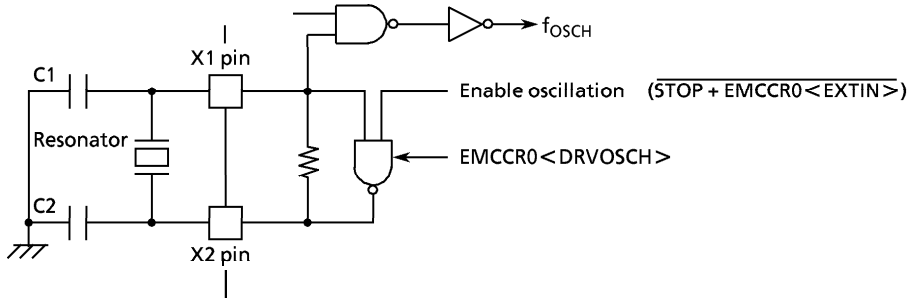
Above function is performed by setting to EMCCR0, EMCCR1 register.

(1) Reduce drivability for high-frequency oscillator

(Purpose)

Reduce noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

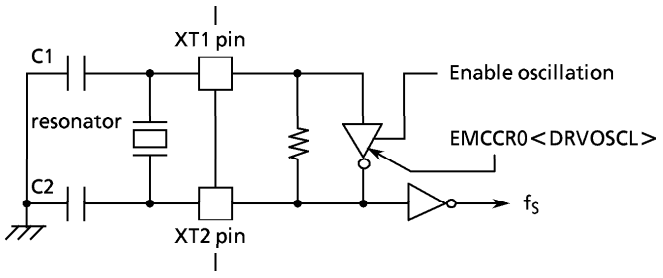
The drivability of oscillator is reduced by writing 0 to EMCCR0<DRVOSCH> register. By reset, <DRVOSCH> is initialized to 1 and the oscillator starts oscillation by normal drivability when the power supply is on.

(2) Reduce drivability for low-frequency oscillator

(Purpose)

Reduce noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

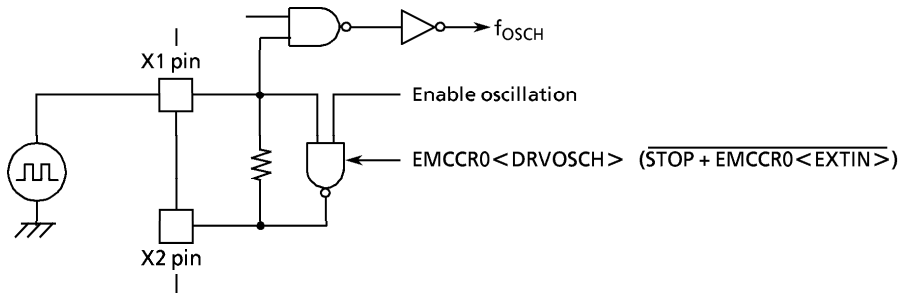
The drivability of oscillator is reduced by writing 0 to EMCCR0<DRVOSCL> register. By reset, <DRVOSCL> is initialized to 1.

(3) Single drive for high-frequency oscillator

(Purpose)

Not need twin drive and protect mistake operation by inputted noise to X2 pin when the external oscillator is used.

(Block diagram)



(Setting method)

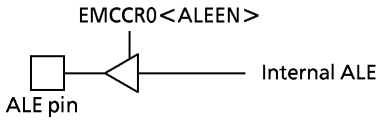
The oscillator is disabled and starts operation as buffer by writing 1 to EMCCR0<EXTIN> register. X2-pin is always outputted 1.
By reset, <EXTIN> is initialized to 0.

(4) Disables Output for ALE-pin

(Purpose)

Disables output ALE pulse for reducing noise when CPU does not access to external area.

(Block diagram)



(Setting method)

ALE pin is set to high-impedance by writing 0 to EMCCR0<ALEEN> register. By reset, <ALEEN> is initialized to 0 .
Write 1 to <ALEEN> before access when CPU will access to external-area.

(5) Protect register

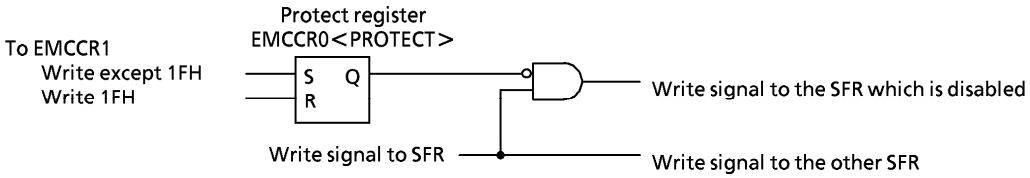
(Purpose)

An item for mistake operation by inputted noise.
To execute the program certainty which is occurred mistake-operation, the protect register can be disabled write operation for the specific SFR.

The SFR which is disabled to write.

1. CS/WAIT controller
   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0, MSAR1, MSAR2, MSAR3,
   MAMR0, MAMR1, MAMR2, MAMR3
2. Clock gear (Only EMCCR1 is available to write.)
   SYSCR0, SYSCR1, SYSCR2, EMCCR0
3. DFM
   DFMCR0

(Block diagram)



(Setting method)

The protect-status is ON by writing except 1FH codes to EMCCR1 register, and CPU is disabled to write-operation to the specific-SFR.
The protect-status is OFF by writing 1FH code to EMCCR1.
The protect-status is set to EMCCR0<PROTECT> register.
It is initialized to OFF by resetting.

3.3.7 Standby Controller

(1) HALT mode

When the HALT instruction is executed, the operating mode changes IDLE2, IDLE1 or STOP mode depending on the contents of SYSCR2<HALTM1:0> register.
The future of IDLE2, IDLE1 and STOP modes are as follows.

① IDLE2: Only the CPU halts.
The internal I/O is available to select operation during IDLE2 mode by setting the following register.
Table 3.3.3 shows the register of setting operation during IDLE2 mode.

Table 3.3.3 SFR Setting Operation during IDLE2 mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
TMRA45	TA45RUN<I2TA45>
TMRA67	TA67RUN<I2TA67>
TMRB0	TB0RUN<I2TB0>
TMRB1	TB1RUN<I2TB1>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
SBI	SBI0BR0<I2SBI0>
AD converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

② IDLE1: Only both the oscillator and RTC (Real time clock) operate.

③ STOP: All internal circuits stop.

The operation in the HALT modes is described in table 3.3.4

Table 3.3.4 I/O Operation During HALT Mode

<table>
<tr><td colspan="2">HALT Mode</td><td>IDLE2</td><td>IDLE1</td><td>STOP</td></tr>
<tr><td colspan="2">SYSCR2<HALTM1:0></td><td>11</td><td>10</td><td>10</td></tr>
<tr><td rowspan="8">Block</td><td>CPU</td><td colspan="3">Stop</td></tr>
<tr><td>I/O port</td><td colspan="2">Keep the state when the HALT instruction was executed.</td><td>See Table 3.3.7, Table 3.3.8</td></tr>
<tr><td>TMRA, TMRB</td><td rowspan="5">Available to select operation block</td><td colspan="2" rowspan="6">Stop</td></tr>
<tr><td>RTC</td><td>Operation available</td></tr>
<tr><td>SIO, SBI</td></tr>
<tr><td>AD converter</td></tr>
<tr><td>WDT</td></tr>
<tr><td>Interrupt controller</td><td>Operate</td></tr>
</table>

(2) How to Release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combinations between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.5.

- Released by requesting an interrupt

The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the HALT instruction exceeds the value of the interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU starts executing an instruction that follows the HALT instruction. When the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (in non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.)
However only for INT0 to INT4 and RTC interrupts, even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is executed. In this case, interrupt processing is not processed, and CPU starts executing the instruction next to the HALT instruction, but the interrupt request flag is held at 1.

Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0 to INT4, INTRTC), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.) If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

- Release by resetting

Releasing all halt status is executed by resetting.
When the STOP mode is released by reset, it is necessary enough resetting time (3 ms or more) to set the operation of the oscillator to be stable.
When releasing the HALT mode by resetting, the internal RAM data keeps the state before the HALT instruction is executed. However the other setting contents are initialized. (Releasing due to interrupts keep the state before the HALT instruction is executed.)

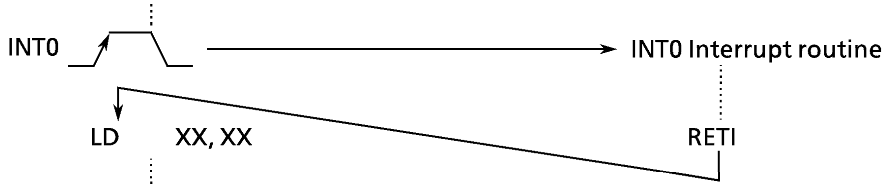
Table 3.3.5 Halt Releasing Source and Halt Releasing Operation

Interrupt Receiving Status			Interrupt Enable (Interrupt level) ≧ (Interrupt mask)			Interrupt Disable (Interrupt level) < (Interrupt mask)		
HALT Mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Halt releasing source	Interrupt	NMI	◆	◆	◆*1	–	–	–
		INTWDT	◆	×	×	–	–	–
		INT0 to INT4 (Note 1)	◆	◆	◆*1	○	○	○*1
		INTRTC	◆	◆	×	○	○	×
		INT5 to INT8 (Note 2)	◆	×	×	×	×	×
		INTTA0 to INTTA7	◆	×	×	×	×	×
		INTTB00, 01, 10, 11, OF0, OF1	◆	×	×	×	×	×
		INTRX0 to 1, TX0 to TX1	◆	×	×	×	×	×
		INTSBI	◆	×	×	×	×	×
		INTAD	◆	×	×	×	×	×
	Reset		Initialize LSI					

◆: After releasing the HALT mode, CPU starts interrupt processing.
○: After releasing the HALT mode, CPU starts executing an instruction that follows the HALT instruction.
×: It can not be used to release the HALT mode.
–: The priority level (Interrupt request level) of non-maskable interrupts is fixed to highest priority level 7. There is not this combination type.
*1: Releasing the HALT mode is executed after passing the warm-up time.
Note 1: When releasing the HALT mode is executed by INT0 interrupt of the level mode in the interrupt enabled status, hold level H until starting interrupt processing. If level L is set before holding level L, interrupt processing is correctly started.
Note 2: When the external interrupts INT5 to INT8 are used during IDLE2 mode, set to 1 for TB0RUN<I2TB0>, TB1RUN<I2TB1>.

(Example releasing IDLE1 mode)
INT0 interrupt releases halt state when the IDLE1 mode is on.

```
Address
8203H    LD     (IIMC), 00H       ; Selects INT0 interrupt rising edge.
8206H    LD     (INTE0AD), 06H    ; Sets interrupt level to 6 for INT0.
8209H    EI     5                 ; Sets interrupt level to 5 for CPU.
820BH    LD     (SYSCR2), 28H     ; Sets HALT mode to IDLE1.
820EH    HALT                     ; Halts CPU.
```




```
820FH    LD     XX, XX
```

(3) Operation

① IDLE2 mode

In the IDLE2 mode, only specific internal I/O which is selected operation by IDLE2 setting register operates, and the CPU stops executing the current instruction.

Figure 3.3.6 illustrates the timing example for releasing the halt state by interrupts in the IDLE2 mode.

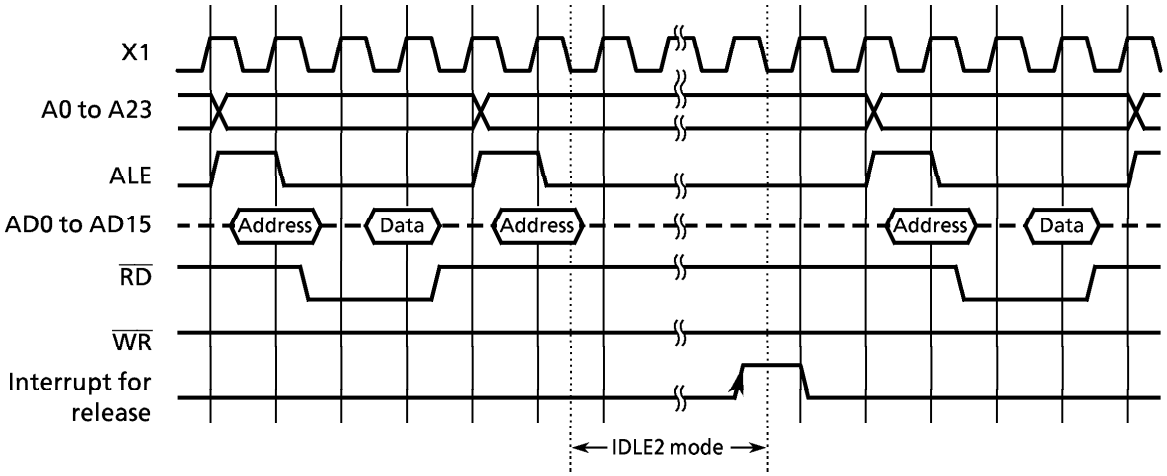


Figure 3.3.6 Timing Chart of HALT Released by Interrupts in IDLE2 Mode

② IDLE1 mode

In the IDLE1 mode, only the internal oscillator and RTC operate. The system clock in the MCU stops. In the halt state, and interrupt request is sampled asynchronunsly with the system clock, however the halt release (Restart of operation) is performed synchronously with it.

Figure 3.3.7 illustrates the timing for releasing the halt state by interrupts in the IDLE1 mode.

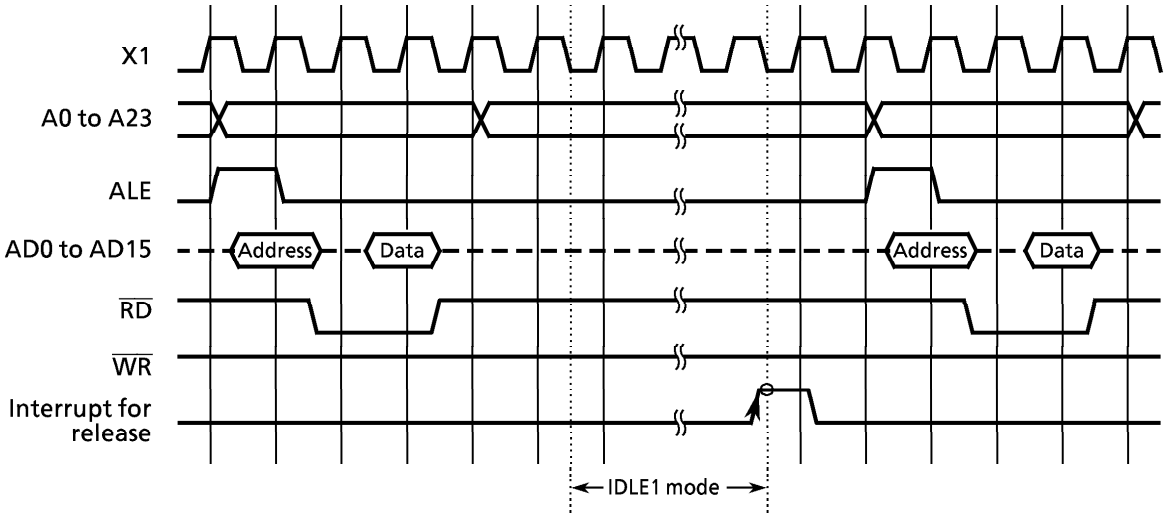


Figure 3.3.7 Timing Chart of Halt Released by Interrupts in IDLE1 Mode

③ STOP mode

The STOP mode is selected to stop all internal circuits including the internal oscillator. The pin status in the STOP mode is depended on setting the register SYSCR2<DRVE>. Table 3.3.7, Table 3.3.8 summarizes the state of these pins in the STOP mode.
When the STOP mode is released, the system clock is started outputting after warm-up timer to get the stabilized oscillation. The NORMAL/SLOW mode selection is possible after releasing STOP mode. This is selected by SYSCR0<RSYSCK> register. Therefore, setting to <RSYSCK>, <RXEN>, <RXTEN> is necessary before HALT instruction is executed. A warm-up time can be set using SYSCR2<WUPTM1:0>. See the example of warm-up time in Table 3.3.6.

Figure 3.3.8 illustrates the timing for releasing the halt state by interrupts in the STOP mode.

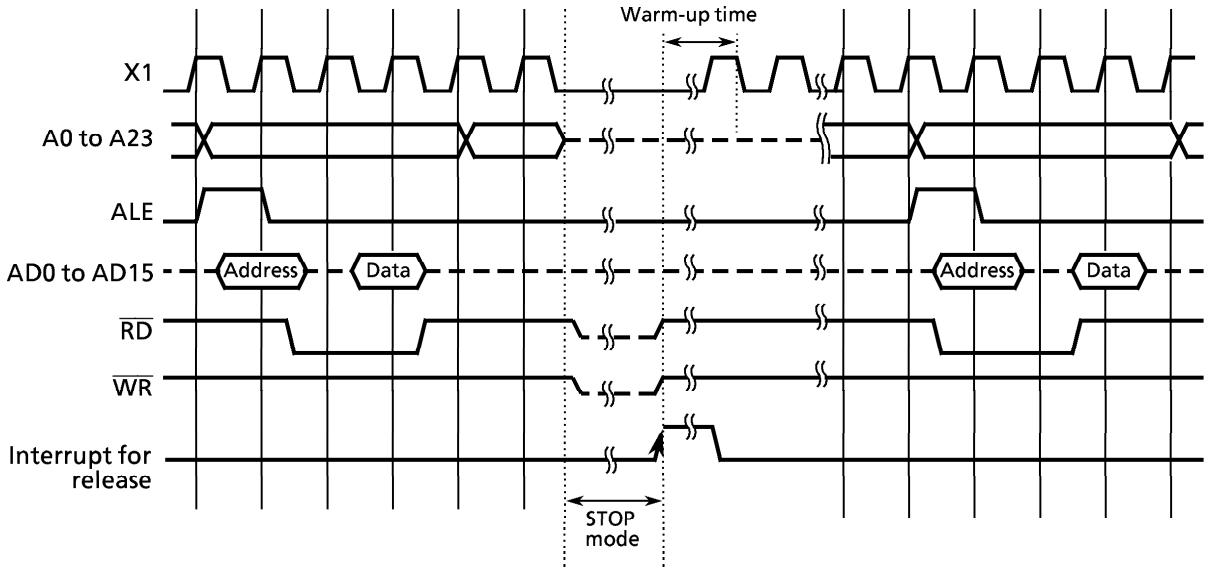


Figure 3.3.8 Timing Chart of Halt Released by Interrupts in STOP Mode

Table 3.3.6 The Example of Warm-up Time after Releasing the STOP Mode

at f_{OSCH} = 16 MHz, fs = 32.768 kHz

SYSCR0 <RSYSCK>	SYSCR2<WUPTM1:0>		
	01 (2^8)	10 (2^{14})	11 (2^{16})
0 (fc)	16 μs	1.024 ms	4.096 ms
1 (fs)	7.8 ms	500 ms	2000 ms

(Setting example) The STOP mode is entered when the low-frequency operates, and high-frequency operates after releasing due to NMI.

```
Address
            SYSCR0      EQU     00E0H
            SYSCR1      EQU     00E1H
            SYSCR2      EQU     00E2H
8FFDH                   LD      (SYSCR1), 08H           ; fSYS = fs/2
9000H                   LD      (SYSCR2), X-1001 × 1B   ; Sets warm-up time to 2^14/fOSCH
9002H                   LD      (SYSCR0), 011000 - - B  ; Operates high-frequency after released.
9005H                   HALT
                  NMI
                                                        Clears and starts warm-up
                                                        timer
                                                        (high-frequency clock)
                                                            ↓
                                                          End
                                                            ↓
                                                        NMI Interrupt routine
                                                            ⋮
9006H                   LD      XX, XX                  RETI
                        ⋮
- : No change
```

Note: When different modes are used before and after STOP mode as the above mentioned, there is possible to release the HALT mode without changing the operation mode by acceptance of the halt release interrupt request during execution of HALT instruction (during 6 states). In the system which accepts the interrupts during execution HALT instruction, set the same operation mode before and after the STOP mode.

Table 3.3.7 Input Buffer State Table

Port Name	Input Function Name		Input Buffer State								
			During Reset	When the CPU is Operating		In HALT Mode (IDLE2/IDLE1)		In HALT mode (STOP)			
								<DRVE> = 1		<DRVE> = 0	
				When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port
P00 to P07	AD0 to AD7		OFF	ON upon external read	ON	OFF	OFF	OFF	OFF	OFF	OFF
P10 to P17	AD8 to AD15										
P20 to P27	–			–		–		–		–	
P32 (*1)	–										
P33 (*1)	$\overline{WAIT}$		ON	ON		OFF		OFF		OFF	
P34 (*1)	$\overline{BUSRQ}$					ON	ON	ON	ON		
P35 to P37 (*1)	–			–		–		–		–	
P40 to P43 (*1)	–										
P50 to P52, P54 to P57 (*2)	–		OFF		ON upon port read		OFF		OFF		
P53 (*2)	$\overline{ADTRG}$			ON		ON		ON		ON	
P60	SCK		ON		ON		ON		ON	OFF	
P61	SDA										
P62	SCL, SI										
P63	INT0									ON	ON
P64 to P66	–			–		–		–		–	OFF
P70	TA0IN			ON		ON		ON		OFF	
P73	TA4IN										
P71 to P72, P74 to P75	–			–		–		–		–	
P80	INT5, TB0IN0			ON		ON		ON		OFF	
P81	INT6, TB0IN1										
P84	INT7, TB1IN0										
P85	INT8, TB1IN1										
P82 to P83, P86 to P87	–			–		–		–		–	
P90, P93	–										
P91	RXD0			ON		ON		ON		OFF	
P92	SCLK0/$\overline{CTS0}$										
P94	RXD1										
P95	SCLK1/$\overline{CTS1}$										
P96	XT1	For oscillator	OFF		OFF		OFF	OFF	OFF		
		For port	ON	OFF		OFF	ON		ON		
P97	–			–		–		–		–	
PA0	INT1		OFF	ON	ON	ON	OFF	ON	OFF	ON	
PA1	INT2										
PA2	INT3										
PA3	INT4										
PA4 to PA7	–			–		–		–		–	
$\overline{NMI}$, $\overline{RESET}$, AM0, AM1	–		ON	ON	–	ON	–	ON	–	ON	–
X1	–					–	–	OFF	–	OFF	–

ON: The buffer is always turned on. A current flows through the input buffer if the input pin is not driven.
OFF: The buffer is always turned off.
– : Not applicable
*1: Port having a pull-up/pull-down resistor
*2: AIN input does not cause a current to flow through the buffer.

Table 3.3.8 Output Buffer State Table

Port Name	Input Function Name		Output Buffer State							
			During Reset	When the CPU is Operating		In HALT Mode (IDLE2/IDLE1)		In HALT mode (STOP)		
								<DRVE> = 1		<DRVE> = 0	
				When Used as Function Pin	When Used as Output Port	When Used as Function Pin	When Used as Output Port	When Used as Function Pin	When Used as Output Port	When Used as Function Pin	When Used as Output Port
P00 to P07	AD0 to AD7		OFF	ON upon external write	ON	OFF	ON	OFF	ON	OFF	OFF
P10 to P17	AD8 to AD15										
P20 to P27	A16 to A23			ON		ON		ON			
P30	$\overline{RD}$		ON								
P31	$\overline{WR}$										
P32 (*1)	$\overline{HWR}$		OFF								
P33 to P34, P37 (*1)	–			–		–		–		–	
P35 (*1)	$\overline{BUSAK}$			ON		ON		ON		OFF	
P36 (*1)	R/W										
P40 (*1)	$\overline{CS0}$										
P41 (*1)	$\overline{CS1}$										
P42 (*1)	$\overline{CS2}$										
P43 (*1)	$\overline{CS3}$										
P60	SCK										
P61	SDA, SO										
P62	SCL										
P63, P65 to P66	–			–	ON	–	ON	–	ON	–	OFF
P64	SCOUT			ON		ON		ON		OFF	
P70, P73	–			–		–		–		–	
P71	TA1OUT			ON		ON		ON		OFF	
P72	TA3OUT										
P74	TA5OUT										
P75	TA7OUT										
P80 to P81, P84 to P85	–			–		–		–		–	
P82	TB0OUT0			ON		ON		ON		OFF	
P83	TB0OUT1										
P86	TB1OUT0										
P87	TB1OUT1										
P90	TXD0										
P91, P94	–			–		–		–		–	
P92	SCLK0			ON		ON		ON		OFF	
P93	TXD1										
P95	SCLK1										
P96	–		ON	–		–		–		–	
P97	XT2	For oscillator	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	
		For port	ON	OFF	ON	OFF	ON		ON		
PA0 to PA7	–		OFF	–		–		–		–	
ALE	–			ON	–	ON	–	Output "L" level			
X2	–		ON					Output "H" level			

ON: The buffer is always turned on. When the bus is released, however, output buffers for some pins are turned off.
OFF: The buffer is always turned off.
– : Not applicable
*1: Port having a pull-up/pull-down resistor

3.4 Interrupts

Interrupts are controlled by the CPU interrupt mask register <IFF2:0> (Bits 14 to 12 of the status register) and by the built-in interrupt controller.
TMP91CW12 has a total of 45 interrupts divided into the following five types:

Interrupts generated by CPU: 9
- Software interrupts: 8
- Illegal instruction: 1

Internal interrupts: 26
- Internal I/O interrupts: 22
- Micro DMA transfer end interrupts: 4

External interrupts: 10
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0 to INT8)

A (fixed) individual interrupt vector number is assigned to each interrupt.
One of seven (Variable) priority levels can be assigned to each maskable interrupt. The priority level of non-maskable interrupts is fixed at 7, the highest level.
When an interrupt is generated, the interrupt controller sends the priority of that interrupt to the CPU. If multiple interrupts are generated simultaneously, the interrupt controller sends the interrupt with the highest priority to the CPU. (The highest priority possible is level 7, used for non-maskable interrupts.)
The CPU compares the priority level of the interrupt with the value of the CPU interrupt mask register <IFF2:0>. If the priority level of the interrupt is higher than the value of the interrupt mask register, the CPU accepts the interrupt. However, software interrupts and illegal instruction interrupts generated by the CPU are processed without comparison with the <IFF2:0> value.
The interrupt mask register <IFF2:0> value can be updated using the value of the EI instruction (executing EI num sets the content of <IFF2:0> to num). For example, specifying EI3 enables the acceptance of maskable interrupts whose priority level set in the interrupt controller is 3 or higher, and enables the acceptance of non-maskable interrupts. However, if EI or EI0 is specified, maskable interrupts with a priority level of 1 or higher and non-maskable interrupts are accepted (Operationally identical to EI1).
Operationally, the DI instruction (<IFF2:0> is 7) is identical to the EI7 instruction, but as the priority level of maskable interrupts is 0 to 6, the DI instruction is used to disable maskable interrupts. The EI instruction is valid immediately after execution begins. (with TLCS-90, the EI instruction is valid after execution of the instruction following the EI instruction.)

In addition to the general-purpose interrupt processing mode described above, TLCS-900/L1 interrupts have a micro DMA processing mode as well.
Because the CPU transfers data (Byte transfer, word transfer, or 4-byte transfer) automatically in micro DMA mode, this mode can be used for speeding up interrupt processing, such as transferring data to I/O.
TMP91CW12 also has a micro DMA soft start function for requesting micro DMA processing by software not by interrupt.
Figure 3.4.1 shows the overall interrupt processing flow.

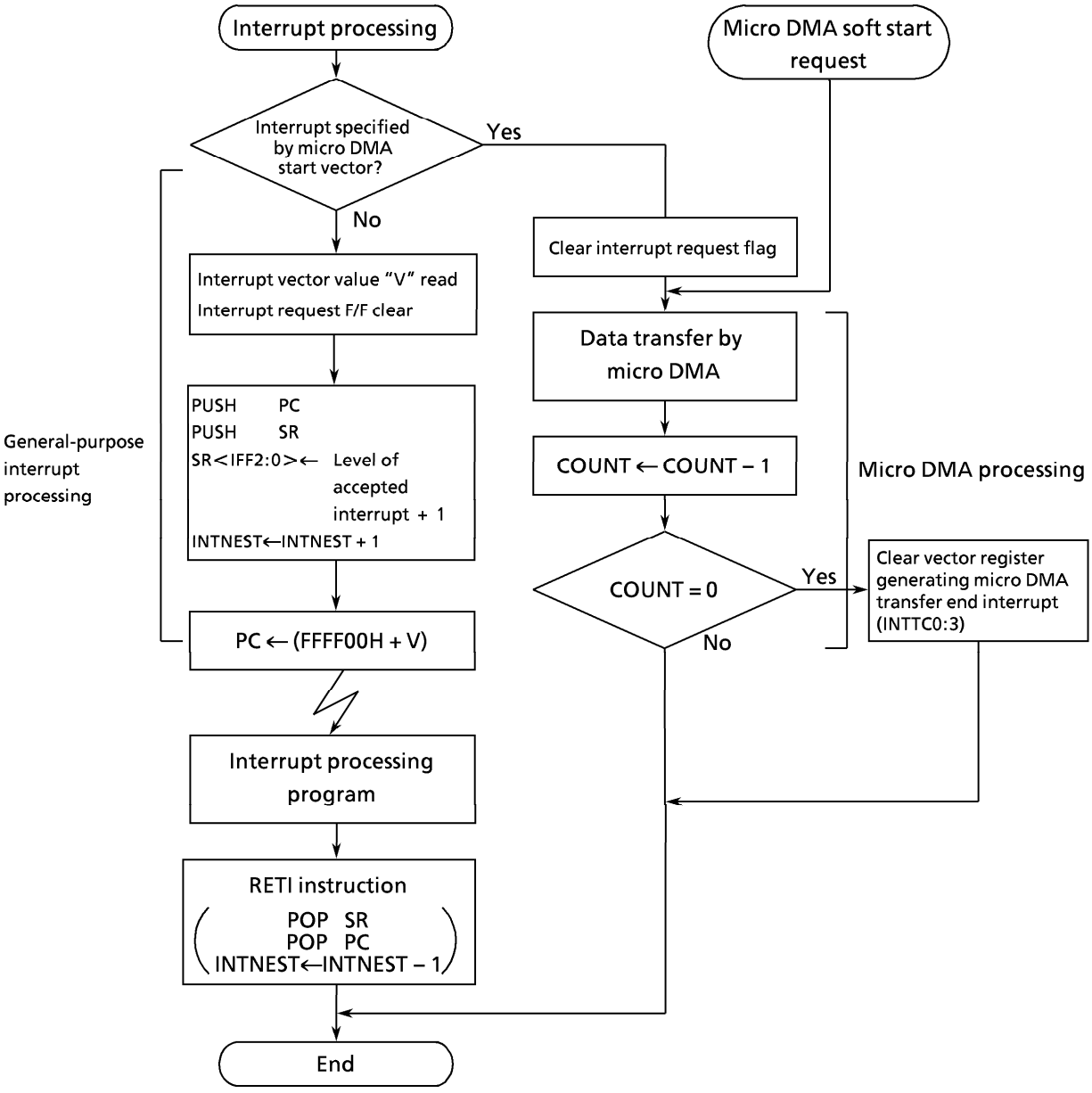


Figure 3.4.1 Interrupt and Micro DMA Processing Flow

3.4.1 General-purpose Interrupt Processing

When the CPU accepts an interrupt, the CPU performs the following processing. However, in the case of software interrupts and illegal instruction interrupts generated by the CPU, the CPU skips ① and ③ and executes steps ②, ④, and ⑤.

① The CPU reads the interrupt vector from the interrupt controller. If there are simultaneous interrupts set to the same level, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt request.
(The default priority is already fixed for each interrupt: the smaller the vector value, the higher the priority level.)

② The CPU saves the contents of the program counter (PC) and status register (SR) to the stack area (indicated by XSP).

③ The CPU sets the value of the CPU's interrupt mask register <IFF2:0> to the received interrupt level incremented by 1. However, if the incremented value level is 7 or higher, the CPU just sets the register to 7.

④ The CPU increments interrupt nesting counter INTNEST by 1.

⑤ The CPU jumps to the address indicated by the data at address FFFF00H + interrupt vector, and starts the interrupt processing routine.

The above proccesing time is 18 states (2.25 μs at 16 MHz) as the best case (16 bits data-bus width and 0 waits).
When the CPU completed the interrupt processing, use the RETI instruction to return to the main routine. This instruction restores the contents of the program counter and status register from the stack, and decrements interrupt nesting counter INTNEST by 1.
Non-maskable interrupts cannot be disabled by program. Maskable interrupts can be enabled or disabled by program. The program can set a priority level for every interrupt source. (Setting the priority level to 0 (or 7) disables the interrupt request.)

If a request is received for an interrupt with a higher priority level than that set in the CPU interrupt mask register <IFF2:0>, the CPU accepts the interrupt. Set the CPU interrupt mask register <IFF2:0> to the received interrupt priority level incremented by 1.
If, during interrupt processing, an interrupt is generated with a higher level than the interrupt being currently processed, or if, during non-maskable interrupt processing, a non-maskable interrupt request is generated from another source, the CPU suspends the currently processing routine and accepts the later interrupt. Then, after the CPU finished processing the later interrupt, the CPU returns to the interrupt it previously suspended and resumes processing.
If the CPU receives a request for another interrupt while performing processing in steps ① to ⑤, the second interrupt is sampled immediately after execution of the start instruction for its interrupt processing routine. Specifying DI as the start instruction disables maskable interrupt nesting. (Note: In the 900 and 900/L, sampling is performed before execution of the start instruction.)

After a reset, the interrupt mask register <IFF2:0> is initialized to 111, thus disabling maskable interrupts.

Table 3.4.1 shows the TMP91CW12 interrupt vectors and micro DMA start vectors. With the TMP91CW12, FFFF00H to FFFFFFH (256 bytes) is allocated to the interrupt vector area.

Table 3.4.1 TMP91CW12 Interrupt Vectors and Micro DMA Start Vectors

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value V	Vector Reference Address	Micro DMA Start Vector
1	Non-maskable	Reset or [SWI0] instruction	0000H	FFFF00H	–
2		[SWI1] instruction	0004H	FFFF04H	–
3		Illegal instruction or [SWI2] instruction	0008H	FFFF08H	–
4		[SWI3] instruction	000CH	FFFF0CH	–
5		[SWI4] instruction	0010H	FFFF10H	–
6		[SWI5] instruction	0014H	FFFF14H	–
7		[SWI6] instruction	0018H	FFFF18H	–
8		[SWI7] instruction	001CH	FFFF1CH	–
9		NMI: $\overline{\text{NMI}}$ pin input	0020H	FFFF20H	–
10		INTWD: Watchdog timer	0024H	FFFF24H	–
–	–	Micro DMA (Note)	–	–	–
11	Maskable	INT0: INT0 pin input	0028H	FFFF28H	0AH
12		INT1: INT1 pin input	002CH	FFFF2CH	0BH
13		INT2: INT2 pin input	0030H	FFFF30H	0CH
14		INT3: INT3 pin input	0034H	FFFF34H	0DH
15		INT4: INT4 pin input	0038H	FFFF38H	0EH
16		INT5: INT5 pin input	003CH	FFFF3CH	0FH
17		INT6: INT6 pin input	0040H	FFFF40H	10H
18		INT7: INT7 pin input	0044H	FFFF44H	11H
19		INT8: INT8 pin input	0048H	FFFF48H	12H
20		INTTA0: 8-bit timer 0	004CH	FFFF4CH	13H
21		INTTA1: 8-bit timer 1	0050H	FFFF50H	14H
22		INTTA2: 8-bit timer 2	0054H	FFFF54H	15H
23		INTTA3: 8-bit timer 3	0058H	FFFF58H	16H
24		INTTA4: 8-bit timer 4	005CH	FFFF5CH	17H
25		INTTA5: 8-bit timer 5	0060H	FFFF60H	18H
26		INTTA6: 8-bit timer 6	0064H	FFFF64H	19H
27		INTTA7: 8-bit timer 7	0068H	FFFF68H	1AH
28		INTTB00: 16-bit timer 0 (TB0RG0)	006CH	FFFF6CH	1BH
29		INTTB01: 16-bit timer 0 (TB0RG1)	0070H	FFFF70H	1CH
30		INTTB10: 16-bit timer 1 (TB1RG0)	0074H	FFFF74H	1DH
31		INTTB11: 16-bit timer 1 (TB1RG1)	0078H	FFFF78H	1EH
32		INTTBOF0: 16-bit timer 0 (Overflow)	007CH	FFFF7CH	1FH
33		INTTBOF1: 16-bit timer 1 (Overflow)	0080H	FFFF80H	20H
34		INTRX0: Serial receive (Channel 0)	0084H	FFFF84H	21H
35		INTTX0: Serial transmission (Channel 0)	0088H	FFFF88H	22H
36		INTRX1: Serial receive (Channel 1)	008CH	FFFF8CH	23H
37		INTTX1: Serial transmission (Channel 1)	0090H	FFFF90H	24H
38		INTSBI: Serial bus interface	0094H	FFFF94H	25H
39		INTRTC: Timer for realtime clock	0098H	FFFF98H	26H
40		INTAD: AD conversion end	009CH	FFFF9CH	27H
41		INTTC0: Micro DMA end (Channel 0)	00A0H	FFFFA0H	–
42		INTTC1: Micro DMA end (Channel 1)	00A4H	FFFFA4H	–
43		INTTC2: Micro DMA end (Channel 2)	00A8H	FFFFA8H	–
44		INTTC3: Micro DMA end (Channel 3)	00ACH	FFFFACH	–
–		(Reserved)	00B0H	FFFFB0H	–
to		to	to	to	to
–		(Reserved)	00FCH	FFFFFCH	–

Note: Micro DMA default priority
If an interrupt request is generated by a source specified by micro DMA, the interrupt has the highest priority of the maskable interrupts (Irrespective of the default priority allocated to all channels).

3.4.2 Micro DMA Processing

In addition to general-purpose interrupt processing, TMP91CW12 supports a micro DMA function. Interrupt requests set by the micro DMA perform micro DMA processing at the highest priority level of maskable interrupts (Level 6), regardless of the priority level of the particular interrupt source.
Because the function of micro DMA has been implemented with the cooperative operation of CPU, when CPU is a state of stand-by by HALT instruction, the requirement of micro DMA will be ignored (Pending).

(1) Micro DMA operation

When an interrupt request is generated by an interrupt source specified by the micro DMA start vector register, the micro DMA triggers a micro DMA request to the CPU at interrupt priority level 6 and starts processing the request. The four micro DMA channels allow micro DMA processing to be set for up to four types of interrupts at any one time.
When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared. The data are automatically transferred from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented by 1. If the decremented counter reads other than 0, DMA processing ends with no change in the value of the micro DMA start vector register. If the decremented reading is 0, the micro DMA transfer end interrupt (INTTC0 to INTTC3) passes from the CPU to the interrupt controller. In addition, the micro DMA start vector register is cleared to 0, the next micro DMA is disabled, and micro DMA processing completes.
If a micro DMA request is set for more than one channel at a time, the priority is not based on the interrupt priority level but on the channel number: the smaller the channel number the higher the priority. (Channel 0 (High) --> Channel 3 (Low)).
If an interrupt request is triggered for the interrupt source in use during the interval between the clearing of the micro DMA start vector and the next setting, general-purpose interrupt processing executes at the interrupt level set. Therefore, if only using the interrupt for starting the micro DMA (Not using the interrupt as a general-purpose interrupt), first set the interrupt level to 0 (Interrupt requests disabled).
If using micro DMA and general-purpose interrupts together as described above, first set the level of the interrupt used to start micro DMA processing lower than all the other interrupt levels. In this case, the cause of general interrupt is limited to the edge interrupt.
Like other maskable interrupts, the priority of the micro DMA transfer end interrupt is determined by the interrupt level and the default priority.

While the register for setting the transfer source/transfer destination addresses is a 32-bit control register, this register can only effectively output 24-bit addresses. Accordingly, micro DMA can access 16 Mbytes (The upper eight bits of the 32 bits are not valid).
Three micro DMA transfer modes are supported: 1-byte transfer, 2-byte (One word) transfer, and 4-byte transfer. After a transfer in any mode, the transfer source/destination addresses are incremented, decremented, or remain unchanged. This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the transfer modes, see 3.4.2 (4), Transfer mode register.
As the transfer counter is a 16-bit counter, micro DMA processing can be set for up to 65536 times per interrupt source. (The micro DMA processing count is maximized when the transfer counter initial value is set to 0000H.)
Micro DMA processing can be started by the 30 interrupts shown in the micro DMA start vectors of Table 3.4.1 and by the micro DMA soft start, making a total of 31 interrupts.
Figure 3.4.2 shows the word transfer micro DMA cycle in transfer destination address INC mode (except for COUNTER mode, the same as for other modes). (The conditions for this cycle are based on an external 16-bit bus, 0 waits, transfer source/transfer destination addresses both even-numbered values).

① Word transfer

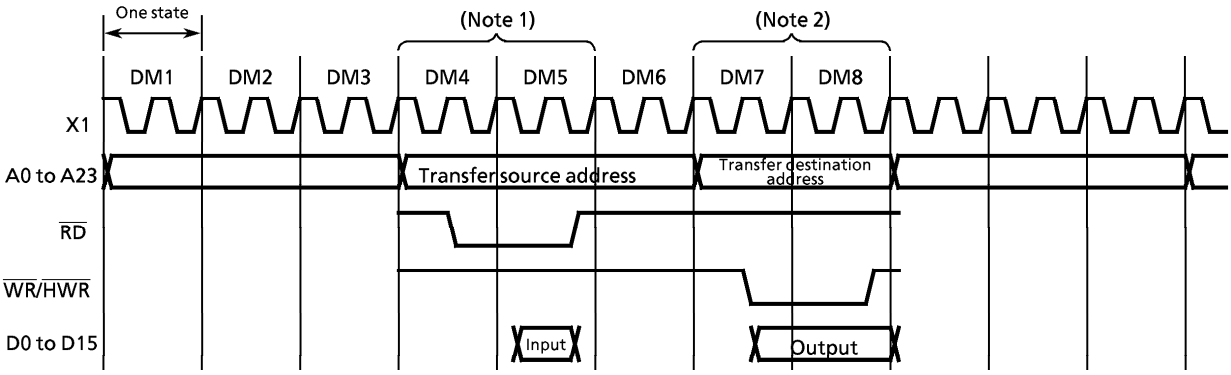


Figure 3.4.2 Timing of Micro DMA Cycle

States 1 to 3: Instruction fetch cycle (Gets next address code).
If three or more instruction codes are inserted in the instruction queue buffer, this cycle becomes a dummy cycle.

States 4 to 5: Micro DMA read cycle
State 6: Dummy cycle (The address bus remains as in state 5)
States 7 to 8: Micro DMA write cycle

Note 1: If the source address area is an 8-bit bus, it is incremented by two states.
If the source address area is a 16-bit bus and the address starts from an odd number, it is incremented by two states.

Note 2: If the destination address area is an 8-bit bus, it is incremented by two states.
If the destination address area is a 16-bit bus and the address starts from an odd number, it is incremented by two states.

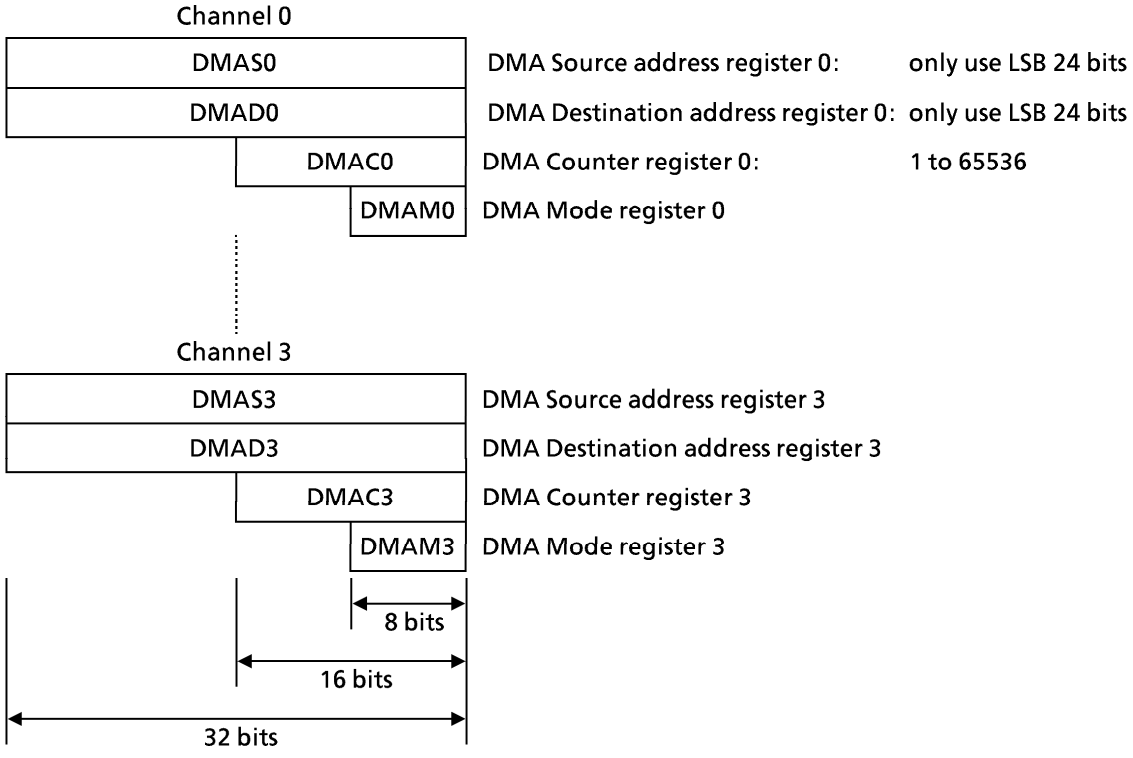
(2) Software start function

In addition to starting the micro DMA function by interrupts, TMP91CW12 includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the DMAR register.
Writing 1 to each bit of DMAR register causes micro DMA once. At the end of transfer, the bits of the DMAR register which support the end channel are automatically cleared to 0.
Only one-channel can be set for DMA request at once. (Do not write 1 to more than single bit.)
When writing again 1 to the DMAR register, check whether the bit is 0 before writing 1.
When the burst is specified by DMAB register, data is continuously transferred until the value in the micro DMA transfer counter is 0 after startup of the micro DMA.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request register	89H					DMA request			
							DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							0	0	0	0

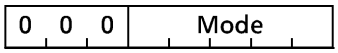
(3) Transfer control register

The transfer source address and the transfer destination address are set by the following registers. These registers set data using "LDC cr,r" instruction.



(4) Detailed description of the transfer mode register: DMAM0 to DMAM3

(DMAM0 to DMAM3)



Note: When setting a value in this register, write 0 to the upper 3 bits.

ZZ: 0 = Byte transfer, 1 = Word transfer,
2 = 4-byte transfer, 3 = Reserved

						Execution time
0	0	0	Z	Z	Transfer destination address INC mode . . . I/O to memory (DMADn +) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	8 states (1000 ns) at byte/ word transfer 12 states (1500 ns) at 4-byte transfer
0	0	1	Z	Z	Transfer destination address DEC mode . . I/O to memory (DMADn-) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	8 states (1000 ns) at byte/ word transfer 12 states (1500 ns) at 4-byte transfer
0	1	0	Z	Z	Transfer source address INC mode . . . . . Memory to I/O (DMADn) ← (DMASn +) DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	8 states (1000 ns) at byte/ word transfer 12 states (1500 ns) at 4-byte transfer
0	1	1	Z	Z	Transfer source address DEC mode . . . . . Memory to I/O (DMADn) ← (DMASn –) DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	8 states (1000 ns) at byte/ word transfer 12 states (1500 ns) at 4-byte transfer
1	0	0	Z	Z	Address fixed mode . . . . . I/O to I/O (DMADn) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	8 states (1000 ns) at byte/ word transfer 12 states (1500 ns) at 4-byte transfer
1	0	1	0	0	Counter mode for counting number of times interrupt is generated DMACn ← DMASn + 1 DMACn ← DMACn – 1 if DMACn = 0 then INTTC is generated	5 states (625 ns)

Note 1: "n" is the corresponding micro DMA channels 0 to 3
DMADn + /DMASn + : Post-increment (Increment register value after transfer)
DMADn-/DMASn – : Post-decrement (Decrement register value after transfer)
The I/Os in the table mean fixed address and the memory means increment (INC) or decrement (DEC) addresses.

Note 2: Execution time is under the condition of:
16-bit bus width/0 waits.
fc = 16 MHz/selected high-frequency mode (fc × 1)

Note 3: Do not use an undefined code for the transfer mode register except for the defined codes listed in the above table.

3.4.3 Interrupt Controller Operation

Figure 3.4.3 is a block diagram of the interrupt circuit. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each interrupt channel (36 channels in total), an interrupt request flag (flip-flop), an interrupt priority setting register, and a micro DMA start vector register. The interrupt request flag latches interrupt request from the peripherals. The flag is cleared to zero in the following cases: when reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), when the micro DMA burst transfer is terminated, and when an instruction that clears the interrupt for that channel is executed (by writing 0 to the clear bit in the interrupt priority setting register).

The interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTE0AD, INTE12). Six interrupt priorities from 1 to 6 are provided. Setting 0 (or 7) disables the interrupt request. The priority of non-maskable interrupts (NMI pin, watchdog timer) is fixed at 7. If interrupt requests with the same level are generated at the same time, the default priority (the interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request to accept first.

Reading the 3rd bit and the 7th bit in the interrupt priority setting register sees the state of the interrupt request flag and whether there are the interrupt request of each channel.

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2:0> set in the status register by the interrupt request signal with the priority value set; if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 in the CPU SR <IFF2:0>. Interrupt request where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine.

When interrupt processing is completed (after execution of the RETI instruction), the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR<IFF2:0>.

The interrupt controller also has four registers used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.4.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter register (e.g., DMAS and DMAD) prior to the micro DMA processing.

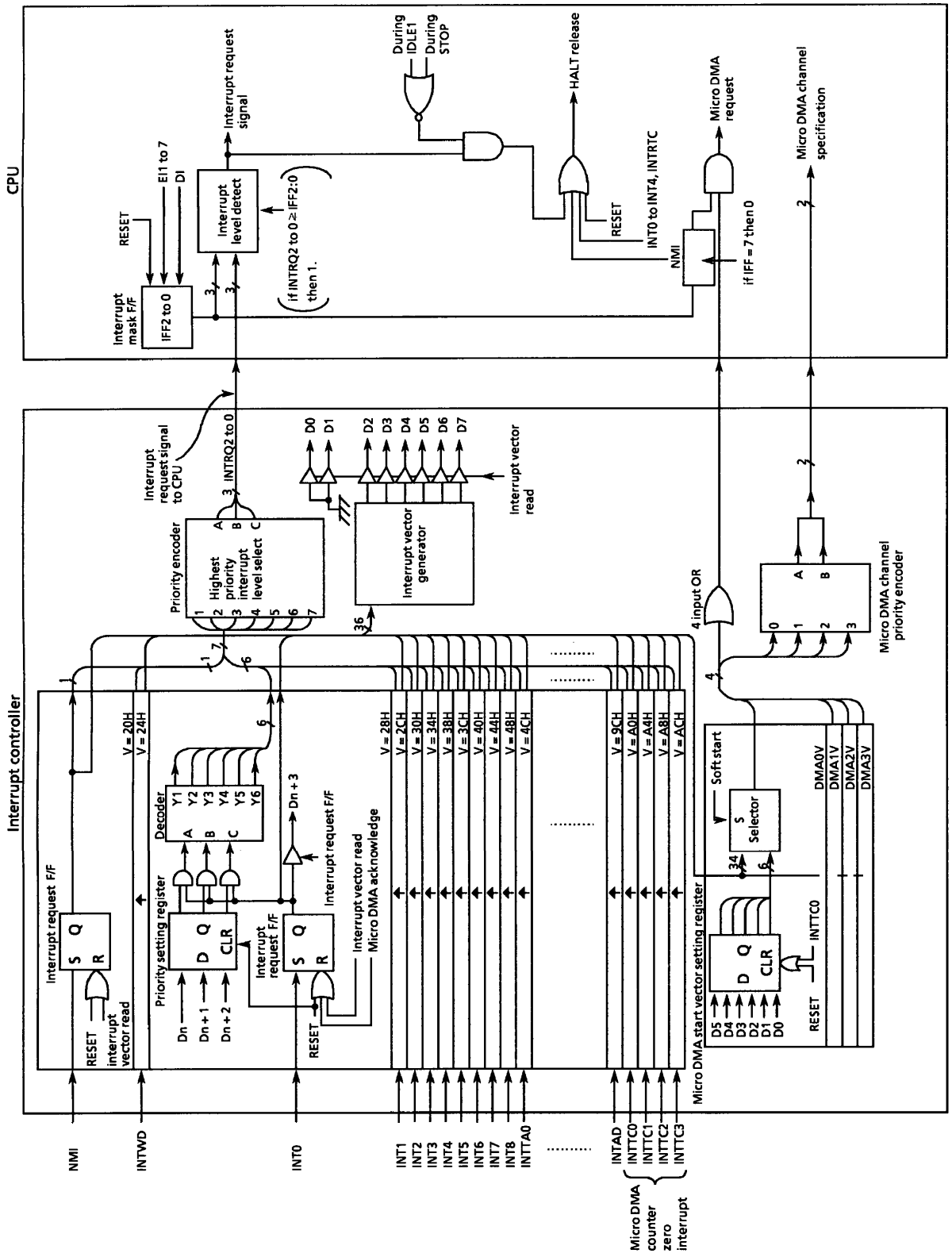


Figure 3.4.3 Block Diagram of Interrupt Controller

(1) Interrupt level setting register

Symbol	Address	7	6	5	4	3	2	1	0	
INTE0AD	90H	INTAD				INT0				←Interrupt source
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0	←Bit symbol
		R	R/W			R	R/W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
INTE12	91H	INT2				INT1				
		I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTE34	92H	INT4				INT3				
		I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTE56	93H	INT6				INT5				
		I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTE78	94H	INT8				INT7				
		I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETA01	95H	INTTA1 (TMRA1)				INTTA0 (TMRA0)				
		ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETA23	96H	INTTA3 (TMRA3)				INTTA2 (TMRA2)				
		ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETA45	97H	INTTA5 (TMRA5)				INTTA4 (TMRA4)				
		ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETA67	98H	INTTA7 (TMRA7)				INTTA6 (TMRA6)				
		ITA7C	ITA7M2	ITA7M1	ITA7M0	ITA6C	ITA6M2	ITA6M1	ITA6M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt request

Interrupt request flag

Symbol	Address	7	6	5	4	3	2	1	0	
INTETB0	99H	INTTB01 (TMRB0)				INTTB00 (TMRB0)				←Interrupt source
		ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0	←Bit symbol
		R	R/W			R	R/W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
INTETB1	9AH	INTTB11 (TMRB1)				INTTB10 (TMRB1)				
		ITB11C	ITB11M2	ITB11M1	ITB11M0	ITB10C	ITB10M2	ITB10M1	ITB10M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETB01V	9BH	INTTBOF1 (TMRB1) overflow				INTTBOF0 (TMRB0) overflow				
		ITF1C	ITF1M2	ITF1M1	ITF1M0	ITF0C	ITF0M2	ITF0M1	ITF0M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTES0	9CH	INTTX0				INTRX0				
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTES1	9DH	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTES2RTC	9EH	INTRTC				INTSBI				
		IRTCC	IRTCM2	IRTCM1	IRTCM0	ISBIC	ISBIM2	ISBIM1	ISBIM0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETC01	A0H	INTTC1				INTTC0				
		ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	
INTETC23	A1H	INTTC3				INTTC2				
		ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0	
		R	R/W			R	R/W			
		0	0	0	0	0	0	0	0	

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt request

Interrupt request flag

(2) External interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt input mode control	8CH (Prohibit RMW)	–	I4EDGE	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W							
			0	0	0	0	0	0	0	0
			Write"0"	INT4EDGE 0: Rising 1: Falling	INT3EDGE 0: Rising 1: Falling	INT2EDGE 0: Rising 1: Falling	INT1EDGE 0: Rising 1: Falling	INT0EDGE 0: Rising 1: Falling	0: INT0 edge mode 1: INT0 level mode	1: Operate even at $\overline{NMI}$ rise edge

INT0 level Enable

0	Edge detect INT
1	H level INT

$\overline{NMI}$ rising edge Enable

0	INT request generation at falling edge
1	INT request generation at rising/falling edge

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the micro DMA start vector, which is listed in table 3.4.1, to the INTCLR register.
For example, to clear the INT0 interrupt flag, operate the following register after execution of DI instruction.

INTCLR ← 0AH Clears INT0 interrupt request flag

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt clear control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					0	0	0	0	0	0
					Interrupt vector					

(4) Micro DMA start vector register

This register assigns micro DMA processing to an interrupt source. The interrupt source with a micro DMA start vector that matches the vector set in this register is assigned as the micro DMA start source. When the micro DMA transfer counter value reaches zero, the micro DMA transfer end interrupt corresponding to the channel is set to the interrupt controller, the micro DMA start vector register is cleared, and the micro DMA start source of the channel is cleared. Therefore, to continue micro DMA processing, set the micro DMA start vector register again during the processing of the micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vector registers of more than one channel, the channel with the lowest number has a higher priority.
Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until the micro DMA transfer is complete. If the micro DMA start vector of this channel is not set again, the next micro DMA is started for the channel with the higher number (Micro DMA chaining).

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 start vector	80H			DMA0 start vector					
					DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
DMA1V	DMA 1 start vector	81H			DMA1 start vector					
					DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
DMA2V	DMA 2 start vector	82H			DMA2 start vector					
					DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
DMA3V	DMA 3 start vector	83H			DMA3 start vector					
					DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0

(5) Micro DMA burst specification

Specifying the micro DMA burst continues the micro DMA transfer until the transfer counter register reaches zero after micro DMA start. Setting a bit which corresponds to the micro DMA channel of the DMAB registers mentioned below to 1 specifies a burst.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA software request register	89H					DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							1: DMA software request			
							0	0	0	0
DMAB	DMA burst register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W			
							0	0	0	0

(6) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag (Note) between accepting and reading the interrupt vector. In this case, the CPU reads the default vector 0008H and reads the interrupt vector at address FFFF08H.
To avoid the above problem, place instructions that clear interrupt request flags after a DI instruction. And in the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing and more than 1 instruction (e.g., "NOP" * 1 time). If placed EI instruction without waiting NOP instruction after execution of clearing instruction, interrupt will be enable before request flag is cleared.
In the case of changing the value of the interrupt mask register <IFF2:0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.
In addition, take care as the following 2 circuits are exceptional and demand special attention.

INT0 level mode	INT0 in level mode is not an edge-detect interrupt, so the interrupt request flip-flop function is canceled. The peripheral interrupt request bypasses the S input of the flip-flop, and acts as the Q output. Changing modes from edge to level automatically clears the interrupt request flag.
	If the CPU enters the interrupt response sequence as a result of setting INT0 from 0 to 1, INT0 must be held at 1 until the interrupt response sequence is completed. If the INT0 level mode is used to release a halt, INT0 must be held at 1 from the time INT0 changes from 0 to 1, to the time when the halt is released. (Ensure that INT0 does not go back 0 due to noise before the halt is released.)
	When switching modes from level to edge, any interrupt request flag set in level mode is not cleared. Accordingly, clear the interrupt request flag using the following sequence.
	
	DI
	LD (IIMC), 00H ; Switches from level to edge.
	LD (INTCLR), 0AH ; Clears interrupt request flag.
	NOP ; Wait EI instruction
	EI

INTRX	The interrupt request flip-flop can only be cleared by reset or by reading the serial channel receive buffer, not by an instruction.

Note: The following instructions or pin changes are equivalent to instructions that clear the interrupt request flag.
INT0: Instructions that switch to level mode after an interrupt request is generated in edge mode.
The pin input changes from high to low after an interrupt request is generated in level mode. (H → L)
INTRX: Instructions that read the receive buffer.

3.5 Functions of Ports

The TMP91CW12 has 81 bits for I/O ports.
These port pins have I/O functions for the built-in CPU and internal I/Os as well as general-purpose I/O port functions. Table 3.5.1 lists the function of each port pin. Table 3.5.2 lists I/O registers and specification.

Table 3.5.1 Functions of Ports (R: ↑ = with programmable pull-up resistor)

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port 0	P00 to P07	8	I/O	–	Bit	AD0 to AD7
Port 1	P10 to P17	8	I/O	–	Bit	AD8 to AD15/A8 to A15
Port 2	P20 to P27	8	I/O	–	Bit	A16 to A23/A0 to A7
Port 3	P30	1	Output	–	Bit	$\overline{RD}$
	P31	1	Output	–	Bit	$\overline{WR}$
	P32	1	I/O	↑	Bit	$\overline{HWR}$
	P33	1	I/O	↑	Bit	$\overline{WAIT}$
	P34	1	I/O	↑	Bit	$\overline{BUSRQ}$
	P35	1	I/O	↑	Bit	$\overline{BUSAK}$
	P36	1	I/O	↑	Bit	R/$\overline{W}$
	P37	1	I/O	↑	Bit	
Port 4	P40	1	I/O	↑	Bit	$\overline{CS0}$
	P41	1	I/O	↑	Bit	$\overline{CS1}$
	P42	1	I/O	↑	Bit	$\overline{CS2}$
	P43	1	I/O	↑	Bit	$\overline{CS3}$
Port 5	P50 to P57	8	Input	–	(Fixed)	AN0 to AN7 $\overline{ADTRG}$ (P53)
Port 6	P60	1	I/O	–	Bit	SCK
	P61	1	I/O	–	Bit	SO/SDA
	P62	1	I/O	–	Bit	SI/SCL
	P63	1	I/O	–	Bit	INT0
	P64	1	I/O	–	Bit	SCOUT
	P65	1	I/O	–	Bit	
	P66	1	I/O	–	Bit	
Port 7	P70	1	I/O	–	Bit	TA0IN
	P71	1	I/O	–	Bit	TA1OUT
	P72	1	I/O	–	Bit	TA3OUT
	P73	1	I/O	–	Bit	TA4IN
	P74	1	I/O	–	Bit	TA5OUT
	P75	1	I/O	–	Bit	TA7OUT
Port 8	P80	1	I/O	–	Bit	TB0IN0/INT5
	P81	1	I/O	–	Bit	TB0IN1/INT6
	P82	1	I/O	–	Bit	TB0OUT0
	P83	1	I/O	–	Bit	TB0OUT1
	P84	1	I/O	–	Bit	TB1IN0/INT7
	P85	1	I/O	–	Bit	TB1IN1/INT8
	P86	1	I/O	–	Bit	TB1OUT0
	P87	1	I/O	–	Bit	TB1OUT1
Port 9	P90	1	I/O	–	Bit	TXD0
	P91	1	I/O	–	Bit	RXD0
	P92	1	I/O	–	Bit	SCLK0/$\overline{CTS0}$
	P93	1	I/O	–	Bit	TXD1
	P94	1	I/O	–	Bit	RXD1
	P95	1	I/O	–	Bit	SCLK1/$\overline{CTS1}$
	P96	1	I/O	–	Bit	XT1
	P97	1	I/O	–	Bit	XT2
Port A	PA0 to PA3	4	I/O	–	Bit	INT1 to INT4
	PA4 to PA7	4	I/O	–	Bit	

Table 3.5.2 I/O Registers and Specification (1/2)

Port	Name	Specification	I/O Register		
			Pn	PnCR	PnFC
Port 0	P00 to P07	Input port	×	0	None
		Output port	×	1	
		AD (0 to 7) bus	×	×	
Port 1	P10 to P17	Input port	×	0	0
		Output port	×	1	0
		AD (8 to 15) bus	×	0	1
		A (8 to 15) output	×	1	1
Port 2	P20 to P27	Input port	×	0	0
		Output port	×	1	0
		A (0 to 7) output	×	0	1
		A (16 to 23) output	×	1	1
Port 3	P30	Output port	×	None	0
		Outputs $\overline{RD}$ only when accessing external space	1		1
		Always outputs $\overline{RD}$	0		1
	P31	Output port	×	None	0
		Outputs $\overline{WR}$ only when accessing external space	×		1
	P32 to P37	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P32	$\overline{HWR}$ output	×	1	1
	P33	$\overline{WAIT}$ input (without PU)	0	0	None
		$\overline{WAIT}$ input (with PU)	1	0	
	P34	$\overline{BUSRQ}$ input (without PU)	0	0	1
		$\overline{BUSRQ}$ input (with PU)	1	0	1
	P35	$\overline{BUSAK}$ output	×	1	1
	P36	R/$\overline{W}$ output	×	1	1
Port 4	P40 to P43	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P40	$\overline{CS0}$ output	×	1	1
	P41	$\overline{CS1}$ output	×	1	1
	P42	$\overline{CS2}$ output	×	1	1
	P43	$\overline{CS3}$ output	×	1	1
Port 5	P50 to P57	Input port	×	None	
		AN (0 to 7) input (Note 1)	×		
	P53	$\overline{ADTRG}$ input (Note 2)	×		
Port 6	P60 to P67	Input port	×	0	0
		Output port	×	1	0
	P60	SCK input	×	0	0
		SCK output	×	1	1
	P61	SDA input	×	0	0
		SDA output (Note 3)	×	1	1
		SO output	×	1	1

X: Don't care

Note 1: When P50 to P57 are used as AD converter input channels, AD mode control register <ADCH2:0> is used to select the channel.

Note 2: When P53 is used as $\overline{ADTRG}$ input, ADMOD1<ADTRGE> is used to enable External trigger input.

Note 3: When P61/P62 are used as SDA/SCL open-drain output, ODE<ODE62:61> is used to set the open-drain output mode.

Table 3.5.2 I/O Registers and Specification (2/2)

Port	Name	Specification	I/O Register		
			Pn	PnCR	PnFC
Port 6	P62	SI input	×	0	0
		SCL input	×	0	0
		SCL output (Note 3)	×	1	1
	P63	INT0 input	×	0	1
	P64	SCOUT output	×	1	1
Port 7	P70 to P75	Input port	×	0	0
		Output port	×	1	0
	P70	TA0IN input	×	0	None
	P71	TA1OUT output	×	1	1
	P72	TA3OUT output	×	1	1
	P73	TA4IN input	×	0	None
	P74	TA5OUT output	×	1	1
	P75	TA7OUT output	×	1	1
Port 8	P80 to P87	Input port	×	0	0
		Output port	×	1	0
	P80	TB0IN0, INT5 input	×	0	1
	P81	TB0IN1, INT6 input	×	0	1
	P82	TB0OUT0 output	×	1	1
	P83	TB0OUT1 output	×	1	1
	P84	TB1IN0, INT7 input	×	0	1
	P85	TB1IN1, INT8 input	×	0	1
	P86	TB1OUT0 output	×	1	1
	P87	TB1OUT1 output	×	1	1
Port 9	P90 to P95	Input port	×	0	0
		Output port	×	1	0
	P90	TXD0 output	×	1	1
	P91	RXD0 input	×	0	None
	P92	SCLK0 input	×	0	0
		SCLK0 output	×	1	1
		$\overline{\text{CTS0}}$ input	×	0	0
	P93	TXD1 output	×	1	1
	P94	RXD1 input	×	0	None
	P95	SCLK1 input	×	0	0
		SCLK1 output	×	1	1
		$\overline{\text{CTS1}}$ input	×	0	0
	P96 to P97	Input port	×	0	None
		Output port (Note 4)	×	1	
		XT1 to XT2 (Note 5)	×	0	
Port A	PA0 to PA7	Input port	×	0	0
		Output port	×	1	0
	PA0	INT1 input	×	0	1
	PA1	INT2 input	×	0	1
	PA2	INT3 input	×	0	1
	PA3	INT4 input	×	0	1

X: Don't care
Note 4: Using P96/P97 as output port, output is through the open drain buffer.
Note 5: Using P96/P97 as XT1/XT2, SYSCR0 register has to be set enable oscillation.

Resetting makes the port pins listed below function as general-purpose I/O ports.
I/O pins programmable for input or output are set to input ports except P96/XT1, P97/XT2.
To set port pins for built-in functions, a program is required.

Note about the bus release and programmable pull-up I/O ports.

When the bus is released ($\overline{BUSAK} = 0$), the output buffer of AD0 to AD15, A0 to A23, control signal ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$ to $\overline{CS3}$) is off and their state become high-impeadance.
However, the output of built-in programmable pull-up resistors are kept before the bus is released. These programmable pull-up resistors can be selected ON/OFF by programmable when they are used as the input ports.
The case of they are used as the output ports, they can not be selected ON/OFF by programmable.
Table 3.5.3 shows the pin state when the bus is released.

Table 3.5.3 The pin State (when the bus is released)

Pin Name	The Pin State (when the bus is released)	
	Used as the Port	Used as the Function
P00 to P07 (AD0 to AD7) P10 to P17 (AD8 to AD15/A8 to A15)	The state is not changed. (do not become to high impedance (High-Z).)	Become high impedance (High-Z).
P20 to P27 (A16 to A23)	The state is not changed. (do not become to high impedance (High-Z).)	First sets all bits to high, then sets them to high impedance (High-Z).
P30 ($\overline{RD}$) P31 ($\overline{WR}$)	↑	↑
P32 ($\overline{HWR}$)	↑	The output buffer is OFF. The programmable pull-up resistor is ON irrespective of the output latch.
P36 (R/$\overline{W}$) P40 ($\overline{CS0}$) P41 ($\overline{CS1}$) P42 ($\overline{CS2}$) P43 ($\overline{CS3}$)	↑	↑

Figure 3.5.1 shows the example of the external interface circuit the case of the bus releasing function is used.
When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate.
So, the watchdog timer also continues to run. Therefore, be careful about bus releasing time and setting the detection time of the WDT.

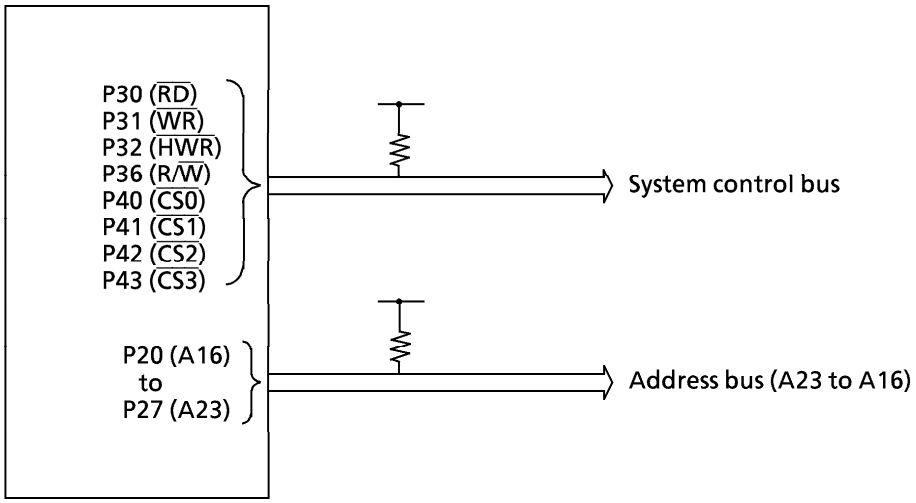


Figure 3.5.1 Example of the Interface Circuit (The case of using bus releasing function)

The above circuit is necessary to fix the signal level in the case of the bus is released.

Resetting sets P30 ($\overline{RD}$), P31 ($\overline{WR}$) to output, P40 ($\overline{CS0}$), P41 ($\overline{CS1}$), P42 ($\overline{CS2}$), P43 ($\overline{CS3}$), P32 ($\overline{HWR}$), P36 (R/$\overline{W}$) and P35 ($\overline{BUSAK}$) to input with pull-up resistor.

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using the control register P0CR. Resetting resets all bits of P0CR to 0 and sets port 0 to input mode.
In addition to functioning as a general-purpose I/O port, port 0 also functions as an address data bus (AD0 to AD7). To access external memory, port 0 functions as an address data bus (AD0 to AD7) and all bits of the control register P0CR are cleared to 0.

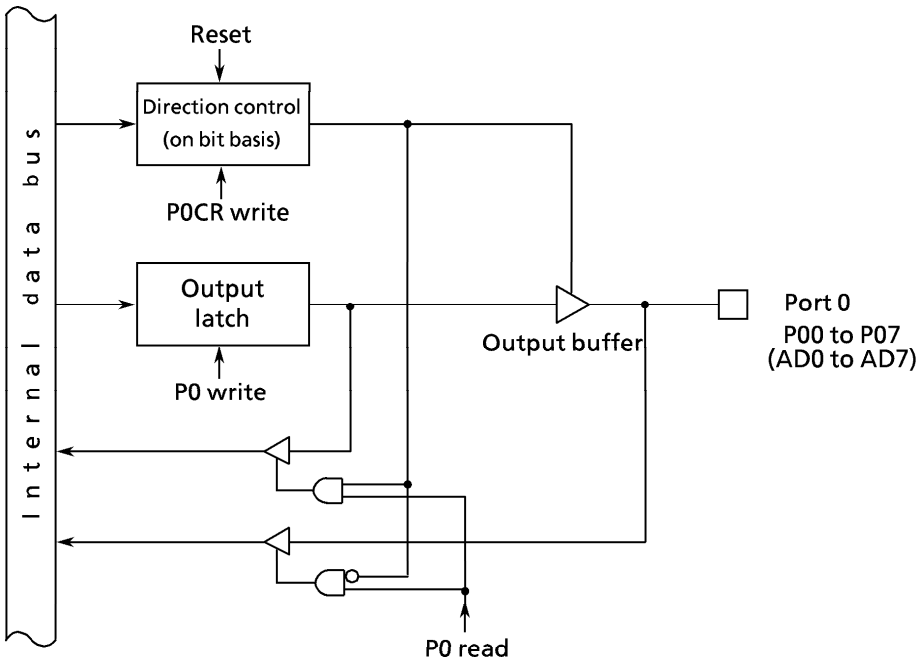


Figure 3.5.2 Port 0

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR and function register P1FC. Resetting resets all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets port 1 to input mode.
In addition to functioning as a general-purpose I/O port, port 1 also functions as an address data bus (AD8 to AD15) or an address bus (A8 to A15).

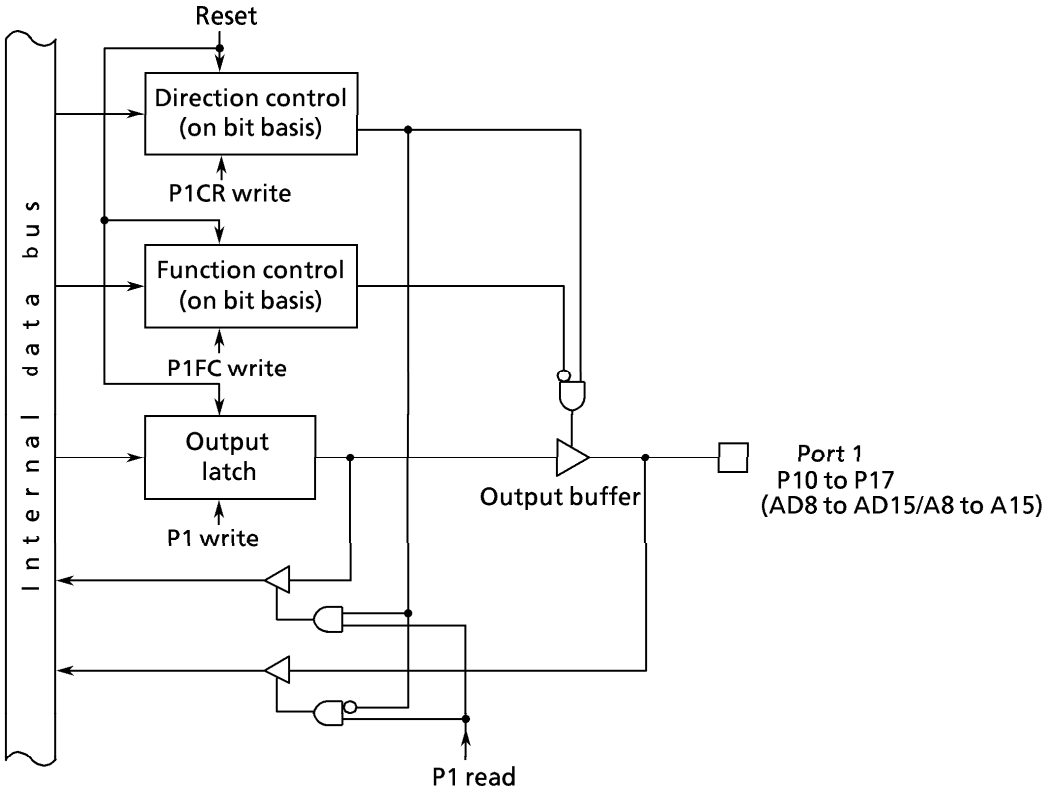


Figure 3.5.3 Port 1

Port 0 Register

P0 (0000H)

	7	6	5	4	3	2	1	0
Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
Read/Write	R/W							
After reset	Data from external port (Output latch register becomes undefined.)							

Port 0 Control Register

P0CR (0002H)

	7	6	5	4	3	2	1	0
Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0:Input 1:Output (At external access, port 0 becomes AD7 to AD0 and P0CR is cleared to 0.)							

Port 0 I/O setting

0	Input
1	Output

Port 1 Register

P1 (0001H)

	7	6	5	4	3	2	1	0
Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	Data from external port (Output latch register is cleared to "0".)							

Port 1 Control Register

P1CR (0004H)

	7	6	5	4	3	2	1	0
Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P1FC below.>>							

Port 1 Function Register

P1FC (0005H)

	7	6	5	4	3	2	1	0
Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00: Input, 01: Output, 10: AD15 to AD8, 11: A15 to A8							

Port 1 function setting

P1CR<P1XC> \ P1FC<P1XF>	0	1
0	Input port	Address data bus (AD15 to AD8)
1	Output port	Address bus (A15 to A8)

Read-modify-write is prohibited for registers P0CR, P1CR, and P1FC.

Note: <P1XF> is bit X in register P1FC; <P1XC>, in register P1CR.

Figure 3.5.4 Registers for Ports 0 and 1

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P2CR and function register P2FC. Resetting set all bits of output latch P2 to "1", and control register P2CR and function register P2FC to "0". It also sets port 2 to input mode.
In addition to functioning as a general-purpose I/O port, port 2 also functions as an address data bus (A0 to A7) and an address bus (A16 to A23).

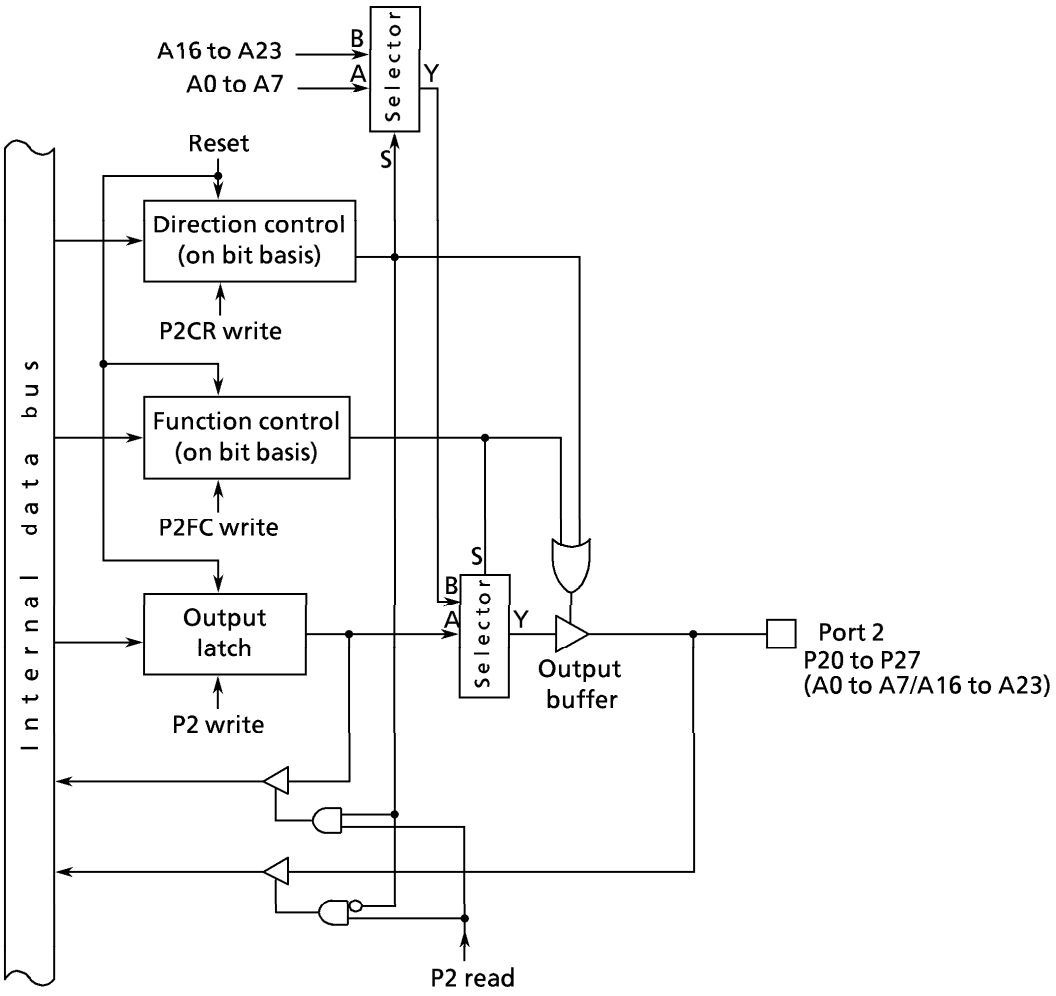


Figure 3.5.5 Port 2

Port 2 Register

P2 (0006H)

	7	6	5	4	3	2	1	0
Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to 1.)							

Port 2 Control Register

P2CR (0008H)

	7	6	5	4	3	2	1	0
Bit symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P2FC below.>>							

Port 2 Function Register

P2FC (0009H)

	7	6	5	4	3	2	1	0
Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P2FC/P2CR = 00: Input, 01: Output, 10: A7 to A0, 11: A23 to A16							

Note: Read-modify-write is prohibited for registers P2CR and P2FC.

Port 2 function setting

P2CR<P2XC> \ P2FC<P2XF>	0	1
0	Input port	Address data bus (A7 to A0)
1	Output port	Address bus (A23 to A16)

Note: <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR.
To set as an address bus A23 to A16, set P2FC after setting P2CR.

Figure 3.5.6 Registers for Port 2

3.5.4 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose I/O port.
I/O can be set on a bit basis, but note that P30 and P31 are used for output only. I/O is set using control register P3CR and function register P3FC. Resetting set all bits of output latch P3 to "1", and control register P3CR (Bits 0 and 1 are unused), and function register P3FC to 0. Resetting also outputs 1 from P30 and P31, sets P32 to P37 to input mode, and connects a pull-up resistor.
In addition to functioning as a general-purpose I/O port, port 3 also functions as an I/O for the CPU's control/status signal.
When P30 pin is defined as $\overline{RD}$ signal output mode (<P30F> = 1), clearing the output latch register <P30> to 0 outputs the $\overline{RD}$ strobe (used for the pseudo static RAM) from the P30 pin even when the internal address area is accessed.
If the output latch register <P30> remains 1, the $\overline{RD}$ strobe signal is output only when the external address area is accessed.

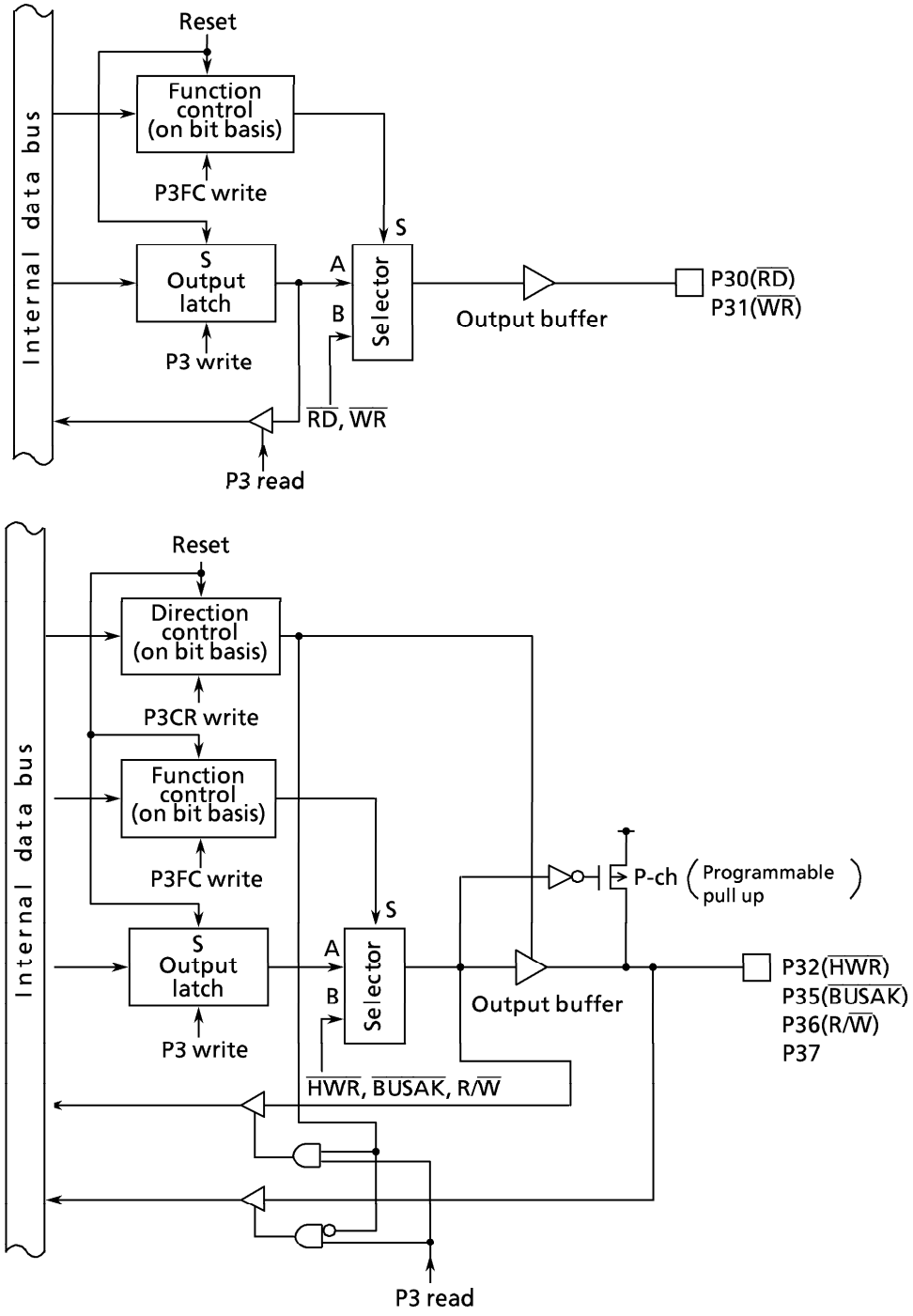


Figure 3.5.7 Port 3 (P30, P31, P32, P35, P36, P37)

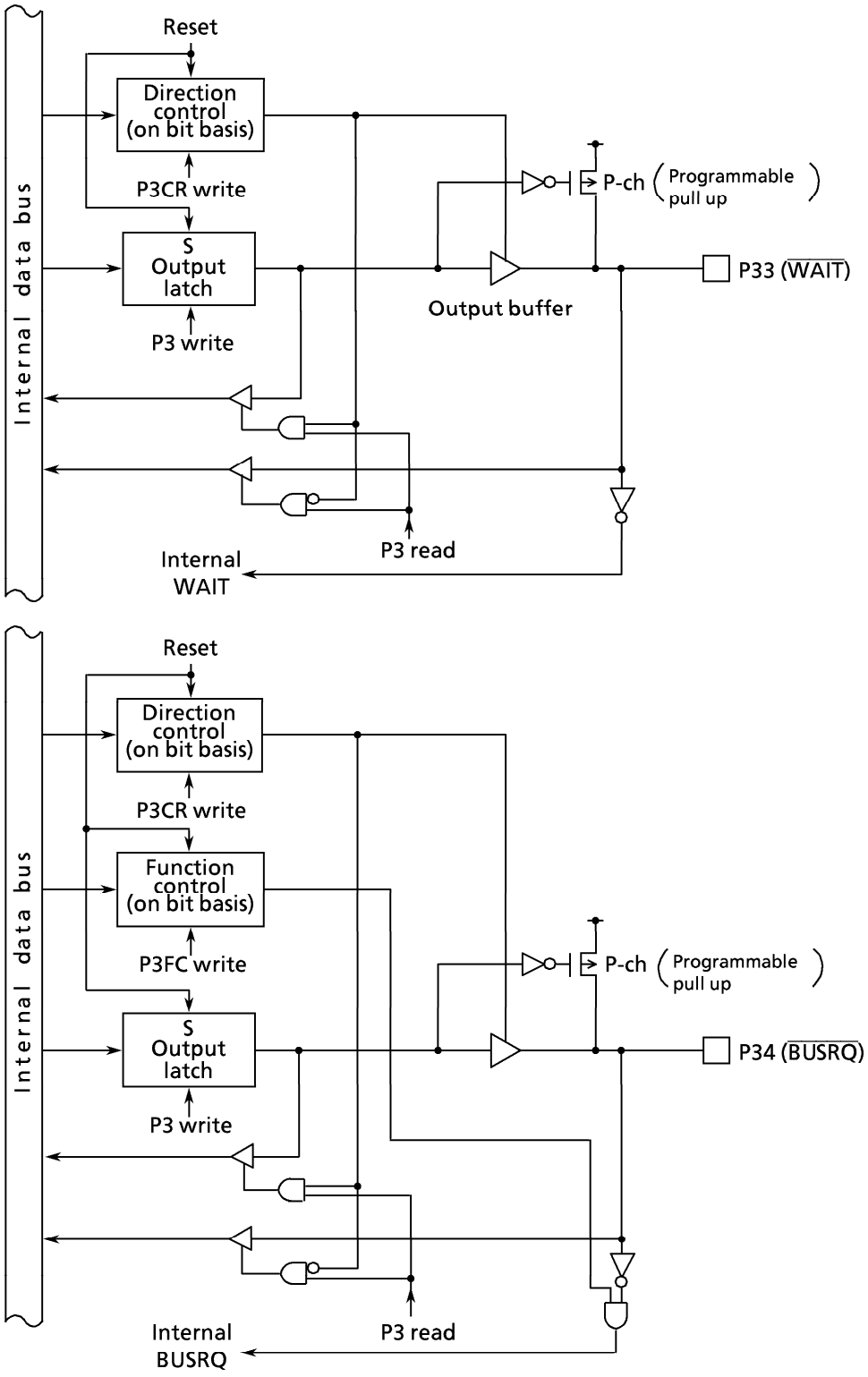


Figure 3.5.8 Port 3 (P33, P34)

Port 3 Register

P3 (0007H)

	7	6	5	4	3	2	1	0
Bit symbol	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to "1")						1	1
Function	0: Pull-up resistor OFF 1: Pull-up resistor ON						–	–

Port 3 Control Register

P3CR (000AH)

	7	6	5	4	3	2	1	0
Bit symbol	P37C	P36C	P35C	P34C	P33C	P32C		
Read/Write	W							
After reset	0	0	0	0	0	0		
	0: Input 1: Output							

I/O setting

0	Input
1	Output

Port 3 Function Register

P3FC (000BH)

	7	6	5	4	3	2	1	0
Bit symbol	–	P36F	P35F	P34F		P32F	P31F	P30F
Read/Write	W					W		
After reset	0	0	0	0		0	0	0
Function	(Note) Always write 0	0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$

$\overline{BUSRQ}$ setting

P3FC <P34F>	1
P3CR <P34C>	0

$\overline{BUSAK}$ setting

P3FC <P35F>	1
P3CR <P35C>	1

R/$\overline{W}$ setting

P3FC <P36F>	1
P3CR <P36C>	1

P30 ($\overline{RD}$) function setting

<P30> / <P30F>	0	1
0	0 output	1 output
1	Always $\overline{RD}$ output (for pseudo SRAM)	$\overline{RD}$ output only for external access

P31 ($\overline{WR}$) function setting

<P31> / <P31F>	0	1
0	0 output	1 output
1	$\overline{WR}$ output only for external access	

$\overline{HWR}$ setting

P3FC <P32F>	1
P3CR <P32C>	1

Note 1: Read-modify-write is prohibited for registers P3CR and P3FC.

Note 2: When port P3 is used in the input mode, P3 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: When P33/$\overline{WAIT}$ pin is used as a $\overline{WAIT}$ pin, set P3CR<P33C> to 0 and Chip Select/WAIT control register<BnW2:0> to 010.

Figure 3.5.9 Registers for Port 3

3.5.5 Port 4 (P40 to P43)

Port 4 is a 4-bit general-purpose I/O port. I/O can be set on a bit basis using control register P4CR and function register P4FC. Resetting does the following:

- Sets all bits of the output latch register P4 to 1.
- Resets all bits of the control register P4CR, and the function register P4FC to 0.
- Sets P40, P41, P42 and P43 to input mode and connects a pull-up resistor.

In addition to functioning as a general-purpose I/O port, port 4 also functions as a chip select output signal ($\overline{CS0}$ to $\overline{CS3}$).

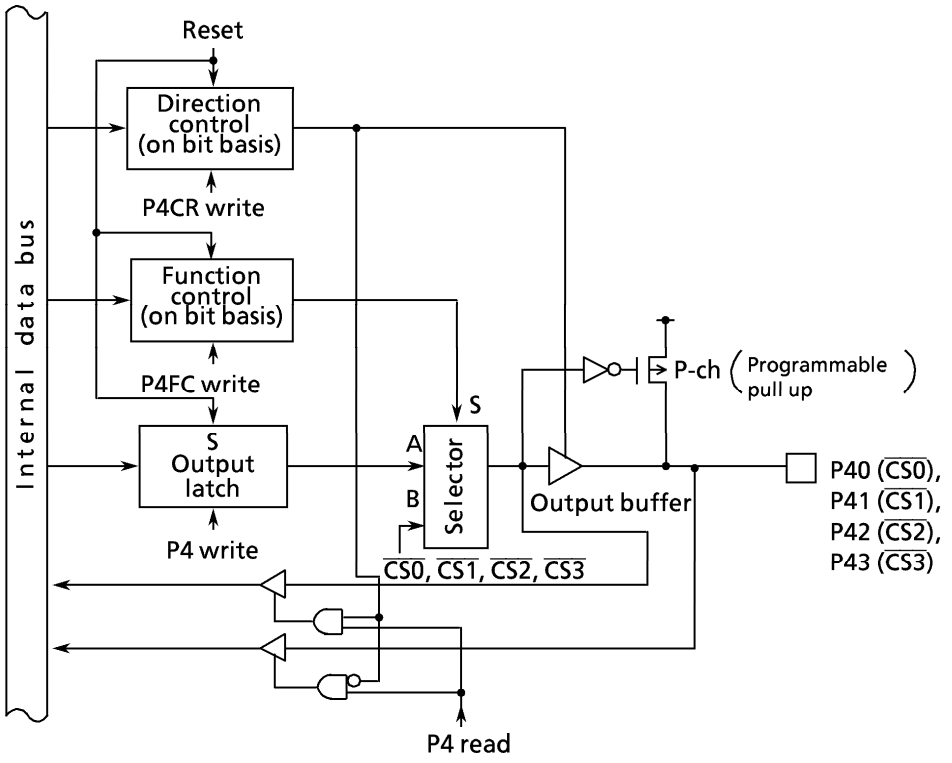


Figure 3.5.10 Port 4

Port 4 Register

P4 (000CH)

	7	6	5	4	3	2	1	0
Bit symbol					P43	P42	P41	P40
Read/Write					R/W			
After reset					Data from exteranal port (Output latch register is set to "1")			
Function					0: Pull-up resistor OFF 1: Pull-up resistor ON			

Port 4 Control Register

P4CR (000EH)

	7	6	5	4	3	2	1	0
Bit symbol					P43C	P42C	P41C	P40C
Read/Write					W			
After reset					0	0	0	0
Function					0 : Input 1 : Output			

I/O setting

0	Input
1	Output

Port 4 Function Register

P4FC (000FH)

	7	6	5	4	3	2	1	0
Bit symbol					P43F	P42F	P41F	P40F
Read/Write					W			
After reset					0	0	0	0
Function					0 : Port 1 : $\overline{CS}$			

0	Port (P40)
1	$\overline{CS0}$

0	Port (P41)
1	$\overline{CS1}$

0	Port (P42)
1	$\overline{CS2}$

0	Port (P43)
1	$\overline{CS3}$

Note 1: Read-modify-write is prohibited for registers P4CR and P4FC.
Note 2: When port P4 is used in the input mode, P4 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.
Note 3: To output chip select signal ($\overline{CS0}$ to $\overline{CS3}$), set the corresponding bits to 1 of the control register P4CR after setting the function register P4FC.

Figure 3.5.11 Registers for Port 4

3.5.6 Port 5 (P50 to P57)

Port 5 is an 8-bit input port, also used as an analog input pin for the internal AD converter.

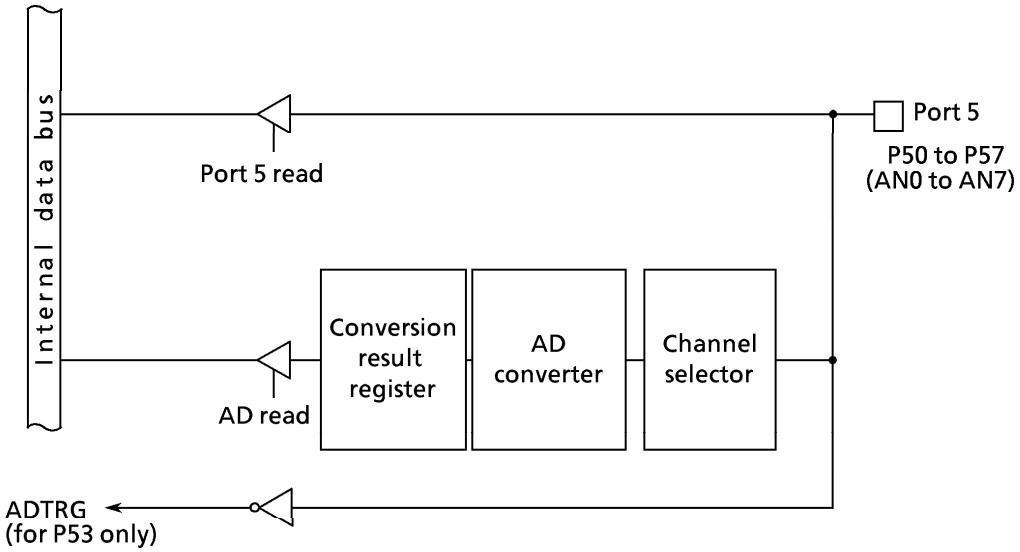


Figure 3.5.12 Port 5

Port 5 Register

P5 (000DH)

	7	6	5	4	3	2	1	0
Bit symbol	P57	P56	P55	P54	P53	P52	P51	P50
Read/Write	R							
After reset	Data from external port							

Note: The input channel selection of AD converter and the permission of ADTRG input are set by AD converter mode register ADMOD1.

Figure 3.5.13 Register for Port 5

3.5.7 Port 6 (P60 to P66)

Port 60 to 66 are an 7-bit general-purpose I/O ports. I/O can be set on bit basis. Resetting sets port 6 as an input port. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, port 60 to 66 also function as serial channels I/O functions. Writing 1 in the corresponding bit of the port 6 function register (P6FC) enables the respective functions.
Resetting resets the function register P6CR and P6FC to 0, and sets all bits to input ports.

(1) Port 60 (SCK)

Port 60 is a general-purpose I/O port. It is also used as a SCK I/O pin for serial channel (SIO mode).

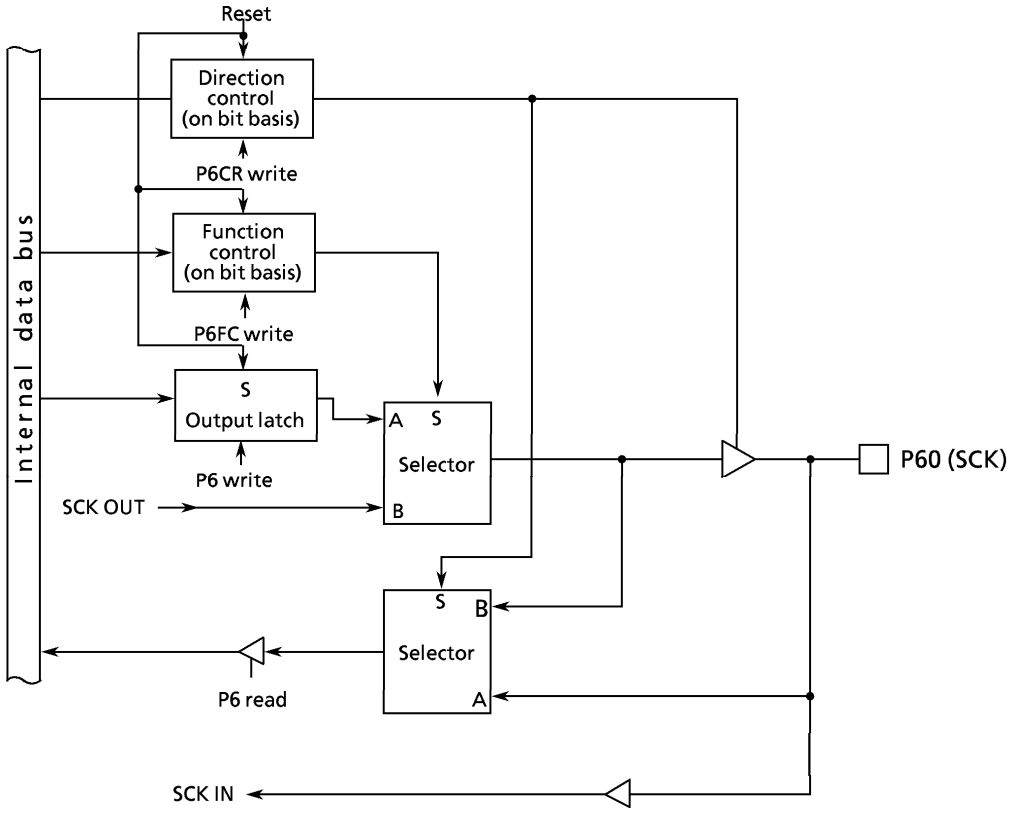


Figure 3.5.14 Port 60

(2) Port 61 (SO/SDA)

Port 61 is a general-purpose I/O port. It is also used as a SDA input pin or (I^2C bus mode) or SO output pin (SIO mode) for serial channel.

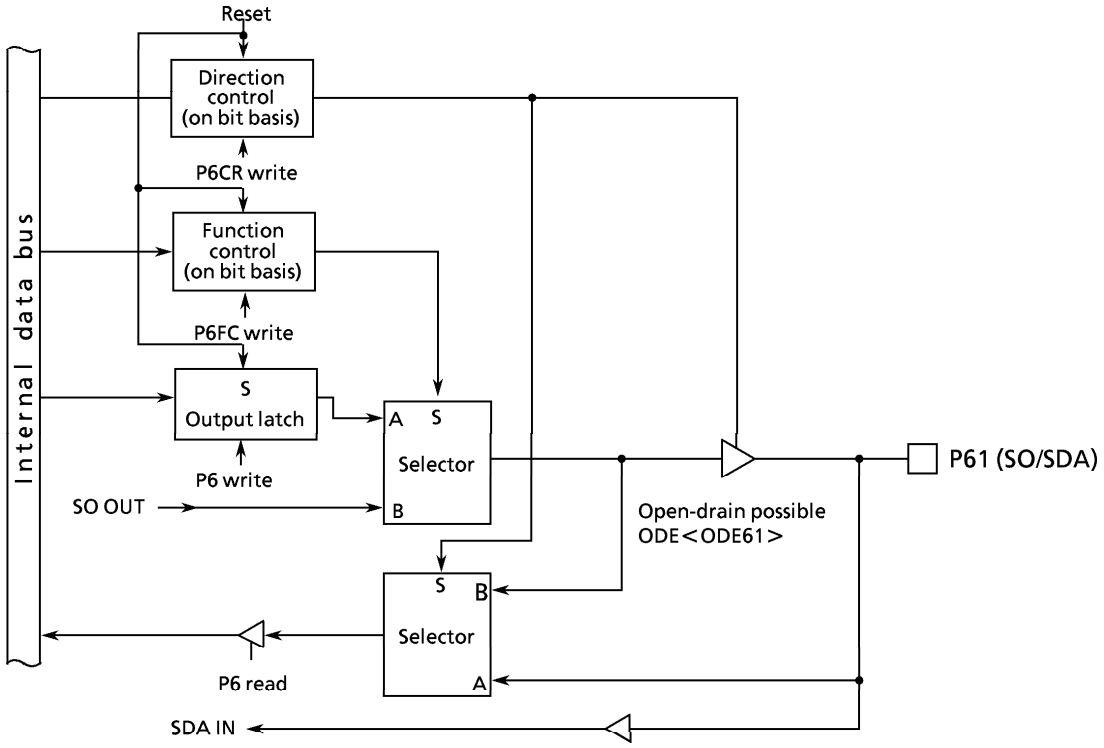


Figure 3.5.15 Port 61

(3) Port 62 (SI/SCL)

Port 62 is a general-purpose I/O port. It is also used as a data input (SIO mode) or SCL I/O pin (I^2C bus mode) for serial channel.

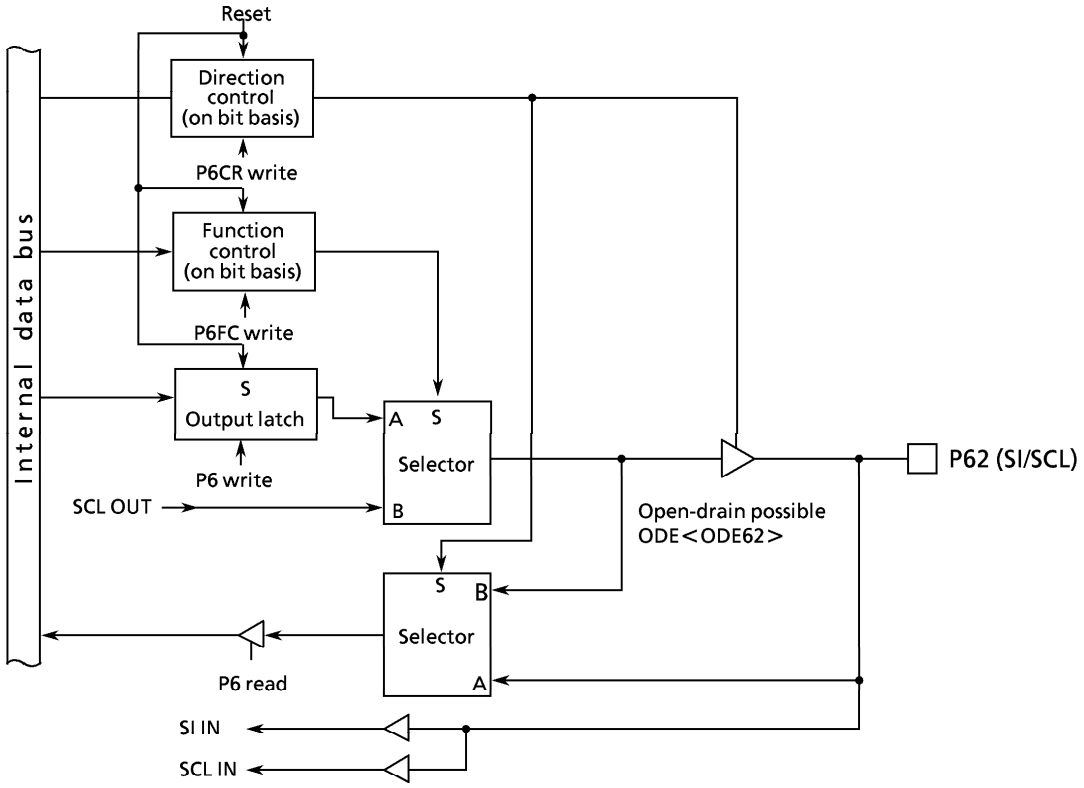


Figure 3.5.16 Port 62

(4) Port 63 (INT0)

Port 63 is a general-purpose I/O port. It is also used as a INT0 input pin.

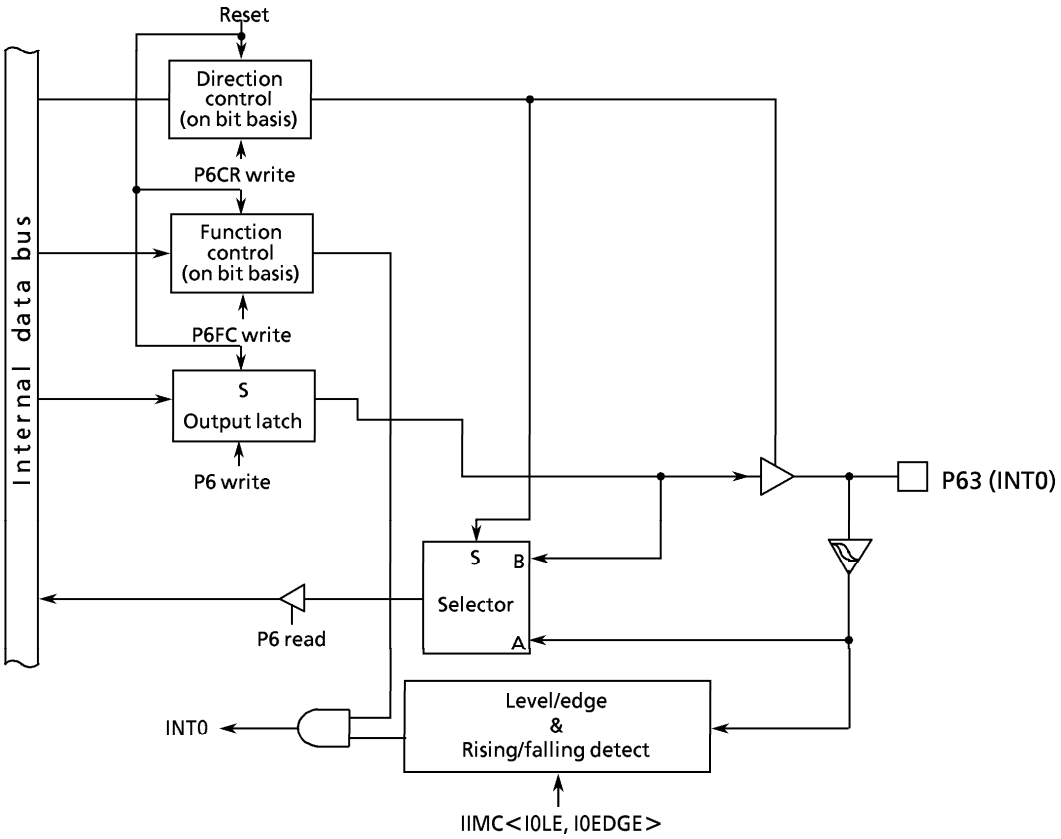


Figure 3.5.17 Port 62

(5) Port 64 (SCOUT)

Port 64 is a general-purpose I/O port. It is also used as a SCOUT output pin.

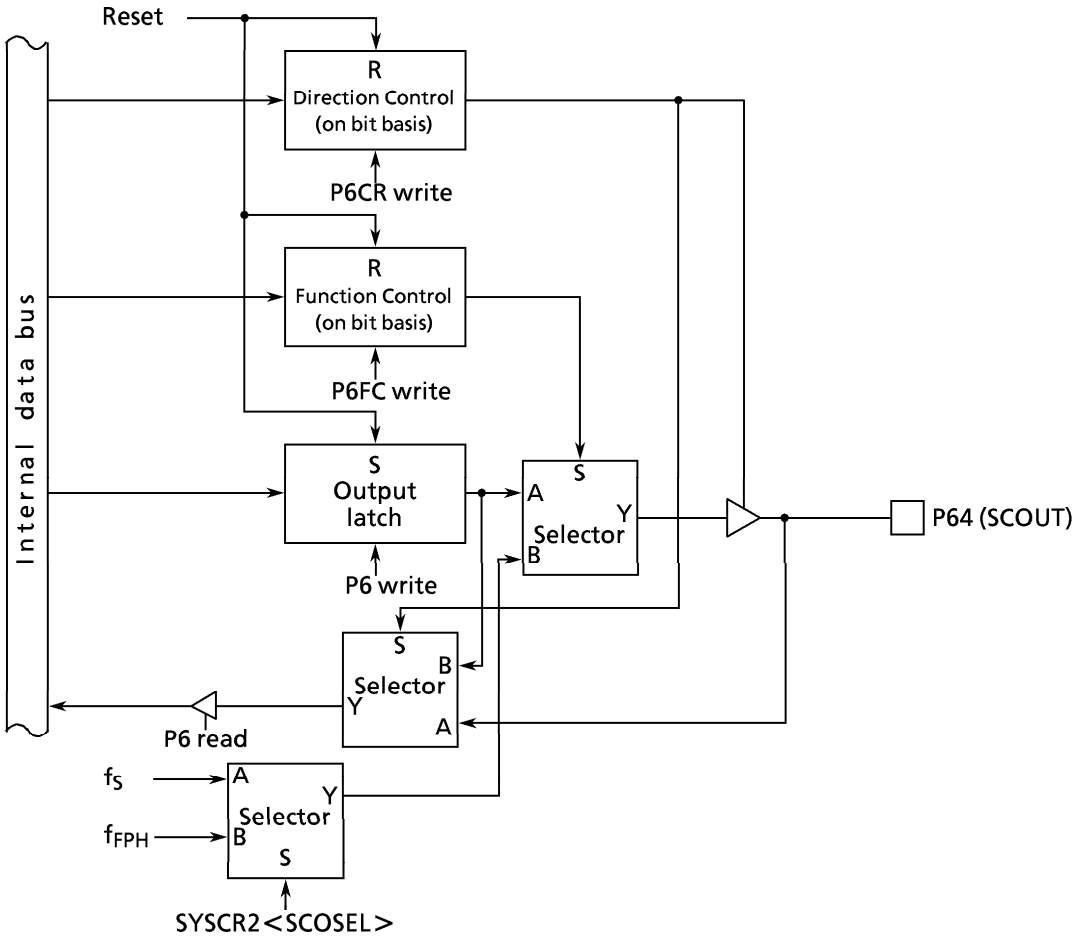


Figure 3.5.18 Port 64

(6) Port 65, 66

Port 65 and 66 are general-purpose I/O ports.

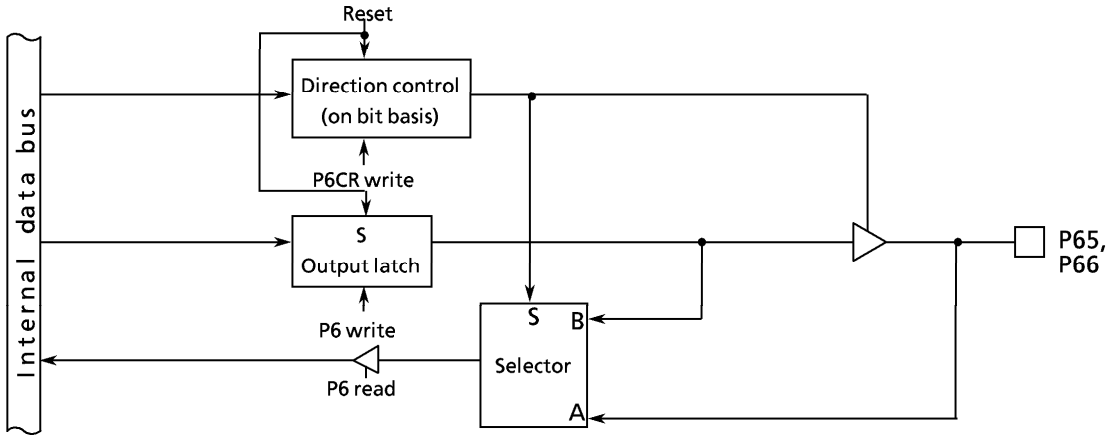


Figure 3.5.19 Port 65, 66

Port 6 Register

P6 (0012H)

	7	6	5	4	3	2	1	0
Bit symbol		P66	P65	P64	P63	P62	P61	P60
Read/Write		R/W						
After reset		Data from external port (Output latch register is set to "1")						

Port 6 Control Register

P6CR (0014H)

	7	6	5	4	3	2	1	0
Bit symbol		P66C	P65C	P64C	P63C	P62C	P61C	P60C
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		0: Input 1: Output						

Port 6 I/O setting

0	Input
1	Output

Port 6 Function Register

P6FC (0015H)

	7	6	5	4	3	2	1	0
Bit symbol				P64F	P63F	P62F	P61F	P60F
Read/Write				W				
After reset				0	0	0	0	0
Function				0: Port 1: SCOUT	0: Port 1: INT0	0: Port 1: SCL	0: Port 1: SDA/SO	0: Port 1: SCK

Note: Read-modify-write is prohibited for registers P6CR and P6FC.

P60 SCK output setting

P6FC <P60F>	1
P6CR <P60C>	1

P61 SDA/SO output setting

P6FC <P61F>	1
P6CR <P61C>	1

P62 SCL output setting

P6FC <P62F>	1
P6CR <P62C>	1

P63 INT0 input setting

P6FC <P63F>	1
P6CR <P63C>	0

P64 SCOUT output setting

P6FC <P64F>	1
P6CR <P64C>	1

Figure 3.5.20 Registers for Port 6

3.5.8 Port 7 (P70 to P75)

Port 7 is a 6-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets port 7 as an input port. In addition to functioning as a general-purpose I/O port, port 70 also functions as an input clock pin TA0IN of an 8-bit timer 0, port 71 as an 8-bit timer output (TA1OUT), port 72 as (TA3OUT), P73 as (TA4IN), P74 as (TA5OUT), P75 as (TA7OUT). Writing 1 in the corresponding bit of the port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7CR, P7FC to 0, and sets all bits to input ports.

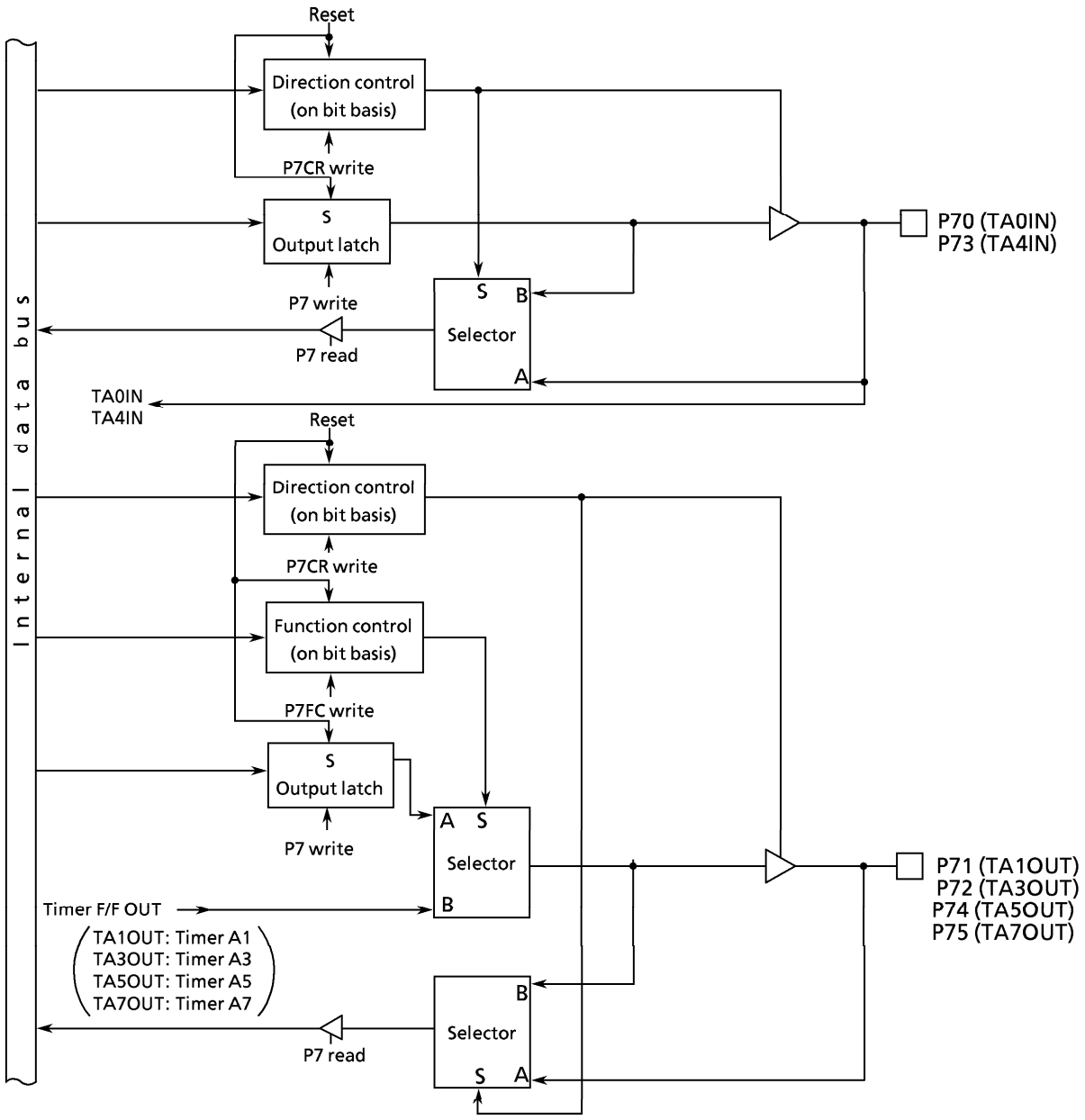


Figure 3.5.21 Port 7

Port 7 Register

P7 (0013H)

	7	6	5	4	3	2	1	0
Bit symbol			P75	P74	P73	P72	P71	P70
Read/Write			R/W					
After reset			External pin data (Output latch register is set to "1")					

Port 7 Control Register

P7CR (0016H)

	7	6	5	4	3	2	1	0
Bit symbol			P75C	P74C	P73C	P72C	P71C	P70C
Read/Write			W					
After reset			0	0	0	0	0	0
Function			0: Input 1: Output					

Port 7 I/O setting

0	Input
1	Output

Port 7 Function Register

P7FC (0017H)

	7	6	5	4	3	2	1	0
Bit symbol			P75F	P74F		P72F	P71F	
Read/Write			W			W		
After reset			0	0		0	0	
Function			0: Port 1: TA7OUT	0: Port 1: TA5OUT		0: Port 1: TA3OUT	0: Port 1: TA1OUT	

Note 1: Read-modify-write is prohibited for registers P7CR and P7FC.

Note 2: P70/TA0IN, P73/TA4IN pin does not have a register changing PORT/FUNCTION. For example, when it is used as an input port, the input signal is inputted to 8-bit timer .

Setting P71 as timer output 1

P7FC <P71F>	1
P7CR <P71C>	1

Setting P72 as timer output 3

P7FC <P72F>	1
P7CR <P72C>	1

Setting P74 as timer output 5

P7FC <P74F>	1
P7CR <P74C>	1

Setting P75 as timer output 5

P7FC <P75F>	1
P7FC <P75C>	1

Figure 3.5.22 Registers for Port 7

3.5.9 Port 8 (P80 to P87)

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets port 8 as an input port. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, port 8 also functions as an input for 16-bit timer clocks, an output for 16-bit timer F/F output, and an input for INT5 to INT8. Writing 1 in the corresponding bit of the port 8 function register (P8FC) enables those functions. Resetting resets the function register P8CR, P8FC to 0 and sets all bits to input ports.

(1) P80, P81, P84, P85

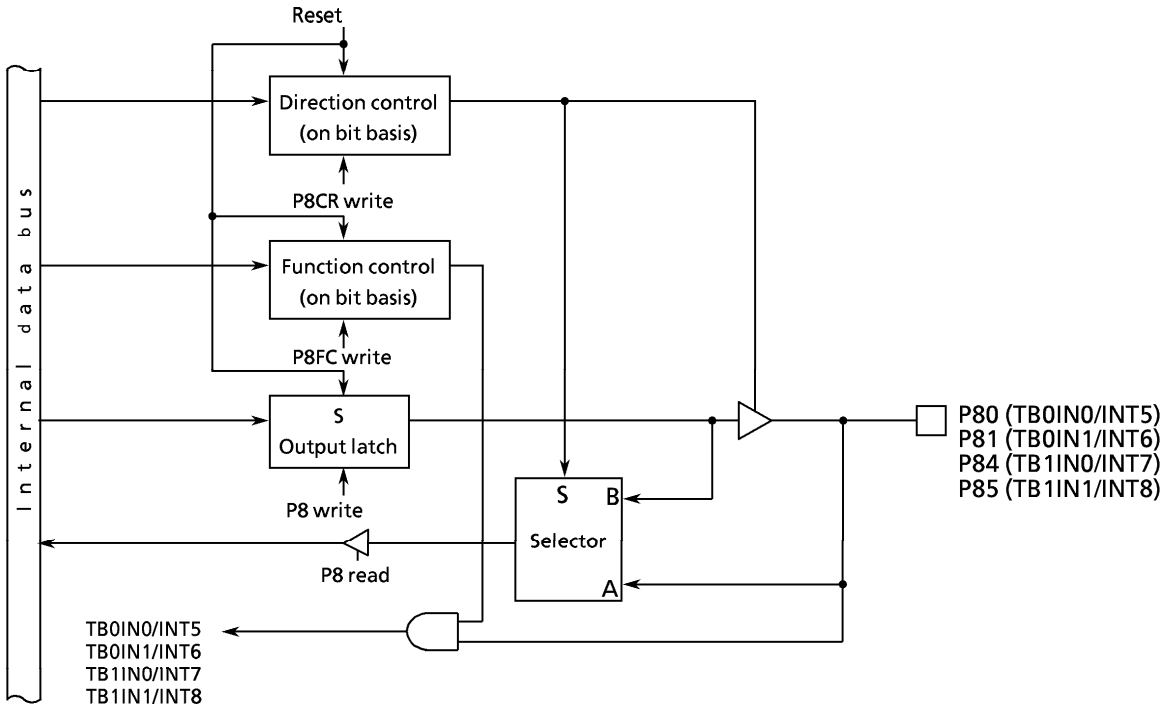


Figure 3.5.23 Port 8 (P80, P81, P84, P85)

(2) P82, P83, P86, P87

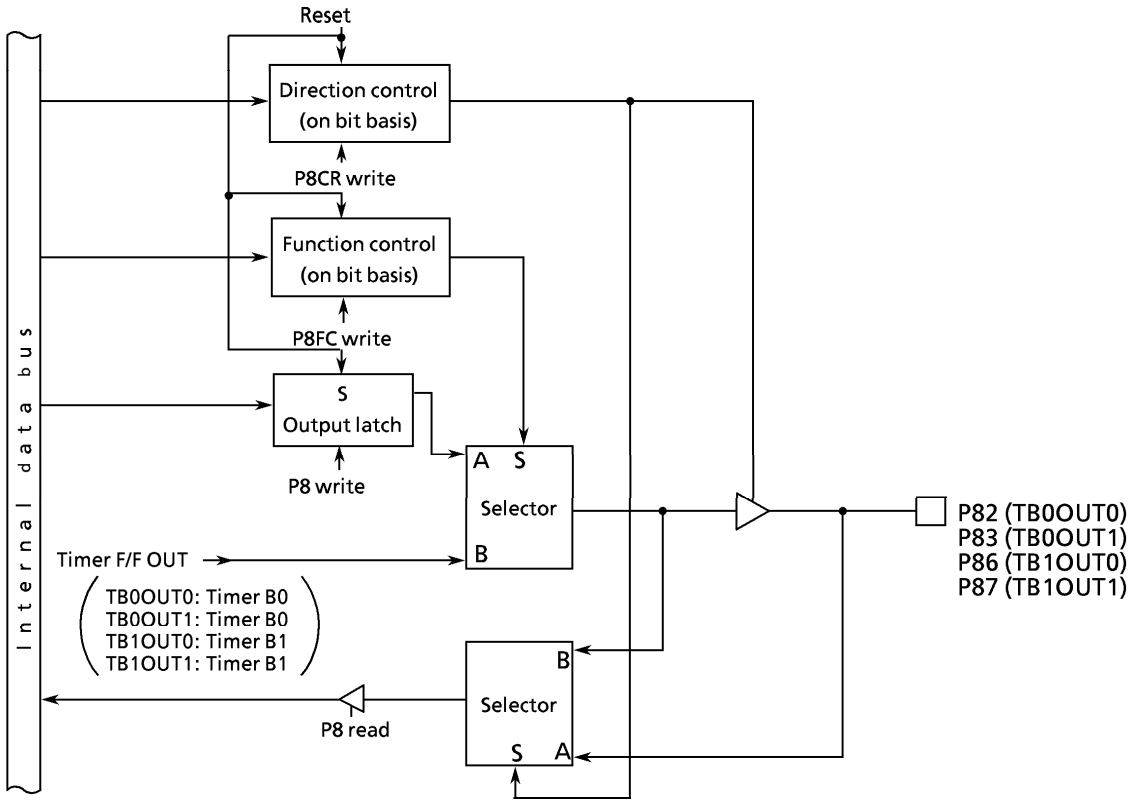


Figure 3.5.24 Port 8 (P82, P83, P86, P87)

Port 8 Register

P8 (0018H)

	7	6	5	4	3	2	1	0
Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	Data from exteranal port (Output latch register is set "1")							

Port 8 Control Register

P8CR (001AH)

	7	6	5	4	3	2	1	0
Bit symbol	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port 8 I/O setting

0	Input
1	Output

Port 8 Function Register

P8FC (001BH)

	7	6	5	4	3	2	1	0
Bit symbol	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Port 1: TB1OUT1	0: Port 1: TB1OUT0	0: Port 1: TB1IN1, INT8 input	0: Port 1: TB1IN0, INT7 input	0: Port 1: TB0OUT1	0: Port 1: TB0OUT0	0: Port 1: TB0IN1, INT6 input	0: Port 1: TB0IN0, INT5 input

Setting P82 as TB0OUT0

P8FC <P82F>	1
P8CR <P82C>	1

Setting P83 as TB0OUT1

P8FC <P83F>	1
P8CR <P83C>	1

Setting P86 as TB1OUT0

P8FC <P86F>	1
P8CR <P86C>	1

Setting P87 as TB1OUT0

P8FC <P87F>	1
P8CR <P87C>	1

Note: Read-modify-write is prohibited for registers P8CR and P8FC.

Figure 3.5.25 Registers for Port 8

3.5.10 Port 9 (P90 to P97)

- Port 90 to 95

  Port 90 to 95 is a 6-bit general-purpose I/O port. I/Os can be set on a bit basis. Resetting sets P90 to P95 an input port.
  It also sets all bits of the output latch register to 1.
  In addition to functioning as a general-purpose I/O port, P90 to P95 can also function as an I/O for serial channels 0 and 1. Writing 1 in the corresponding bit of the port 9 function register (P9FC) enables those functions.
  Resetting resets the function register P9CR, P9FC to 0 and sets all bits to input ports.

- Port 96 to 97

  Port 96 to 97 is a 2-bit general-purpose I/O port. I/Os can be set on a bit basis. The output buffer for P96 to P97 is an open drain type buffer.
  Resetting sets output latch and control registers to 1 and outputs high-impedance (High-Z).
  In addition to functioning as a general-purpose I/O port, P96 to P97 can also function as a low-frequency oscillator connecting pin (XT1, XT2) for dual clock mode. The dual clock function can be set by programming system clock control register SYSCR0, SYSCR1.

(1) Port 90, 93 (TXD0, TXD1)

Ports 90 and 93 also function as serial channel TXD output pins in addition to I/O ports.
They have a programmable open-drain function.

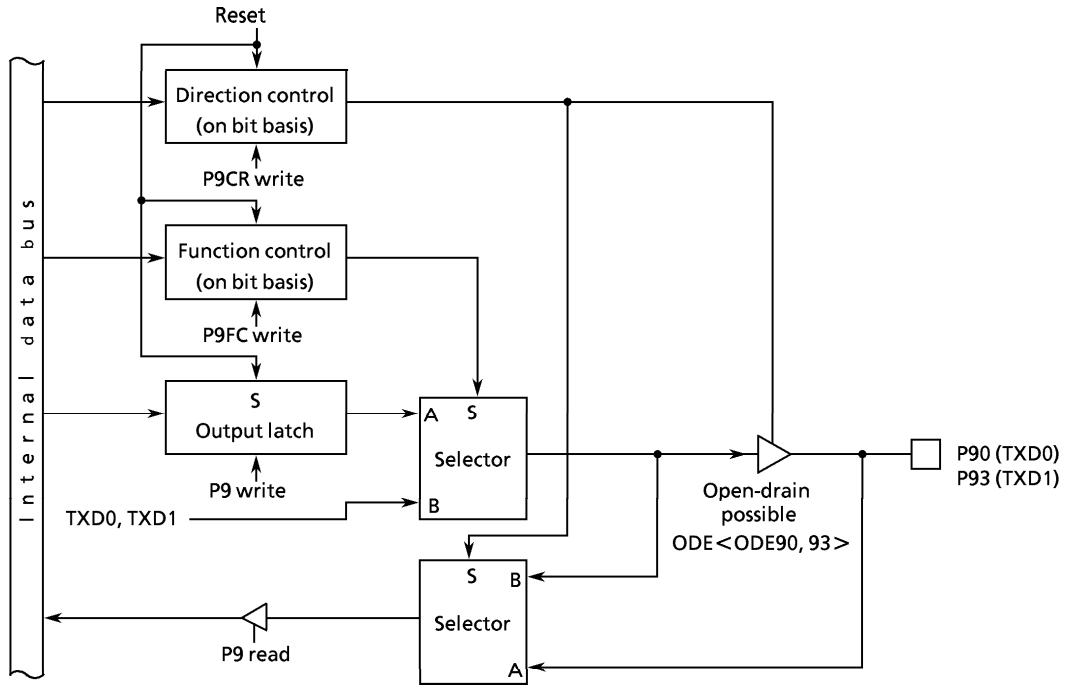


Figure 3.5.26 Ports 90 and 93

(2) Port 91,94 (RXD0, RXD1)

Port 91and 94 are I/O ports, and also used as RXD input pins for serial channels.

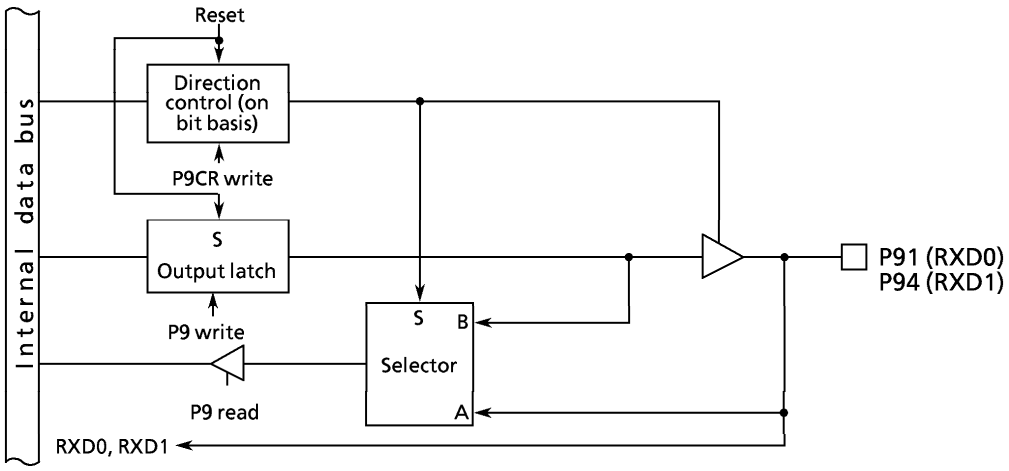


Figure 3.5.27 Ports 91 and 94

(3) Port 92, 95 ($\overline{\text{CTS0}}$/SCLK0, $\overline{\text{CTS1}}$/SCLK1)

Port 92 and P95 are I/O port, and also used as a $\overline{\text{CTS}}$ input pin and as a SCLK I/O pin for serial channel.

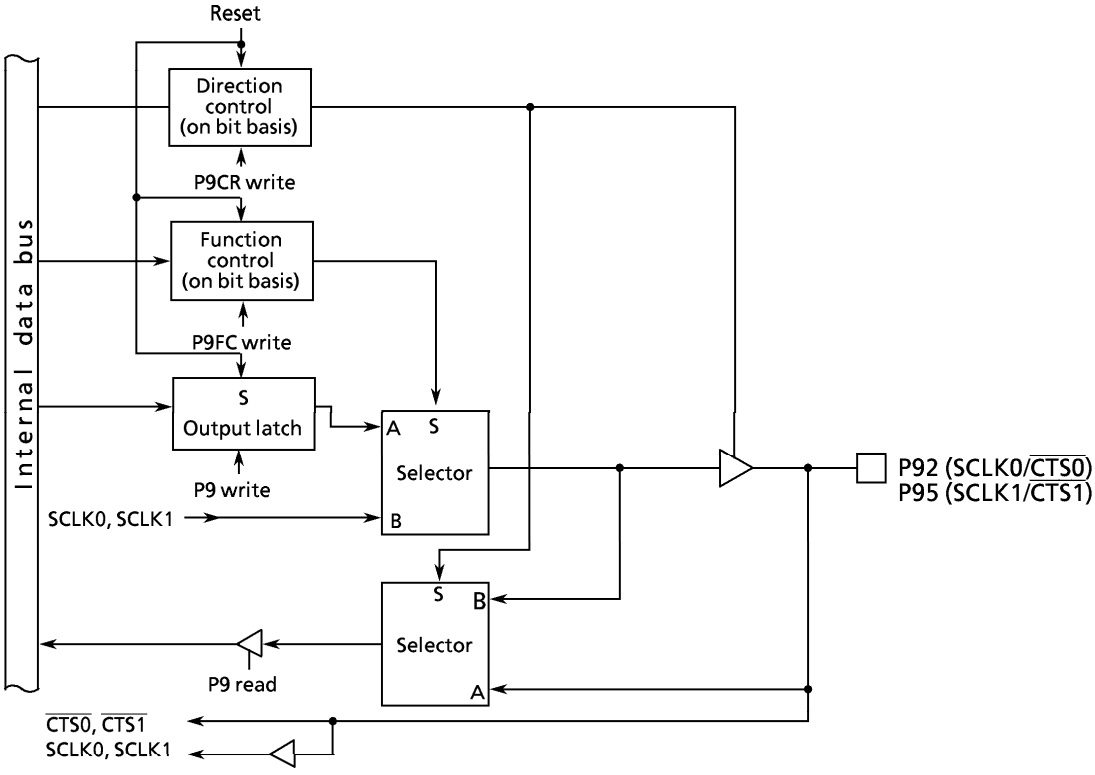


Figure 3.5.28 Ports 92, 95

(4) Port 96 (XT1), 97(XT2)

Port 96, 97 is general-purpose I/O ports. It is also used as a low-frequency oscillator connecting pin.

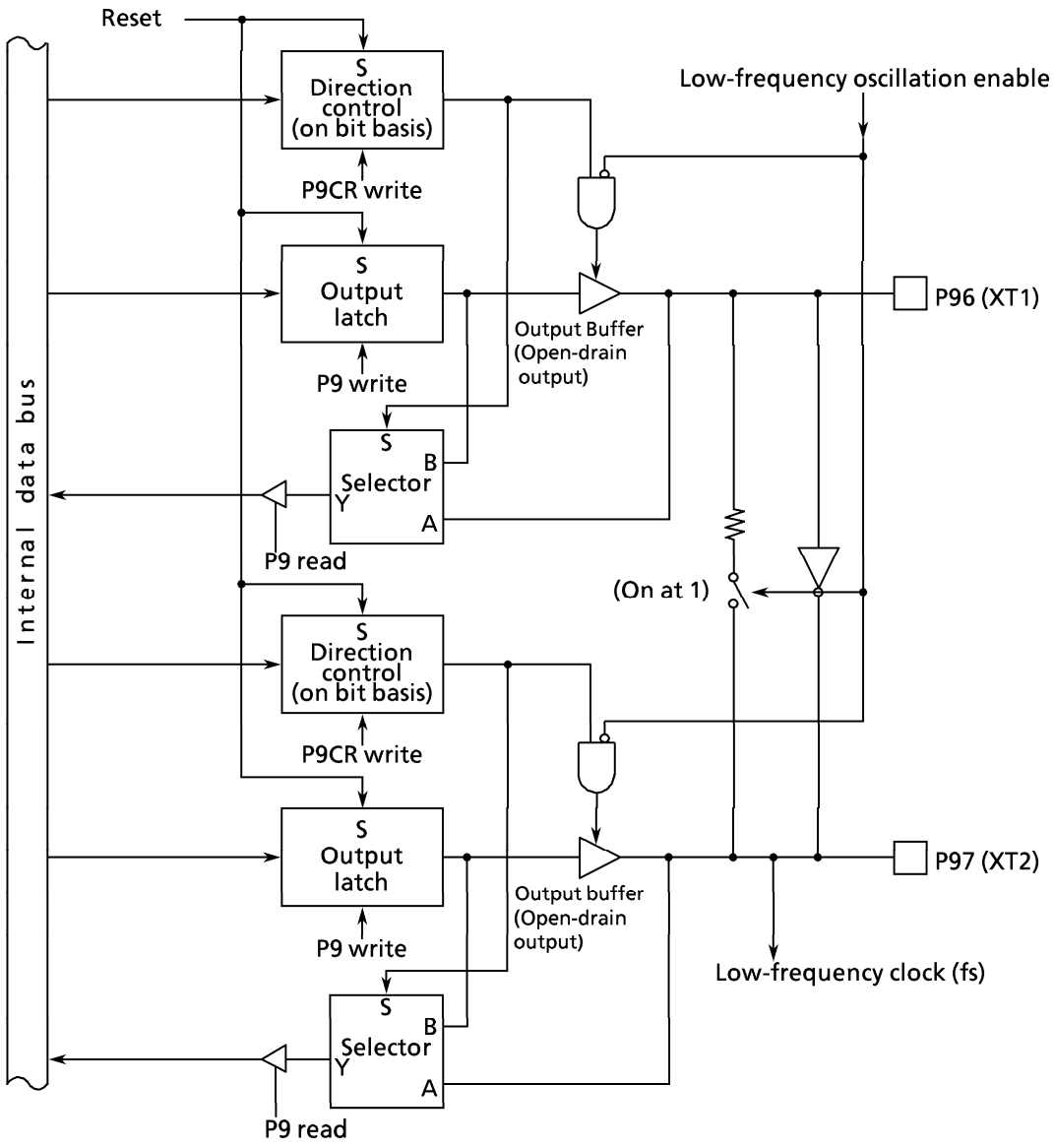


Figure 3.5.29 Port 96 to 97

Port 9 Register

P9 (0019H)

	7	6	5	4	3	2	1	0
Bit symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
After reset	1	1	Data from external port (Output latch register is set to "1")					

Port 9 Control Register

P9CR (001CH)

	7	6	5	4	3	2	1	0
Bit symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write	W							
After reset	1	1	0	0	0	0	0	0
Function	0 : Input 1 : Output							

Port 9 I/O setting

0	Input
1	Output

Note: Port 96, 97's output buffer is an open drain output type.

Port 9 Function Register

P9FC (001DH)

	7	6	5	4	3	2	1	0
Bit symbol			P95F		P93F	P92F		P90F
Read/Write			W		W	W		W
After reset			0		0	0		0
Function			0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

P90 TXD0 output setting

P9FC <P90F>	1
P9CR <P90C>	1

P92 SCLK0 output setting

P9FC <P92F>	1
P9CR <P92C>	1

P93 TXD1 output setting

P9FC <P93F>	1
P9CR <P93C>	1

P95 SCLK1 output setting

P9FC <P95F>	1
P9CR <P95C>	1

Note 1: Read-modify-write is prohibited for registers P9CR and P9FC.

Note 2: To set the TXD pin to open drain, write 1 in bit0 (for TXD0 pin) or 1 (for TXD1) pin of the ODE register. P91/RXD0, P94/RXD1 pins have no port/function switch registers. When using as input port, serial receive data is input to SIO.

Note 3: Notes on using low-frequency oscillation circuit
To connect a low-frequency resonator to port 96, 97, it is necessary to set the following procedures to reduce the consumption power supply.
(connecting to a resonator)
Set P9CR<P96C, P97C> = 11, P<P96:97> = 00.
(connection to an oscillator)
Set P9CR<P96C, P97C> = 11, P<P96:97> = 10.

Figure 3.5.30 Register for Port 9

3.5.11 Port A (PA0 to PA7)

Port A is an 8-bit general-purpose I/O port. I/Os can be set on a bit basis by control register PACR. After reset, PACR is reset to 0 and port A is set to an input port. Port A0 to A3 can also function as inputs for INT1 to INT4.

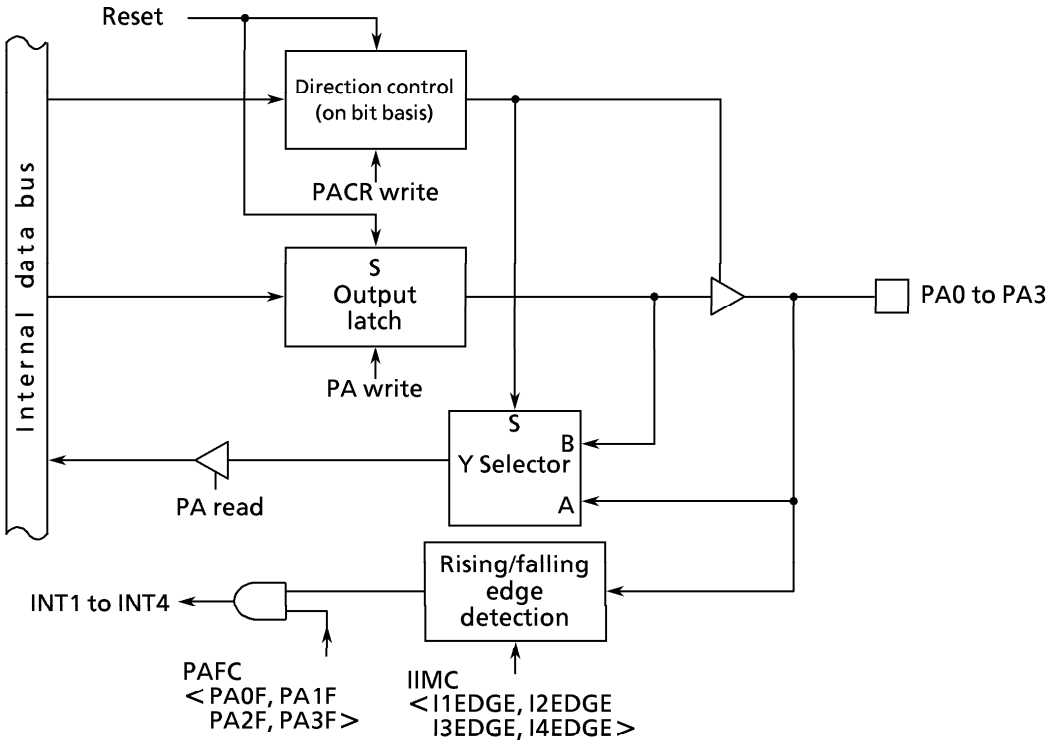


Figure 3.5.31 Port A0 to A3

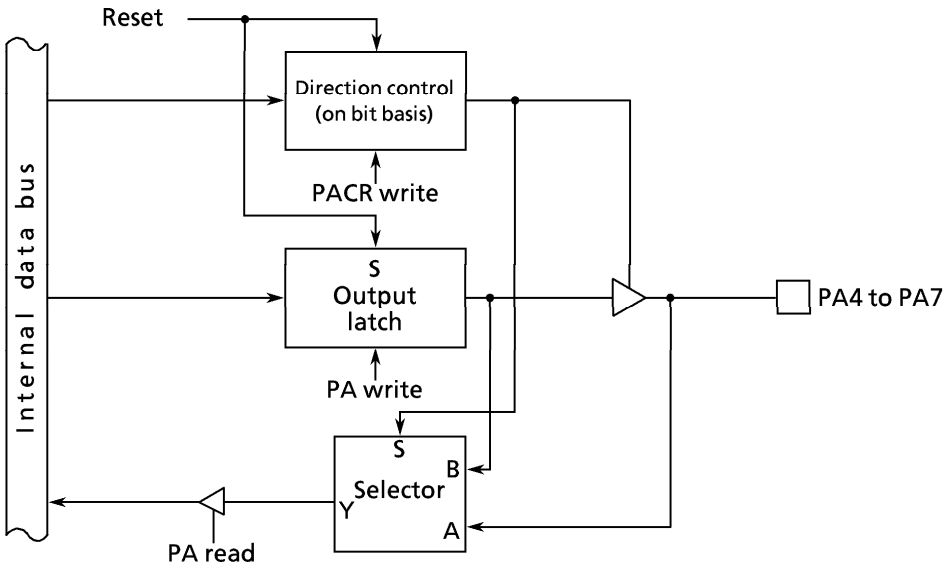


Figure 3.5.32 Port A4 to A7

Port A Register

PA (001EH)

	7	6	5	4	3	2	1	0
Bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to "1")							

Port A Control Register

PACR (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port A I/O setting

0	Input
1	Output

Port A Function Register

PAFC (0021H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	–	–	PA3F	PA2F	PA1F	PA0F
Read/Write	W	W	W	W	W			
After reset	0	0	0	0	0	0	0	0
Function	Write 0				0: Port 1: INT4 input	0: Port 1: INT3 input	0: Port 1: INT2 input	0: Port 1: INT1 input

Note: Read-modify-write is prohibited for registers PACR and PAFC.

PA0 INT1 input setting

0	PACR<PA0C>
1	PAFC<PA0F>

PA1 INT2 input setting

0	PACR<PA1C>
1	PAFC<PA1F>

PA2 INT3 input setting

0	PACR<PA2C>
1	PAFC<PA2F>

PA3 INT4 input setting

0	PACR<PA3C>
1	PAFC<PA3F>

Figure 3.5.33 Registers for Port A

3.6 Chip Select/Wait Controller

In TM91CW12, four user-specifiable address area blocks (CS0 to CS3) can be set. The data bus width and number of waits can be set independently for each address area (CS0 to CS3 and others).
The $\overline{CS0}$ to $\overline{CS3}$ pins (which also function as P40 to P43) are the output pins for the CS0 to CS3 areas. When the CPU specifies an address in one of these areas, these pins output the chip select signal for that address area (ROM/SRAM). However, to output the chip select signal, the port 4 function register P4FC must be set. TM91CW12 supports connection of external ROM and SRAM.
The CS0 to CS3 areas are set by a combination of memory start address registers MSAR0 to MSAR3 and memory address mask registers MAMR0 to MAMR3.
Use chip select/wait control registers B0CS to B3CS and BEXCS to specify the master enable, data bus width, and number of waits for each address area.
The input pin controlling these states is the bus wait request pin ($\overline{WAIT}$).

3.6.1 Specifying Address Areas

The CS0 to CS3 address areas are specified using the start address registers (MSAR0 to MSAR3) and memory address mask registers (MAMR0 to MAMR3).
At each bus cycle, a compare operation is performed to determine if the address on the bus specifies a location in the CS0 to CS3 areas. If the result of the comparison is a match, this indicates an access to the corresponding CS area. In this case, the $\overline{CS0}$ to $\overline{CS3}$ pin outputs the chip select signal and the bus cycle operates in accordance with the settings in chip select/wait control register B0CS to B3CS. (See 3.6.2 “Chip Select/Wait Control Registers”.)

(1) Memory start address registers

Figure 3.6.1 shows the memory start address registers. Memory start address registers MSAR0 to MSAR3 set the start address for the CS0 to CS3 areas. Set the upper eight bits (A23 to A16) of the start address in <S23:16>. The lower 16 bits of the start address (A15 to A0) are permanently set to 0. Accordingly, the start address can only be set in 64-Kbyte increments, starting from 000000H. Figure 3.6.2 shows the relationship between the start address and the start address register value.

Memory start address register (CS0 to CS3 areas)

	7	6	5	4	3	2	1	0
MSAR0 (00C8H) / MSAR1 (00CAH) Bit symbol	S23	S22	S21	S20	S19	S18	S17	S16
Read/Write	R/W							
MSAR2 (00CCH) / MSAR3 (00CEH) After reset	1	1	1	1	1	1	1	1
Function	Sets start address A23 to A16							

Sets start address of CS0 to CS3 area

Figure 3.6.1 Memory Start Address Register

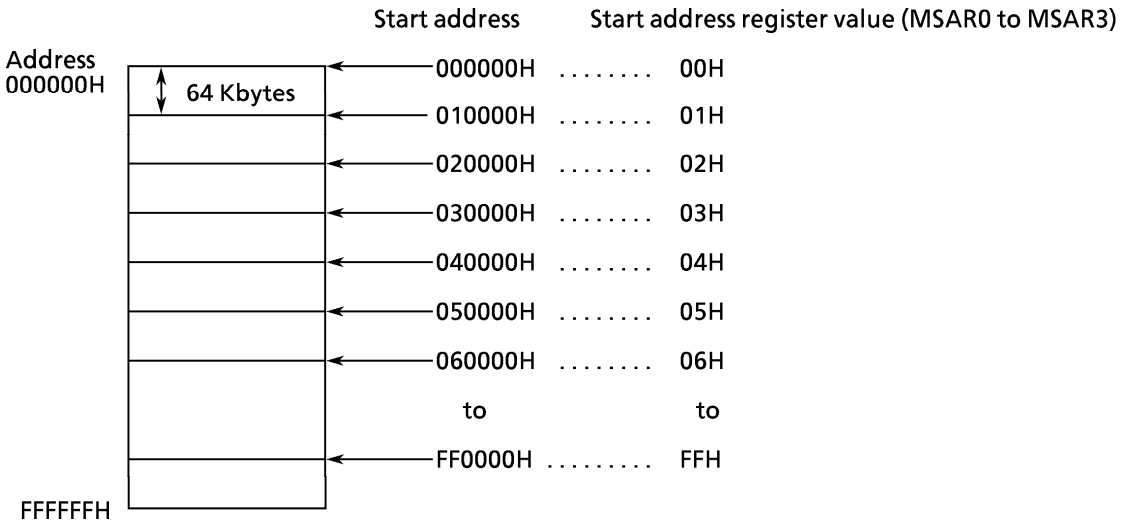


Figure 3.6.2 Relationship between Start Address and Start Address Register Value

(2) Memory address mask registers

Figure 3.6.3 shows the memory address mask registers. Memory address mask registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 areas by specifying a mask for each bit of the start address set in memory start address registers MSAR0 to MSAR3. The compare operation used to determine if an address is in the CS0 to CS3 areas is only performed for bus address bits corresponding to bits set to 0 in these registers.
Also, the address bits that can be masked by MAMR0 to MAMR3 differ between CS0 to CS3 areas. Accordingly, the size that can be set for each area is different.

Memory address mask register (CS0)

MAMR0 (00C9H)

	7	6	5	4	3	2	1	0
Bit symbol	V20	V19	V18	V17	V16	V15	V14 to V9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS0 area 0: Used for address compare							

The CS0 area can be set within the following range: 256 bytes to 2 Mbytes.

Memory address mask register (CS1)

MAMR1 (00CBH)

	7	6	5	4	3	2	1	0
Bit symbol	V21	V20	V19	V18	V17	V16	V15 to V9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS1 area 0: Used for address compare							

The CS1 area can be set within the following range: 256 bytes to 4 Mbytes.

Memory address mask register (CS2, CS3)

MAMR2 (00CDH) / MAMR3 (00CFH)

	7	6	5	4	3	2	1	0
Bit symbol	V22	V21	V20	V19	V18	V17	V16	V15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS2, CS3 area 0: Used for address compare							

The CS2 and CS3 areas can be set within the following range: 32 Kbytes to 8 Mbytes.

Figure 3.6.3 Memory Address Mask Registers

(3) How to set memory start addresses and address areas

Figure 3.6.4 shows an example of specifying a 64-Kbyte address area starting from 010000H using the CS0 area.
Set 01H in memory start address register MSAR0<S23:16> (Corresponding to the upper 8 bits of the start address). Next, calculate the difference between the start address and the anticipated end address (01FFFFH) based on the size of the CS0 area. Bits 20 to 8 of the result correspond to the mask value to be set for the CS0 area. Setting this value in memory address mask register MAMR0<V20:8> sets the area size. This example sets 07H in MAMR0 to specify a 64-Kbyte area.

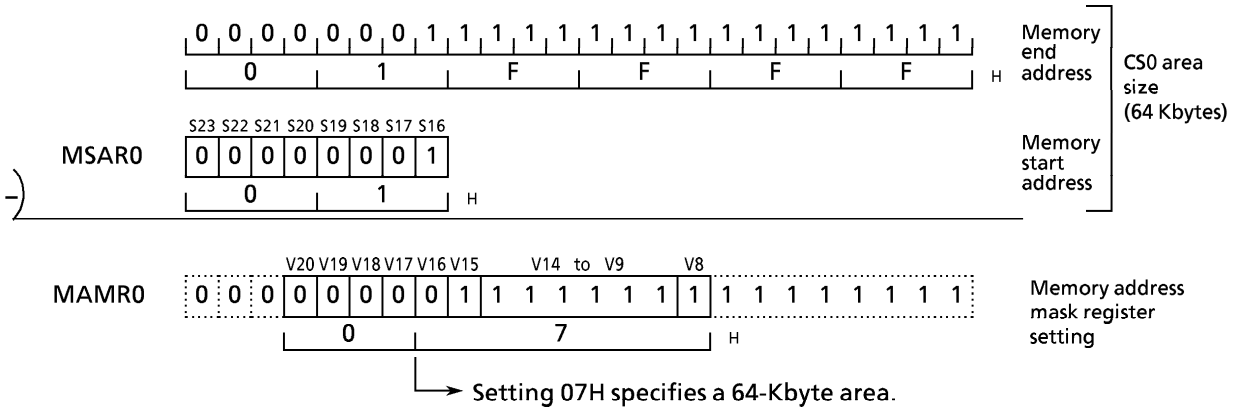


Figure 3.6.4 CS0 Area Setting Example

After a reset, MSAR0 to MSAR3 and MAMR0 to MAMR3 are set to FFH. B0CS<B0E>, B1CS <B1E>, and B3CS<B3E> are reset to 0. This disables the CS0, CS1, and CS3 areas. However, as B2CS<B2M> is reset to 0 and B2CS<B2E> to 1, CS2 is enabled from 002000H to FDFFFFH in TMP91CW12. Also, the bus width and number of waits s.ecified in BEXCS are used for accessing addresses outside the specified CS0 to CS3 areas. (See 3.6.2 "Chip Select/Wait Control Register".)

(4) Address area size specifications

Table 3.6.1 shows the relationship between CS area and area size. △ indicates areas that cannot be set by memory start address register and memory address mask register combinations. When setting an area size using a combination indicated by △, set the start address in the desired steps starting from 000000H.
If the CS2 area is set to 16 Mbytes or if two or more areas overlap, the smaller CS area number has the higher priority.

Example: When setting CS0 as a 128-Kbyte area:

① Available start addresses

000000H) 128 Kbytes
020000H) 128 Kbytes
040000H) 128 Kbytes
060000H
⋮

Any of these start addresses can be set.

② Unavailable start addresses

000000H) 64 Kbytes
010000H) 128 Kbytes
030000H) 128 Kbytes
050000H
⋮

← This exceeds the size of the steps that can be set. In this case, the following start addresses cannot set the desired area size.

Table 3.6.1 CS Area and Area Size

CS Area \ Size (bytes)	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	△	△	△	△	△		
CS1	○	○		○	△	△	△	△	△	△	
CS2			○	○	△	△	△	△	△	△	△
CS3			○	○	△	△	△	△	△	△	△

Note: △ indicates area as that cannot be set by memory start address register and address mask register combinations.

3.6.2 Chip Select/Wait Control Registers

Table 3.6.5 lists the chip select/wait control registers. The master enable/disable, chip select output waveform, data bus width, and number of wait states for each address area (CS0 to CS3 and others) are set in their respective chip select/wait control registers, B0CS to B3CS and BEXCS.

Chip Select/Wait Control Register

		7	6	5	4	3	2	1	0
B0CS (00C0H)	Bit symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: } Don't care 10: } 11: }		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 100 001: 1 wait 101 010: (1 + N) waits 110 011: 0 waits 111		Do not set
B1CS (00C1H)	Bit symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: } Don't care 10: } 11: }		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 100 001: 1 wait 101 010: (1 + N) waits 110 011: 0 waits 111		Do not set
B2CS (00C2H)	Bit symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W							
	After reset	1	0	0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable	CS2 area selection 0: 16-Mbyte area 1: CS area	Chip select output waveform selection 00: For ROM/SRAM 01: } Don't care 10: } 11: }		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 100 001: 1 wait 101 010: (1 + N) waits 110 011: 0 waits 111		Do not set
B3CS (00C3H)	Bit symbol	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: } Don't care 10: } 11: }		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 100 001: 1 wait 101 010: (1 + N) waits 110 011: 0 waits 111		Do not set
BEXCS (00C7H)	Bit symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					W			
	After reset					0	0	0	0
Read-modify-write instructions prohibited.	Function					Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 100 001: 1 wait 101 010: (1 + N) waits 110 011: 0 waits 111		Do not set

Master enable bit

0	CS area disable
1	CS area enable

CS2 area selection

0	16M-byte area
1	Address specification area

Chip select output waveform selection

00	For ROM/SRAM
01 10 11	Don't care

Number of address area waits setting
(See 3.6.2, (3) Wait control.)

Data bus width selection

0	16-bit data bus
1	8-bit data bus

Figure 3.6.5 Chip Select/Wait Control Registers

(1) Master enable bits

Bit7 (<B0E>, <B1E>, <B2E>, and <B3E>) of the chip select/wait control registers is the master bit used to enable or disable settings for the address area. Writing 1 to the bit enables the settings. Reset disables (Sets to 0) <B0E>, <B1E>, and <B3E>, and enables (sets to 1) <B2E>. This enables area CS2 only.

(2) Selection of data bus width

Bit3 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS>, and <BEXBUS>) of the chip select/wait control registers specifies the width of the data bus. Set 0 to access memory when using a 16-bit data bus. Set 1 when using an 8-bit data bus.
This method of changing the data bus width depending on the address being accessed is called dynamic bus sizing. For details of this bus operation, see Table 3.6.2.

Table 3.6.2 Dynamic Bus Sizing

Operand Data Bus Width	Operand Start Address	Memory Data Bus Width	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

xxxxx: Indicates that the input data from these bits are ignored during a read. During a write, indicates that the bus for these bits goes to high impedance; also, that the write strobe signal for the bus remains inactive.

(3) Wait control

Bits 2 to 0 (<B0W2:0>, <B1W2:0>, <B2W2:0>, <B3W2:0>, and <BEXW2:0>) of the chip select/wait control registers specify the number of waits to insert.
The following types of wait operation can be specified using combinations of these bits. Do not set combinations other than those listed in the table.

Table 3.6.3 Wait Operation Settings

<BxW2:0>	No. of Waits	Wait Operation
000	2 waits	Inserts a wait of two states, irrespective of the $\overline{\text{WAIT}}$ pin state.
001	1 wait	Inserts a wait of one state, irrespective of the $\overline{\text{WAIT}}$ pin state.
010	(1 + N) waits	Samples the state of the $\overline{\text{WAIT}}$ pin after inserting a wait of one state. If the $\overline{\text{WAIT}}$ pin is low, the waits continue and the bus cycle is extended until the pin goes high.
011	0 waits	Ends the bus cycle without a wait, regardless of the $\overline{\text{WAIT}}$ pin state.

Reset sets these bits to 000 (2 waits).

(4) Bus width and wait control outside CS0 to CS3 areas

The chip select/wait control register BEXCS controls the bus width and number of waits when locations outside the four user-specified address area blocks (CS0 to CS3) are accessed. The BEXCS register settings are always enabled for areas other than CS0 to CS3.

(5) 16-Mbyte area/address setting area selection

Setting the chip select/wait control register B2CS<B2M> to 0 selects a 16-Mbyte address area (002000H to FDFFFFH) for CS2. Setting B2CS<B2M> to 1 selects the address area specified by start address register MSAR2 and address mask register MAMR2 for CS2, and likewise for CS0, CS1, and CS3. Reset clears this bit to 0 and selects a 16-Mbyte address area.

(6) Chip select/wait control setting procedure

When using the chip select/wait control function, set the registers as follows:

① Set memory start address registers MSAR0 to MSAR3.
Set the CS0 to CS3 start addresses.
② Set memory address mask registers MAMR0 to MAMR3.
Set the size of CS0 to CS3.
③ Set control registers B0CS to B3CS.
Set the chip select output waveform, data bus width, number of waits, and master enable/disable for CS0 to CS3.

The $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins also function as pins P40 to P43. To output the chip select signal from these pins, set the corresponding bits of port 6 function register P4FC to 1.
In the case of addresses set for one of the CS0 to CS3 areas but which specify an internal I/O, RAM, or ROM area, the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins do not output a chip select signal and the CPU accesses the internal area.

(Setting example)
This example sets the CS0 area as 010000H to 01FFFFH (64-Kbyte area) with a 16-bit bus and zero waits:

MSAR0 = 01H ..... Start address: 010000H
MAMR0 = 07H ..... Address area: 64 Kbytes
B0CS = 83H ..... ROM/SRAM, 16-bit data bus, zero waits, CS0 area settings enabled

3.6.3 How to Connect External Memory

Figure 3.6.6 shows an example of connecting external memory to TMP91CW12.
In the example, ROM is connected using a 16-bit bus. RAM and I/O are connected using an 8-bit bus.

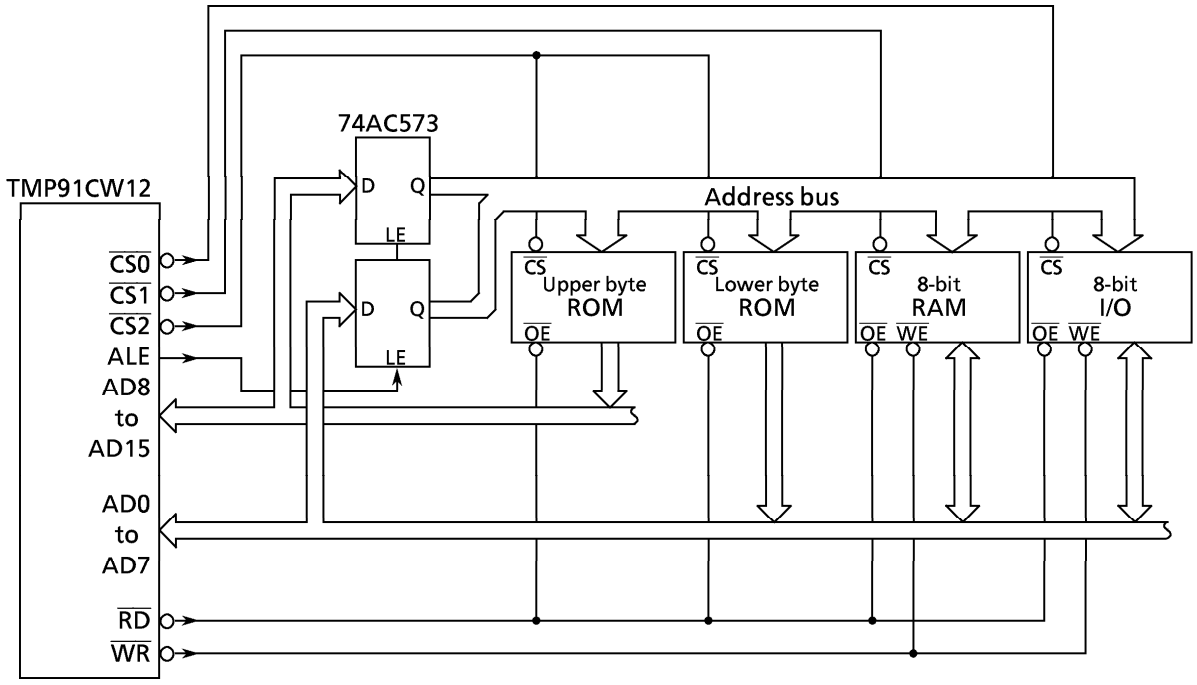


Figure 3.6.6 External Memory Connection Example
(ROM = 16-bit bus, RAM and I/O = 8-bit bus)

After resetting TMP91CW12, port 4 control register P4CR and function register P4FC are cleared to 0, the CS signal output is disabled. When outputting the CS signal, set the necessary bit to 1.

3.7 8-Bit Timers (TMRA)

The 8-channels of 8-bit timer are built-in.
It is consisted of 4-modules which is called TMRA01, TMRA23, TMRA45 and TMRA67. Each module has 2-channels and the following four operating modes.

- 8-bit interval timer mode (4 timers)
- 16-bit interval timer mode (2 timers)
- 8-bit programmable square wave pulse generation (PPG: Variable duty with variable cycle) output mode (1 timer)
- 8-bit pulse width modulation (PWM: Variable duty with constant cycle) output mode (1 timer)

Figure 3.7.1 to 3.7.4 shows the block diagram of TMRA01, TMRA23, TMRA45 and TMRA67.
Each channel consists of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Besides, timer flip-flops and prescaler are provided for pair of 2 channels.
The operation modes and timer flip-flops are controlled by five control registers (SFRs).
Each module (TMRA01, TMRA23, TMRA45 and TMRA67) can be used independently. All modules operate in the same manner, and thus only the operation of TMRA01 will be explained below.
This chapter is explained following contents.

3.7.1 Block Diagram
3.7.2 Operation for Each Circuit
3.7.3 SFRs
3.7.4 Operation for Each Mode
(1) 8-bit timer mode
(2) 16-bit timer mode
(3) 8-bit PPG (Programmable pulse generation) output mode
(4) 8-bit PWM output mode
(5) Setting for each mode

Table 3.7.1 Different Point of Each Module

Module		TMRA01	TMRA23	TMRA45	TMRA67
External pin	Input pin for external clock	TA0IN (Shared with P70)	No	TA4IN (Shared with P73)	No
	Output pin for timer flip-flops	TA1OUT (Shared with P71)	TA3OUT (Shared with P72)	TA5OUT (Shared with P74)	TA7OUT (Shared with P75)
SFR (Address)	Timer run register	TA01RUN (0100H)	TA23RUN (0108H)	TA45RUN (0110H)	TA67RUN (0118H)
	Timer register	TA0REG (0102H) TA1REG (0103H)	TA2REG (010AH) TA3REG (010BH)	TA4REG (0112H) TA5REG (0113H)	TA6REG (011AH) TA7REG (011BH)
	Timer mode register	TA01MOD (0104H)	TA23MOD (010CH)	TA45MOD (0114H)	TA67MOD (011CH)
	Timer flip-flops control register	TA1FFCR (0105H)	TA3FFCR (010DH)	TA5FFCR (0115H)	TA7FFCR (011DH)

3.7.1 Block Diagram

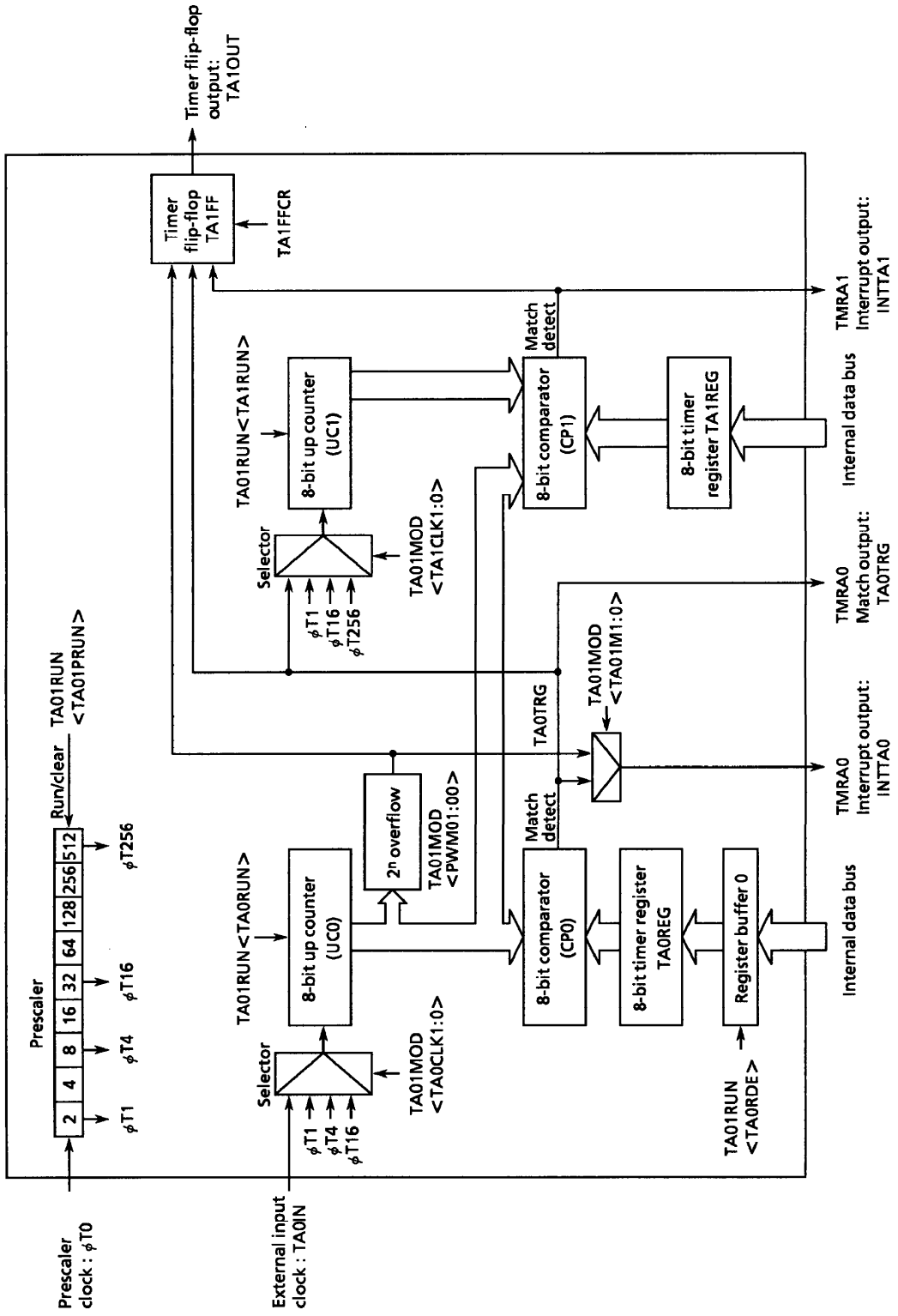


Figure 3.7.1 TMRA01 Block Diagram

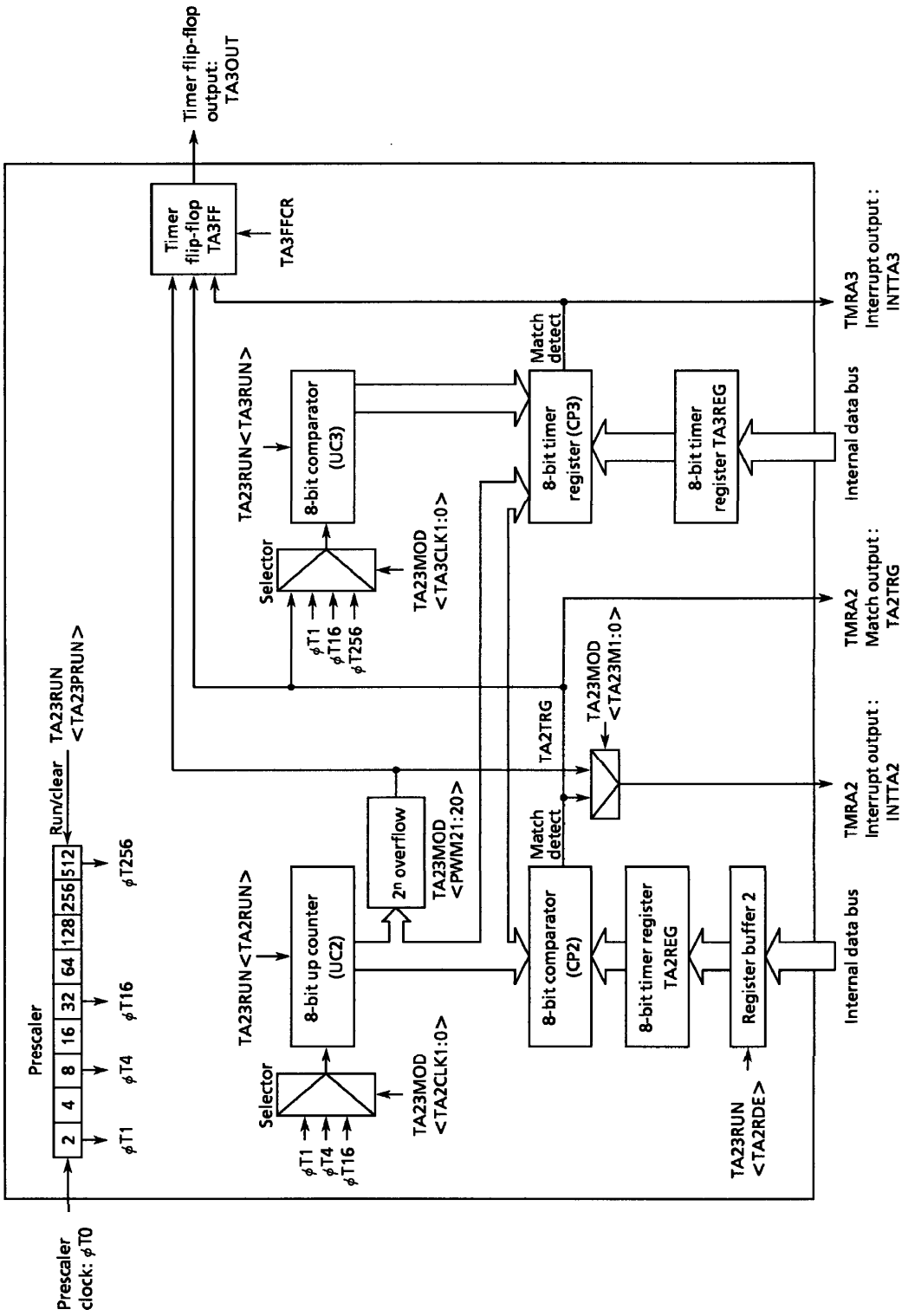


Figure 3.7.2 TMRA23 Block Diagram

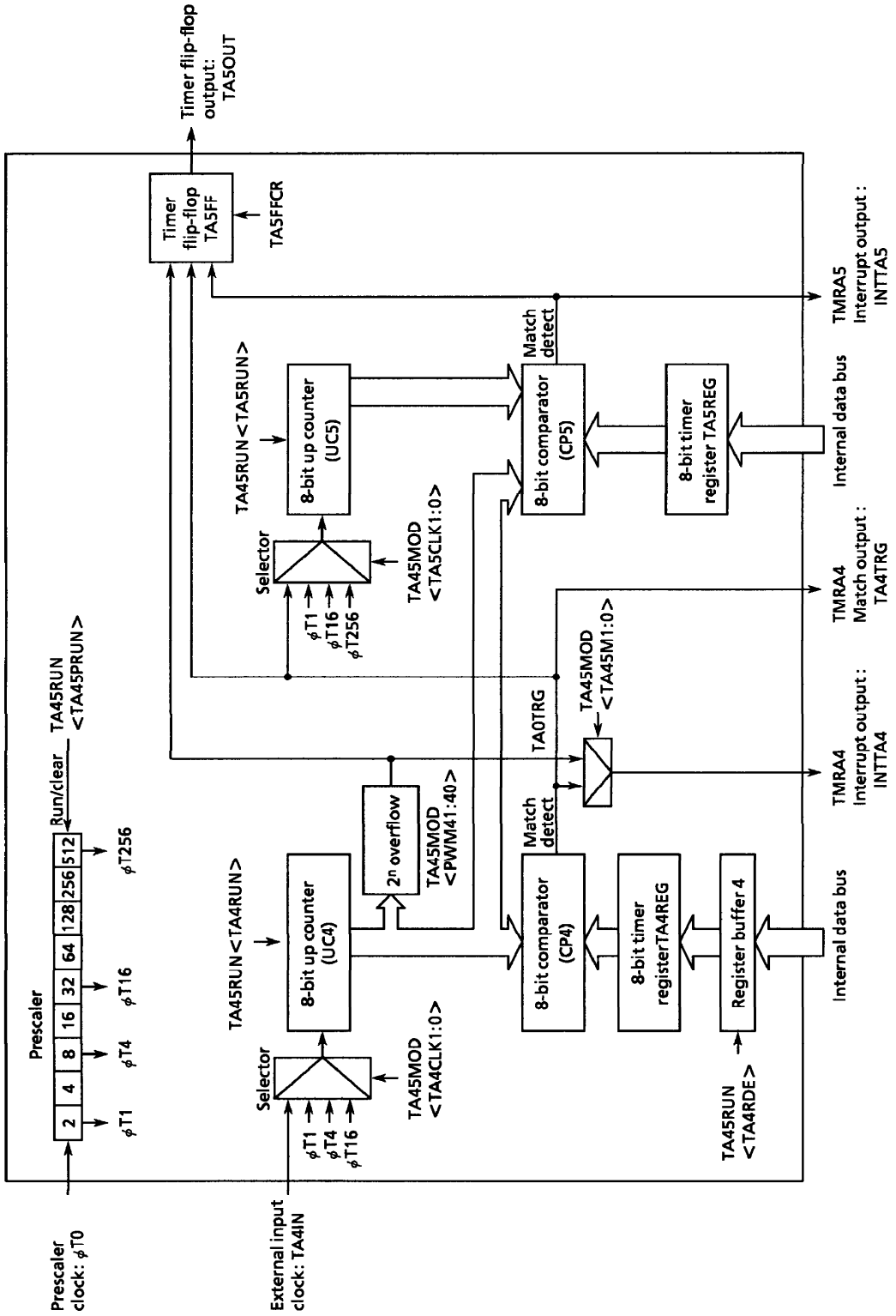


Figure 3.7.3 TMRA45 Block Diagram

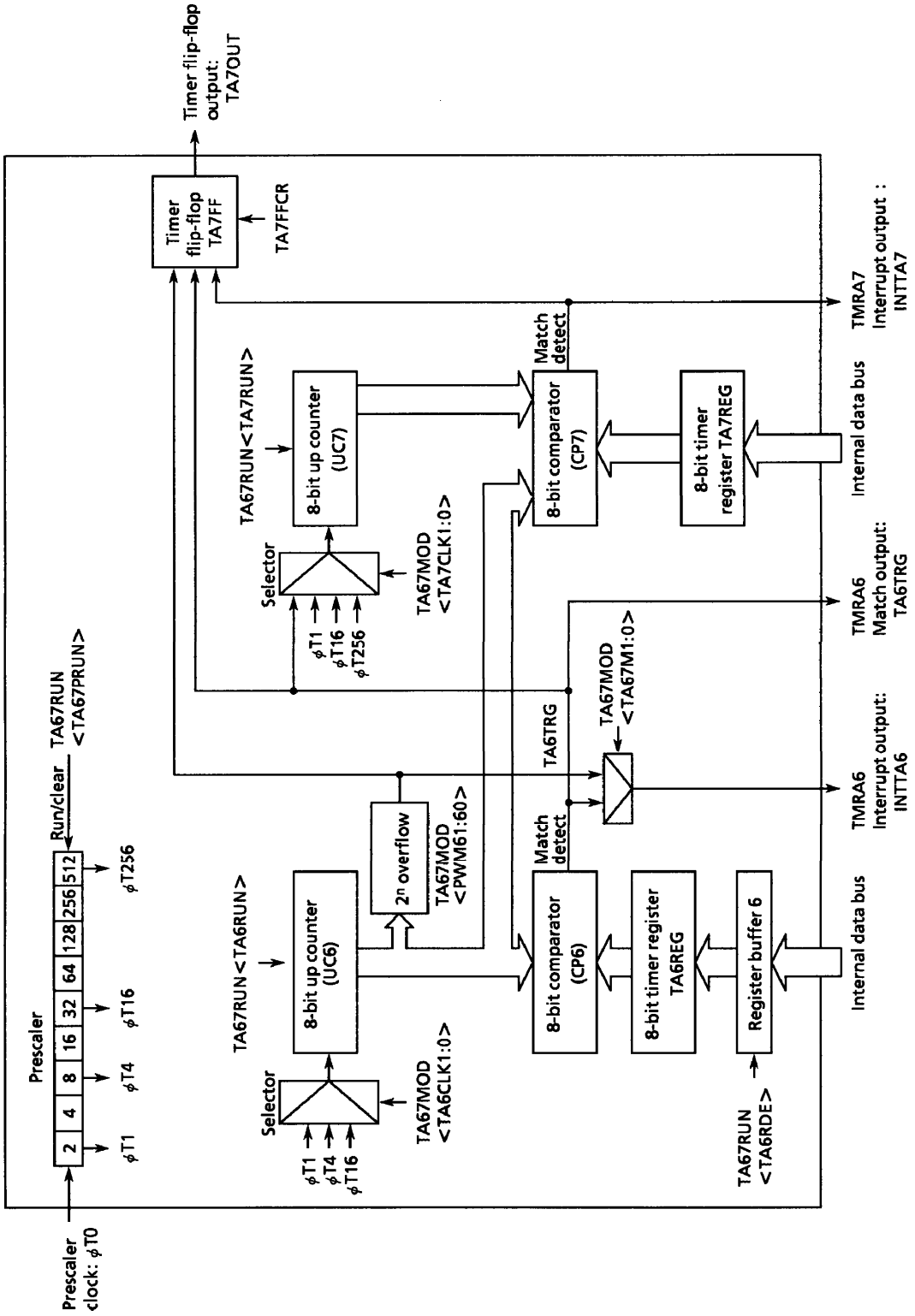


Figure 3.7.4 TMRA67 Block Diagram

3.7.2 Operation for Each Circuit

(1) Prescaler

There are 9-bit prescaler to generate input clock for TMRA01.
The clock ϕT0 selected between f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.
This prescaler can be run or stopped by the timer control register TA01RUN<TA0PRUN>. Counting starts when <TA0PRUN> is set to 1, while the prescaler is cleared to zero and stops operation when <TA0PRUN> is set to 0. Table 3.7.2 shows the prescaler output clock resolution.

Table 3.7.2 Prescaler Output Clock Resolution

at fc = 16 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Output Clock Resolution			
			ϕT1	ϕT4	ϕT16	ϕT256
1 (fs)	00 (f_{FPH})	XXX	fs/2^3 (244 μs)	fs/2^5 (977 μs)	fs/2^7 (3.9 ms)	fs/2^{11} (62.5 ms)
0 (fc)		000 (fc)	fc/2^3 (0.5 μs)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^{11} (128 μs)
		001 (fc/2)	fc/2^4 (1.0 μs)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{12} (256 μs)
		010 (fc/4)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{13} (512 μs)
		011 (fc/8)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{10} (64 μs)	fc/2^{14} (1024 μs)
		100 (fc/16)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)	fc/2^{15} (2048 μs)
	10 (fc/16 clock)	XXX	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)	fc/2^{15} (2048 μs)

XXX: Don't care

(2) Up counter (UC0, UC1)

This is an 8-bit binary counter which counts up by the input clock pulse specified by TA01MOD.
The input clock of UC0 is selected from the external clock from TA0IN pin and the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of TA01MOD<TA0CLK1:0> registers.
The input clock of UC1 differs depending on the operation mode. When set to 16-bit timer mode, the overflow outputs of UC0 are used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (match detection signal) of TMRA0.
The counting and stop and clear of up counter can be controlled for each interval timer by the timer operation control register TA01RUN<TA0RUN>, <TA1RUN>. When reset, all up counters will be cleared to stop the timers.

(3) Timer register (TA0REG, TA1REG)

This is an 8-bit register for setting an interval time. When the set value of timer registers TA0REG, TA1REG matches the value of up counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up counter overflows.
The TA0REG are double buffer structure, each of which makes a pair with register buffer.
The TA01RUN<TA0RDE> register control whether the double buffer structure in the TA0REG should be enabled or disabled. They are disabled when <TA0RDE> = 0 and enabled when they are set to 1.
In the condition of double buffer enable state, the data is transfered from the register buffer to the timer register when the 2^n overflow occurs in PWM mode, or at the PPG cycle in PPG mode. Therefore, during timer mode, the double buffer can not be used.
When reset, it will be initialized to <TA0RDE> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <TA0RDE> to 1, and write the following data in the register buffer. Figure 3.7.5 shows the configuration of TA0REG.

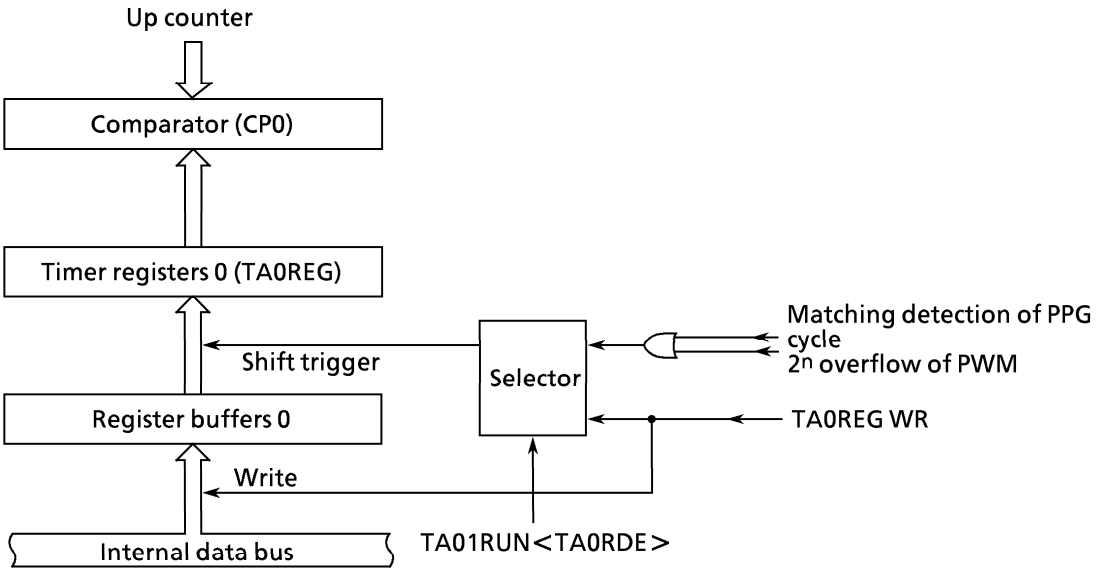


Figure 3.7.5 Configuration of TA0REG

Note : Timer register and the register buffer are allocated to the same memory address. When <TA0RDE> = 0, the same value is written in the register buffer as well as the timer register, while when <TA0RDE> = 1 only the register buffer is written.

The address of each timer register is as follows.

TA0REG: 000102H TA1REG: 000103H
TA2REG: 00010AH TA3REG: 00010BH
TA4REG: 000112H TA5REG: 000113H
TA6REG: 00011AH TA7REG: 00011BH

All register are write only and cannot be read.

(4) Comparator (CP0)

A comparator compares the value in the up counter with the values to which the timer register is set. When they match, the up counter is cleared to zero and an interrupt signal (INTTA0, INTTA1) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

(5) Timer flip-flop (TA1FF)

The timer flip-flop (TA1FF) is a flip-flop inverted by the match detect signal (8-bit comparator output) of each interval timer.
Inverting is disabled or enabled by the timer flip-flop control register TA1FFCR<TA1FFIE>.
Resetting clears to 0 the value of TA1FF. Writing 01 or 10 to TA1FFCR<TA1FFC1:0>sets 0 or 1 to TA1FF. Additionally, writing 00 to this bit inverts the value of TA1FF. (Software inversion)
The signal of TA1FF is output through the TA1OUT pin (also used as P71). When using as the timer output, the timer flip-flop should be set by port 7 function register P7FC beforehand.

3.7.3 SFRs

TMRA01 RUN Register

TA01RUN (0100H)

	7	6	5	4	3	2	1	0
Bit symbol	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Control TA0REG double buffer

0	Disable
1	Enable

Timer run/stop control

0	Stop and Clear
1	Run (Count up)

I2TA01: Operation during IDLE2 mode
TA01PRUN: RUN for prescaler
TA1RUN: RUN for timer 1
TA0RUN: RUN for timer 0

Note: Bit4 to 6 of TA01RUN are read as undefined value.

TMRA23 RUN Register

TA23RUN (0108H)

	7	6	5	4	3	2	1	0
Bit symbol	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Control TA2REG double buffer

0	Disable
1	Enable

Timer run/stop control

0	Stop and clear
1	Run (Count up)

I2TA23: Operation during IDLE2 mode
TA23PRUN: RUN for prescaler
TA3RUN: RUN for timer 3
TA2RUN: RUN for timer 2

Note: Bit4 to 6 of TA23RUN are read as undefined value.

Figure 3.7.6 Register for TMRA

TMRA45 RUN Register

TA45RUN (0110H)

	7	6	5	4	3	2	1	0
Bit Symbol	TA4RDE				I2TA45	TA45PRUN	TA5RUN	TA4RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Control TA4REG double buffer

0	Disable
1	Enable

Timer run/stop control

0	Stop and Clear
1	Run (Count up)

I2TA45: Operation during IDLE2 mode
TA45PRUN: RUN for prescaler
TA5RUN: RUN for timer 5
TA4RUN: RUN for timer 4

Note: Bit4 to 6 of TA45RUN are read as undefined value.

TMRA67 RUN Register

TA67RUN (0118H)

	7	6	5	4	3	2	1	0
Bit Symbol	TA6RDE				I2TA67	TA67PRUN	TA7RUN	TA6RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Control TA6REG double buffer

0	Disable
1	Enable

Timer run/stop control

0	Stop and clear
1	Run (Count up)

I2TA67: Operation during IDLE2 mode
TA67PRUN: RUN for prescaler
TA7RUN: RUN for timer 7
TA6RUN: RUN for timer 6

Note: Bit4 to 6 of TA67RUN are read as undefined value.

Figure 3.7.7 Register for TMRA

TMRA01 Mode Register

TA01MOD (0104H)

	7	6	5	4	3	2	1	0
Bit Symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		Source clock of TMRA1 00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of TMRA0 00: TA0IN pin 01: ϕT1 10: ϕT4 11: ϕT16	

Source clock of TMRA0

00	TA0IN (External input)
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Source clock of TMRA0

	TA01MOD<TA01M1:0> ≠ 01	TA01MOD<TA01M1:0> = 01
00	Comparator output of TMRA0	Overflow output of TMRA0
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

Select PWM cycle

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Operation mode of TMRA01

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA0) + 8-bit timer (TMRA1)

Figure 3.7.8 Register for TMRA

TMRA23 Mode Register

TA23MOD (010CH)

	7	6	5	4	3	2	1	0
Bit symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		Source clock of TMRA3 00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of TMRA2 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

Source clock of TMRA2

00	Don't set
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Source clock of TMRA3

	TA23MOD<TA23M1:0> ≠ 01	TA23MOD<TA23M1:0> = 01
00	Comparator output of TMRA2	Overflow output of TMRA2
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

Select PWM cycle

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Operation mode of TMRA23

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA2) + 8-bit timer (TMRA3)

Figure 3.7.9 Register for TMRA

TMRA45 Mode Register

TA45MOD (0114H)

	7	6	5	4	3	2	1	0
Bit symbol	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		Source clock of TMRA5 00: TA4TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of TMRA4 00: TA4IN pin 01: ϕT1 10: ϕT4 11: ϕT16	

Source clock of TMRA4

00	TA4IN (External input)
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Source clock of TMRA5

	TA45MOD<TA45M1:0> ≠ 01	TA45MOD<TA45M1:0> = 01
00	Comparator output of TMRA4	Overflow output of TMRA4
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

Select PWM cycle

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Operation mode of TMRA45

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA4) + 8-bit timer (TMRA5)

Figure 3.7.10 Register for TMRA

TMRA67 Mode Register

TA67MOD (011CH)

	7	6	5	4	3	2	1	0
Bit symbol	TA67M1	TA67M0	PWM61	PWM60	TA7CLK1	TA7CLK0	T6CLK1	TA6CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		Source clock of TMRA7 00: TA6TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of TMRA6 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

Source clock of TMRA6

00	Don't set
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Source clock of TMRA7

	TA67MOD<TA67M1:0> ≠ 01	TA67MOD<TA67M1:0> = 01
00	Comparator output of TMRA6	Overflow output of TMRA6
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

Select PWM cycle

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Operation mode of TMRA67

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA6) + 8-bit timer (TMRA7)

Figure 3.7.11 Register for TMRA

TMRA1 Flip-flop Control Register

TA1FFCR (0105H)

	7	6	5	4	3	2	1	0
Bit symbol					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don't care		TA1FF control for invert 0: Disable 1: Enable	TA1FF inversion select 0: TMRA0 1: TMRA1

Read-modify-write instructions prohibited.

Select inverse signal of timer flip-flop 1 (TA1FF)
(Don't care except 8-bit timer mode)

0	Inversion by TMRA0
1	Inversion by TMRA1

Inversion of TA1FF

0	Disable invert
1	Enable invert

Control of TA1FF

00	Invert the value of TA1FF
01	Set TA1FF to 1
10	Clear TA1FF to 0
11	Don't care

Note: Bit4 to 7 of TA1FFCR are read as undefined value.

Figure 3.7.12 Register for TMRA

TMRA3 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TA3FFCR (010DH)	Bit symbol					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
	Read/Write					R/W		R/W	
	After reset					1	1	0	0
Read-modify-write instructions prohibited.	Function					00 : Invert TA3FF 01 : Set TA3FF 10 : Clear TA3FF 11 : Don't care		TA3FF control for invert 0: Disable 1: Enable	TA3FF inversion select 0: TMRA2 1: TMRA3

Select inverse signal of timer flip-flop 3 (TA3FF)
(Don't care except 8-bit timer mode)

0	Inversion by TMRA2
1	Inversion by TMRA3

Inversion of TA3FF

0	Disable invert
1	Enable invert

Control of TA3FF

00	Invert the value of TA3FF
01	Set TA3FF to 1
10	Clear TA3FF to 0
11	Don't care

Note: Bit4 to 7 of TA3FFCR are read as undefined value.

Figure 3.7.13 Register for TMRA

TMRA5 Flip-flop Control Register

TA5FFCR (0115H)

	7	6	5	4	3	2	1	0
Bit symbol					TA5FFC1	TA5FFC0	TA5FFIE	TA5FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00 : Invert TA5FF 01 : Set TA5FF 10 : Clear TA5FF 11 : Don't care		TA5FF control for invert 0: Disable 1: Enable	TA5FF inversion select 0: TMRA4 1: TMRA5

Read-modify-write instructions prohibited.

Select inverse signal of timer flip-flop 5 (TA5FF) (Don't care except 8-bit timer mode)

0	Inversion by TMRA4
1	Inversion by TMRA5

Inversion of TA5FF

0	Disable invert
1	Enable invert

Control of TA5FF

00	Invert the value of TA5FF
01	Set TA5FF to 1
10	Clear TA5FF to 0
11	Don't care

Note: Bit4 to 7 of TA5FFCR are read as undefined value.

Figure 3.7.14 Register for TMRA

TMRA7 Flip-flop Control Register

TA7FFCR (011DH)

	7	6	5	4	3	2	1	0
Bit symbol					TA7FFC1	TA7FFC0	TA7FFIE	TA7FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00 : Invert TA7FF 01 : Set TA7FF 10 : Clear TA7FF 11 : Don't care		TA7FF control for invert 0: Disable 1: Enable	TA7FF inversion select 0: TMRA6 1: TMRA7

Read-modify-write instructions prohibited.

Select inverse signal of timer flip-flop 7 (TA7FF) (Don't care except 8-bit timer mode)

0	Inversion by TMRA6
1	Inversion by TMRA7

Inversion of TA7FF

0	Disable invert
1	Enable invert

Control of TA7FF

00	Invert the value of TA7FF
01	Set TA7FF to 1
10	Clear TA7FF to 0
11	Don't care

Note: Bit4 to 7 of TA7FFCR are read as undefined value.

Figure 3.7.15 Register for TMRA

3.7.4 Operation for Each Mode

(1) 8-bit timer mode

Both TMRA0 and TMRA1 can be used independently as 8-bit interval timer.

① Generating interrupts in a fixed cycle (in case of TMRA1)

To generate interrupt at constant intervals using TMRA1 (INTTA1), first stop TMRA1 then set the operation mode, input clock, and a cycle to TA01MOD and TA1REG register, respectively. Then, enable interrupt INTTA1 and start the counting of TMRA1.

Example: To generate INTTA1 interrupt every 20 μs at fc = 16 MHz, set each register in the following manner.

※ Clock condition
System clock: High frequency (fc)
Prescaler clock: f_{FPH}

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← -	X	X	X	-	-	0	-	Stop TMRA1, and clear it to 0.
TA01MOD	← 0	0	X	X	0	1	X	X	Set the 8-bit timer mode, and select ϕT1 (0.5 μs at fc = 16 MHz) as the input clock.
TA1REG	← 0	0	1	0	1	0	0	0	Set the TA1REG 20 μs ÷ ϕT1 = 40 = 28H
INTETA01	← X	1	0	1	-	-	-	-	Enable INTTA1, and set it to "Level 5".
TA01RUN	← -	X	X	X	-	1	1	-	Start TMRA1 counting.

X: Don't care, –: No change

Use the table 3.7.2 for selecting the input clock.

Note: The input clock of TMRA0 and TMRA1 are different from as follows.
TMRA0 : TA0IN input, ϕT1, ϕT4, ϕT16
TMRA1 : Match output of TMRA0, ϕT1, ϕT16, ϕT256

② Generating a 50 % duty square wave pulse

The timer flip-flop (TA1FF) is inverted at constant intervals, and its status is output to timer output pin (TA1OUT).

(Setting example)

To output a 3.0 μs square wave pulse from TA1OUT pin at fc = 16 MHz, set each register in the following procedures. Either TMRA0 or TMRA1 may be used, but this example uses TMRA1.

※ Clock condition

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

	7 6 5 4 3 2 1 0	
TA01RUN	← – X X X – – 0 –	Stop TMRA1, and clear it to 0.
TA01MOD	← 0 0 X X 0 1 X X	Set the 8-bit timer mode, and select φT1 (0.5 μs at fc = 16 MHz) as the input clock.
TA1REG	← 0 0 0 0 0 0 1 1	Set the timer register at 3.0 μs ÷ φT1 ÷ 2 = 3.
TA1FFCR	← X X X X 1 0 1 1	Set TA1FF to 0, and set to invert by the match detect signal from TMRA1.
P7CR	← X X – – – – 1 –	Select P71 as TA1OUT pin.
P7FC	← X X – – X – 1 X	
TA01RUN	← – X X X – 1 1 –	Start TMRA1 counting.

X: Don't care, –: No change

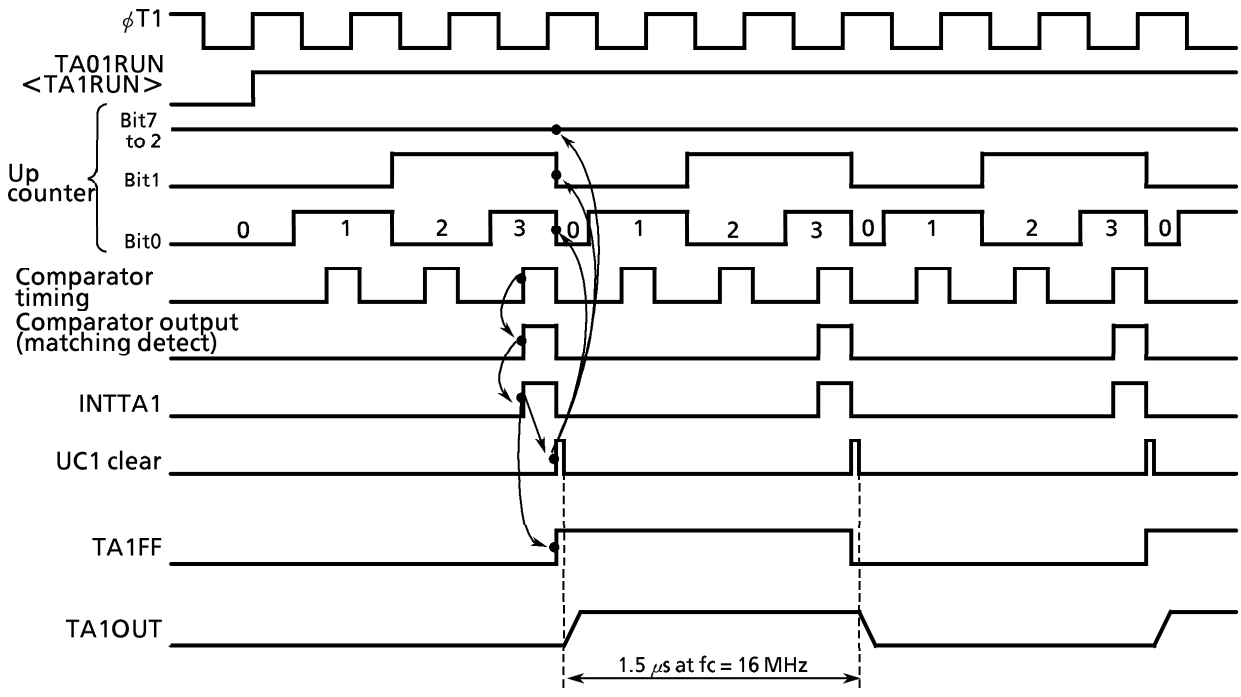


Figure 3.7.16 Square Wave (50% duty) Output Timing Chart

③ Making TMRA1 count up by match signal from TMRA0 comparator

Set the 8-bit timer mode, and set the comparator output of TMRA0 as the input clock to TMRA1.

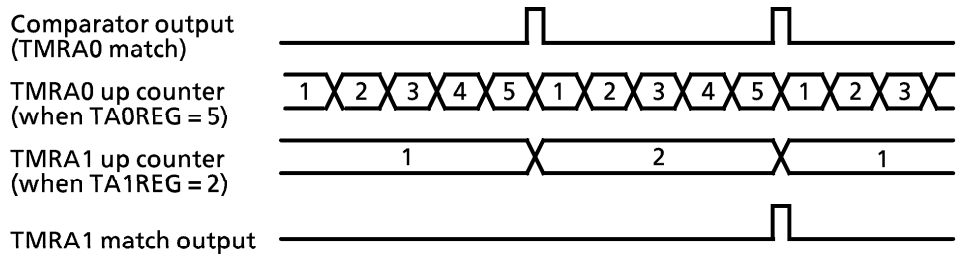


Figure 3.7.17 TMRA1 Count Up by TMRA0

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of TMRA0 and TMRA1.
To make a 16-bit interval timer by cascade connecting TMRA0 and TMRA1, set TA01MOD <TA01M1:0> to 01.
When set in 16-bit timer mode, the overflow output of TMRA0 will become the input clock of TMRA1 regardless of the set value of TA01MOD<TA1CLK1:0>. Table 3.7.2 shows the relation between the cycle of timer (Interrupt) and the selection of input clock.

(Setting example)

To generate an interrupt INTTA1 every 0.5 [s] at fc=16 MHz, set the following values for timer registers TA0REG and TA1REG.

※ Clock condition

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

When counting with input clock of ϕT16 (8.0 μs at 16 MHz)

0.5 s ÷ 8.0 μs = 62500 = F424H

Therefore, set TA1REG = F4H and TA0REG = 24H, respectively.

The comparator match signal is output from TMRA0 each time the up counter UC0 matches TA0REG, where the up counter UC0 is not be cleared.
With the TMRA1 comparator, the match detect signal is output at each comparator timing when up counter UC1 and TA1REG values match. When the match detect signal is output simultaneously from both comparators of TMRA0 and TMRA1, the up counters UC0 and UC1 are cleared to 0, and the interrupt INTTA1 is generated. If inversion is enabled, the value of the timer flip-flop TA1FF is inverted.

Example: When TA1REG = 04H and TA0REG = 80H

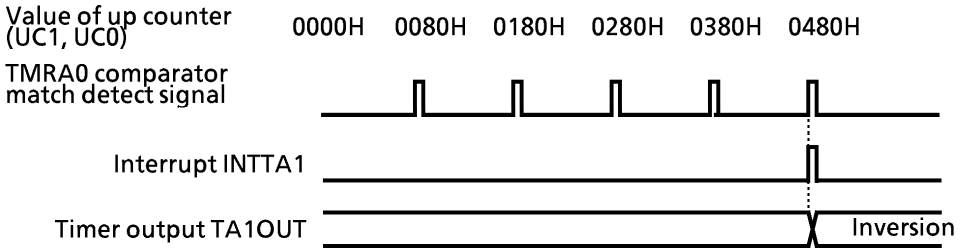


Figure 3.7.18 Timer Output by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable pulse generation) output mode

Square wave pulse can be generated at any frequency and duty by TMRA0. The output pulse may be either low active or high active. In this mode, TMRA1 cannot be used.
TMRA0 outputs pulse to TA1OUT pin (also used as P71).

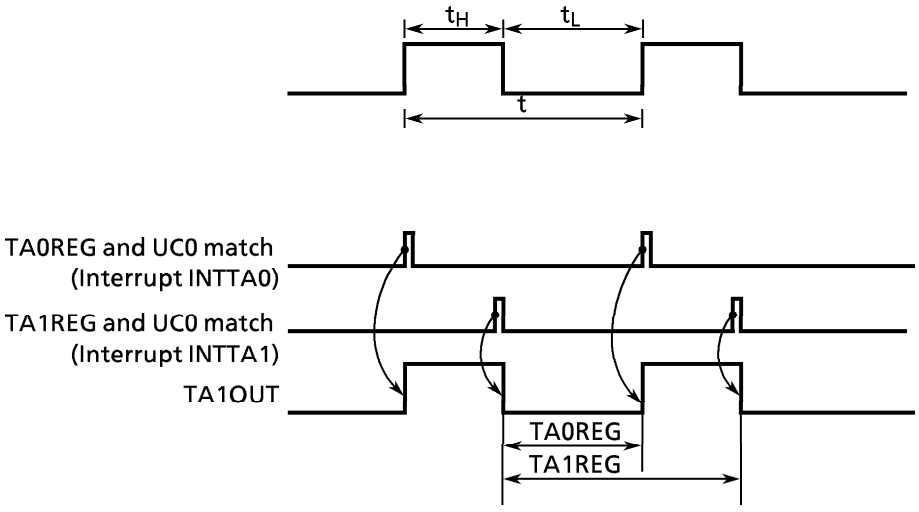


Figure 3.7.19 8-Bit PPG Output Waveforms

In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up counter (UC0) matches the timer registers TA0REG and TA1REG.
However, it is required that the set value of TA0REG is smaller than that of TA1REG.
Though the up counter (UC1) of TMRA1 is not used in this mode, UC1 should be set for counting by setting TA01RUN<TA1RUN> to 1.
Figure 3.7.20 shows the block diagram for this mode.

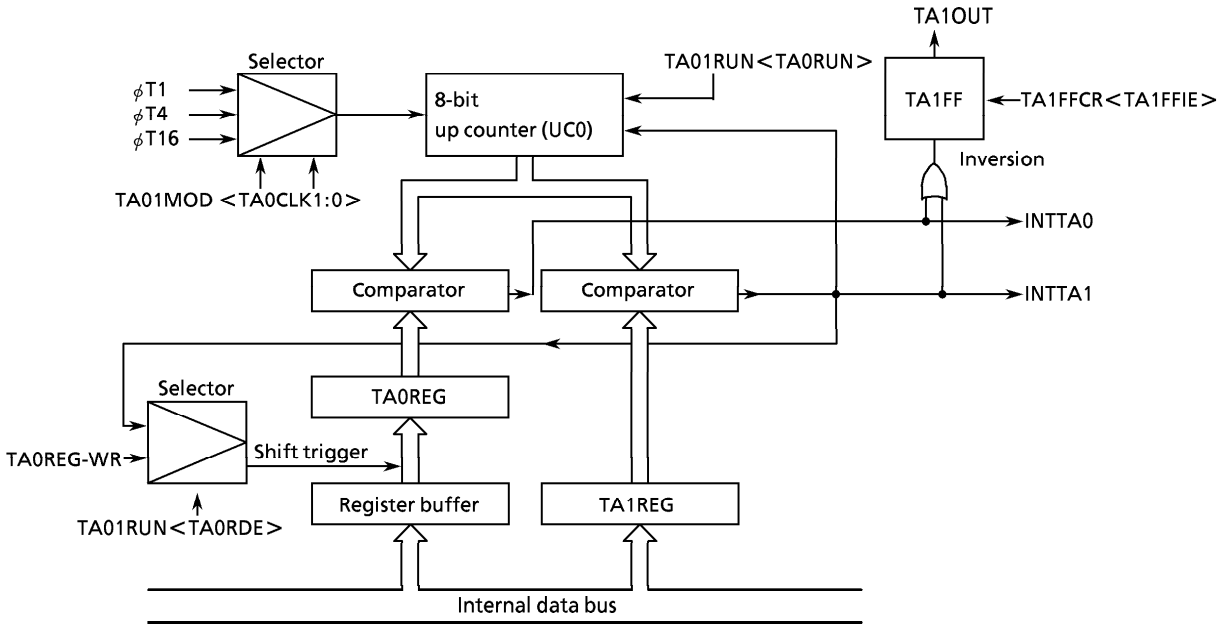


Figure 3.7.20 Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TA0REG is enabled in this mode, the value of register buffer will be shifted in TA0REG each time TA1REG matches UC0.
Use of the double buffer makes easy the handling of low duty waves (when duty is varied).

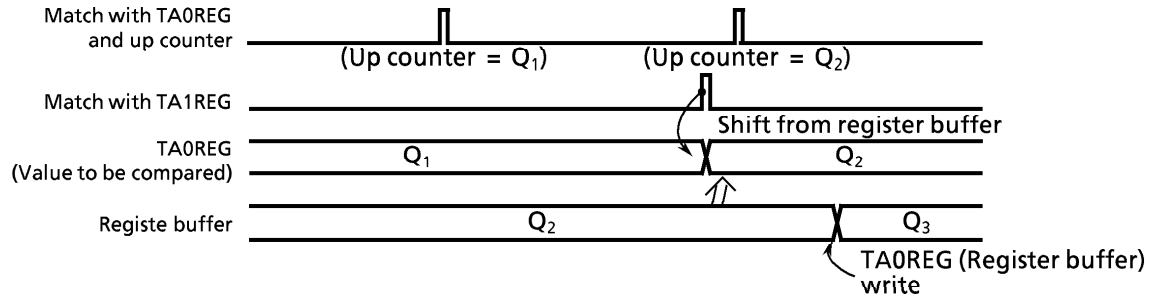
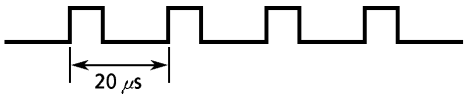


Figure 3.7.21 Operation of Register Buffer

Example: Generating 1/4 duty 50 kHz pulse (at fc = 16 MHz)



※ Clock condition
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

- Calculate the value to be set for timer register.
  To obtain the frequency 50 kHz, the pulse cycle t should be : t = 1/50 kHz = 20 μs.
  Given ϕT1 = 0.5 μs (at 16 MHz),
  20 μs ÷ 0.5 μs = 40
  Consequently, to set TA1REG = 40 = 28H
  and then duty to 1/4, t × 1/4 = 20 μs × 1/4 = 5 μs
  5 μs ÷ 0.5 μs = 10
  Therefore, set TA0REG = 10 = 0AH.

```
          7 6 5 4 3 2 1 0
TA01RUN ← - X X X - - 0 0   Stop TMRA0 and TMRA1 and clear it to 0.
TA01MOD ← 1 0 X X X X 0 1   Set the 8-bit PPG mode, and select φT1 as input clock.
TA0REG  ← 0 0 0 0 1 0 1 0   Write 0AH.
TA1REG  ← 0 0 1 0 1 0 0 0   Write 28H.
TA1FFCR ← X X X X 0 1 1 X   Sets TA1FF and enable the inversion and double buffer
                            enable.
                            Writing 10 provides negative logic pulse.
P7CR    ← X X - - - - 1 -   Set P71 as the TA1OUT pin.
P7FC    ← X X - - X - 1 X
TA01RUN ← 1 X X X - 1 1 1   Start TMRA0 and TMRA1 counting.
```

X: Don't care, −: No change

(4) 8-bit PWM output mode

This mode is valid only for TMRA0. In this mode, maximum 8-bit resolution of PWM pulse can be output.
PWM pulse is output to TA1OUT pin (also used as P71) when using TMRA0. TMRA1 can also be used as 8-bit timer.
Timer output is inverted when up counter (UC0) matches the set value of timer register TA0REG or when 2^n (n = 6, 7, or 8; specified by TA01MOD<PWM01:00>) counter overflow occurs. Up counter UC0 is cleared when 2^n counter overflow occurs.
To use this PWM mode, the following conditions must be satisfied.

(Set value of TA0REG) < (Set value of 2^n counter overflow)
(Set value of TA0REG) ≠ 0

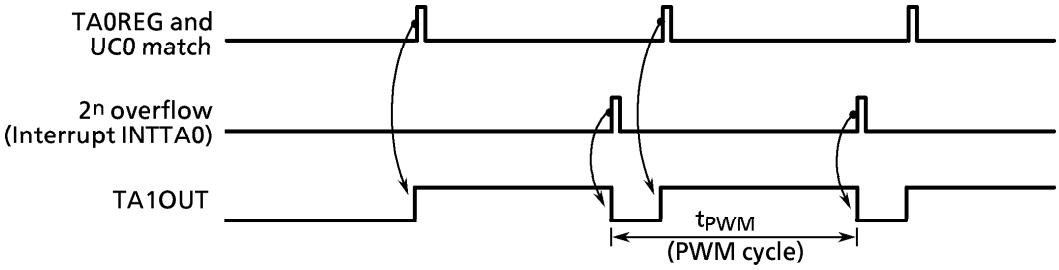


Figure 3.7.22 8-Bit PWM Waveforms

Figure 3.7.23 shows the block diagram of this mode.

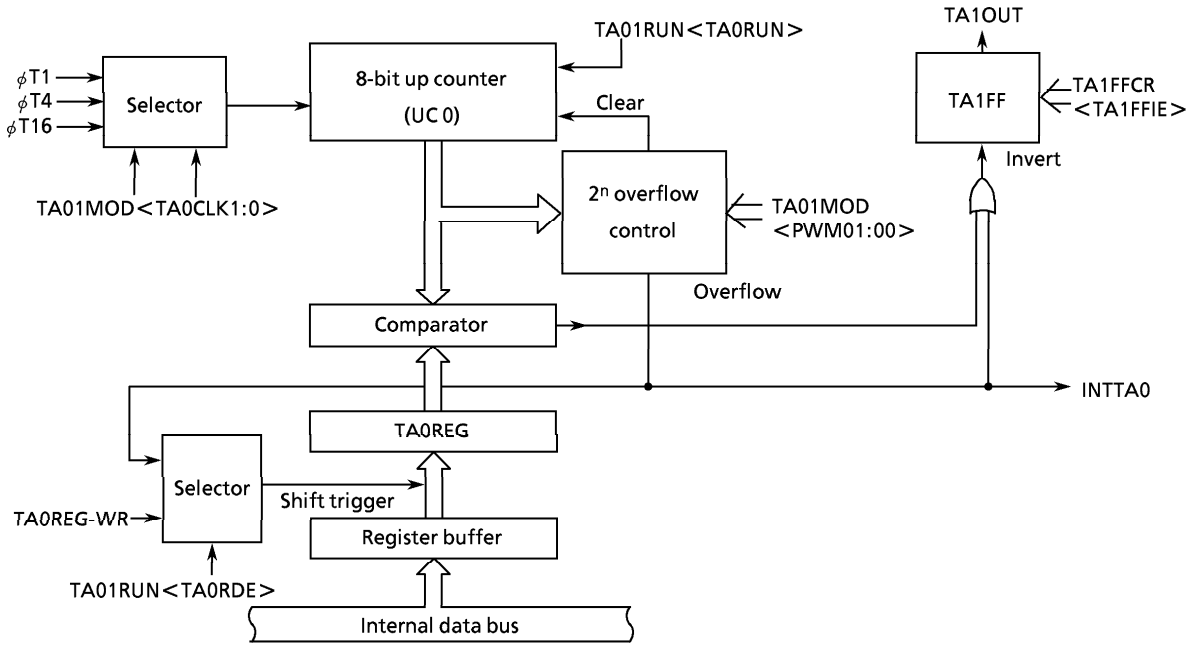


Figure 3.7.23 Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TA0REG if 2^n overflow is detected when the double buffer of TA0REG is enabled.
Use of the double buffer makes easy the handling of small duty waves.

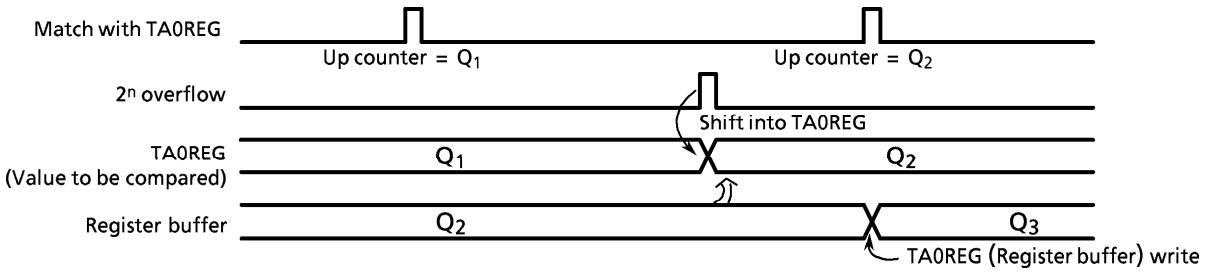
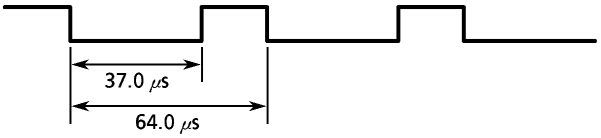


Figure 3.7.24 Operation of Register Buffer

(Setting example)

To output the following PWM waves to TA1OUT pin at fc = 16 MHz.



※ Clock condition

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

To realize 64.0 μs of PWM cycle by $\phi T1 = 0.5$ μs (at fc = 16 MHz),

$64.0\ \mu s \div 0.5\ \mu s = 128 = 2^n$

Consequently, n should be set to 7.
As the period of low level is 37.0 μs, for $\phi T1 = 0.5$ μs,
set the following value for TA0REG.

$37.0\ \mu s \div 0.5\ \mu s = 74 = 4AH$

```
          MSB              LSB
            7 6 5 4 3 2 1 0
TA01RUN ← – X X X – – – 0      Stop TMRA0, and clear it to 0.
TA01MOD ← 1 1 1 0 X X 0 1      Set 8-bit PWM mode (Cycle: 27) and select φT1 as the input clock.
TA0REG  ← 0 1 0 0 1 0 1 0      Writes 4AH.
TA1FFCR ← X X X X 1 0 1 X      Clears TA1FF, enable the inversion and double buffer.

P7CR    ← X X – – – – 1 –  }   Set P71 as the TA1OUT pin.
P7FC    ← X X – – X – 1 X  }
TA01RUN ← 1 X X X – 1 – 1      Start TMRA0 counting.
```

X: Don't care, –: No change

Table 3.7.3 PWM Cycle

at fc = 16 MHz, fs = 32.768 kHz

Select System Clock SYSCR1 <SYSCK>	Select Prescaler Clock SYSCR0 <PRCK1:0>	Gear Value SYSCR1 <GEAR2:0>	PWM Cycle								
			2^6			2^7			2^8		
			ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
1 (fs)	00 (f_{FPH})	XXX	15.6 ms	62.5 ms	250 ms	31.3 ms	125 ms	500 ms	62.5 ms	250 ms	1000 ms
0 (fc)		000 (fc)	32.0 μs	128 μs	512 μs	64.0 μs	256 μs	1024 μs	128 μs	512 μs	2048 μs
		001 (fc/2)	64.0 μs	256 μs	1024 μs	128 μs	512 μs	2048 μs	256 μs	1024 μs	4096 μs
		010 (fc/4)	128 μs	512 μs	2048 μs	256 μs	1024 μs	4096 μs	512 μs	2048 μs	8192 μs
		011 (fc/8)	256 μs	1024 μs	4096 μs	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms
		100 (fc/16)	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms	2048 μs	8192 μs	32.768 ms
	10 (fc/16 clock)	XXX	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms	2048 μs	8192 μs	32.768 ms

XXX: Don't care

(5) Setting for each mode

Table 3.7.4 shows the setting SFR for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit symbol>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	TA1FFIS
Function	Timer Mode	PWM Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
8-bit timer × 2-channels	00	–	Lower timer match ϕT1, ϕT16, ϕT256 (00, 01, 10, 11)	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
16-bit timer mode	01	–	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit PPG × 1-channel	10	–	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit PWM × 1-channel	11	2^6, 2^7, 2^8 (01, 10, 11)	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 1-channel	11	–	ϕT1, ϕT16 , ϕT256 (01, 10, 11)	–	Output disabled

– : Don't care

3.8 16-Bit Timers/Event Counters (TMRB)

The TMP91CW12 contains two (TMRB0, TMRB1) multifunctional 16-bit timer/event counter with the following operation modes.

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) mode
  Can be used following operation modes by capture function.
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

Timer/event counter consists of 16-bit up counter, two 16-bit timer registers (One of them applies double buffer), two 16-bit capture registers, two comparators, capture input controller, and timer flip-flop and the control circuit.
Timer/event counter is controlled by 11 bytes of control registers (SFRs).
2 channels (TMRB0, TMRB1) can be used independently.
Both channel has the same operation except the Table 3.8.1 items. So, only the operation of TMRB0 will be explained below.
This chapter consists of following items.

Table 3.8.1 Difference between TMRB0 and TMRB1

Spec \ Ch		TMRB0	TMRB1
External Pins	External clock/capture trigger input pin	TB0IN0 (Also used as P80) TB0IN1 (Also used as P81)	TB1IN0 (Also used as P84) TB1IN1 (Also used as P85)
	Timer flip-flop output pin	TB0OUT0 (Also used as P82) TB0OUT1 (Also used as P83)	TB1OUT0 (Also used as P86) TB1OUT1 (Also used as P87)
SFR (Address)	Timer RUN register	TB0RUN (0180H)	TB1RUN (0190H)
	Timer mode register	TB0MOD (0182H)	TB1MOD (0192H)
	Timer flip-flop control register	TB0FFCR (0183H)	TB1FFCR (0193H)
	Timer register	TB0RG0L (0188H) TB0RG0H (0189H) TB0RG1L (018AH) TB0RG1H (018BH)	TB1RG0L (0198H) TB1RG0H (0199H) TB1RG1L (019AH) TB1RG1H (019BH)
	Capture register	TB0CP0L (018CH) TB0CP0H (018DH) TB0CP1L (018EH) TB0CP1H (018FH)	TB1CP0L (019CH) TB1CP0H (019DH) TB1CP1L (019EH) TB1CP1H (019FH)

3.8.1 Block Diagram

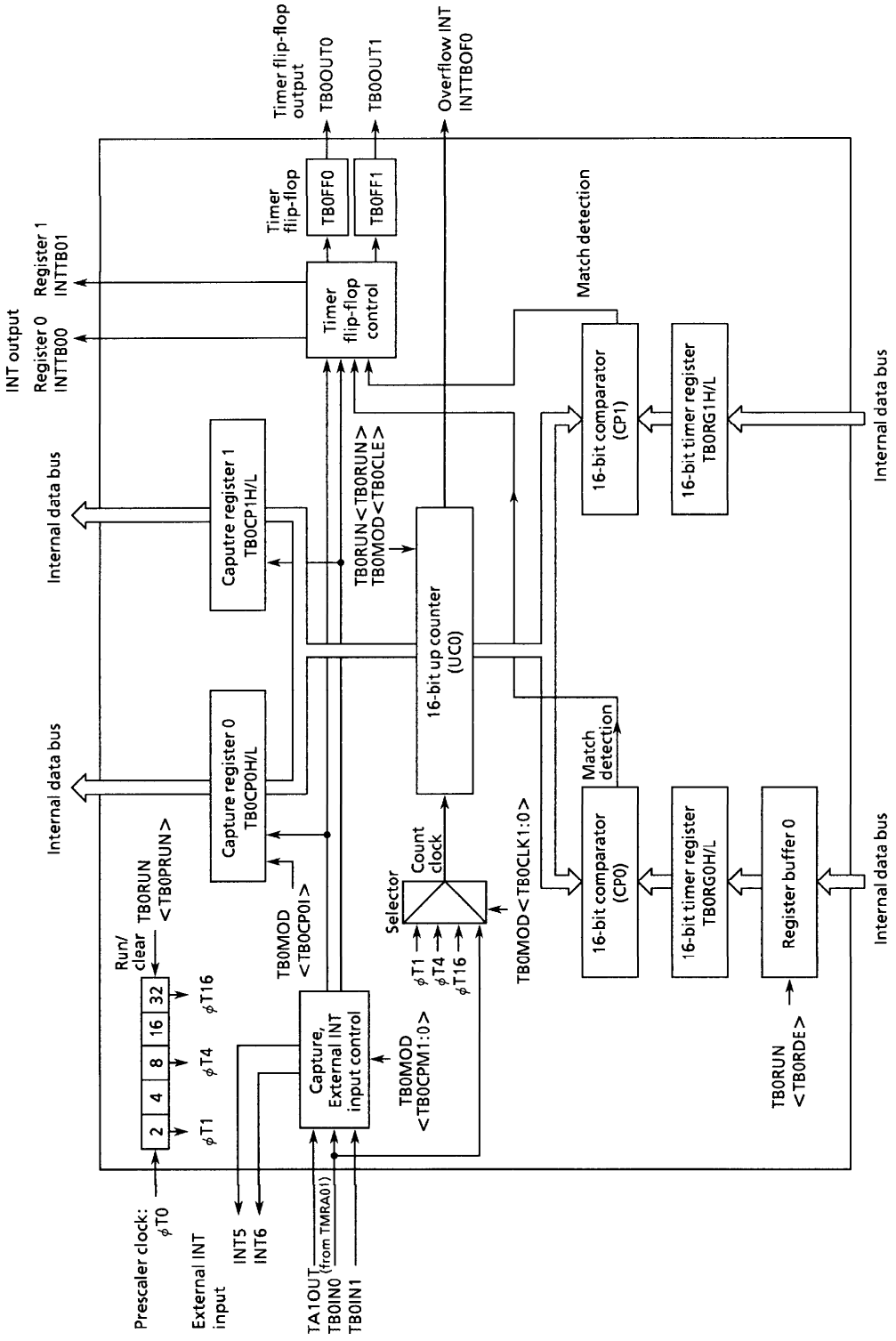


Figure 3.8.1 TMRB0 Block Diagram

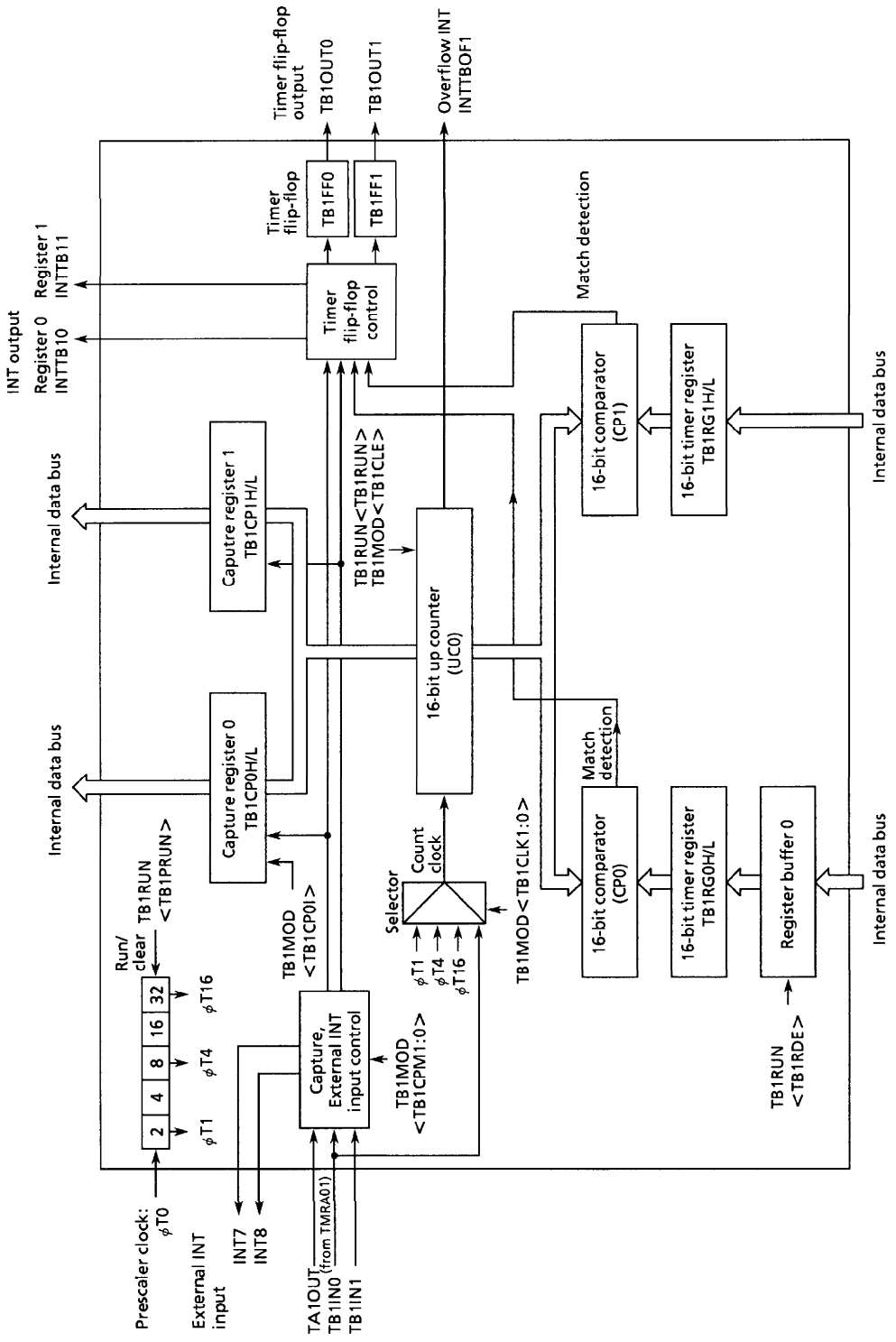


Figure 3.8.2 TMRB1 Block Diagram

3.8.2 Operations of Each Blocks

(1) Prescaler

The 5-bit prescaler to generate source clock of TMRB0. Prescaler clock (ϕT0) is divided clock (divided by 4) from selected clock by the register SYSCR0<PRCK1:0> of clock gear.
This prescaler can be run on stopped by the register TB0RUN<TB0RUN>. Counting starts when <TB0RUN> is set to 1, while the prescaler is cleared to zero and stops operation when <TB0RUN> is set to 0.

Table 3.8.2 Prescaler Clock Resolution

at fc = 16 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Prescaler Clock Selection <PRCK1:0>	Clock Gear Value <GEAR2:0>	Prescaler Clock Resolution		
			ϕT1	ϕT4	ϕT16
1 (fs)	00 (f_{FPH})	XXX	fs/2^3 (244 μs)	fs/2^5 (977 μs)	fs/2^7 (3.9 ms)
0 (fc)		000 (fc)	fc/2^3 (0.5 μs)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)
		001 (fc/2)	fc/2^4 (1.0 μs)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)
		010 (fc/4)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)
		011 (fc/8)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{10} (64 μs)
		100 (fc/16)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)
	10 (Note) (fc/16 clock)	XXX	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)

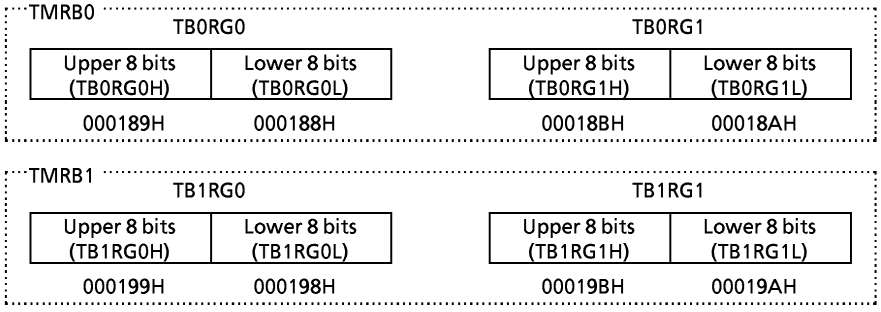
XXX : Don't care

(2) Up counter (UC0)

UC0 is a 16-bit binary counter which counts up according to the input clock specified by TB0MOD <TB0CLK1:0> register.
As the input clock, one of the internal clocks ϕT1, ϕTB0, and ϕT16 from prescaler and external clock from TB0IN0 pin can be selected. Counting or stop & clear of the counter is controlled by timer operation control register TB0RUN<TB0RUN>.
When clearing is enabled, up counter UC0 will be cleared to zero each time it coincides matches the timer register TB0RG1H/L. The "clear enable/disable" is set by TB0MOD<TB0CLE>.
If clearing is disabled, the counter operates as a free-running counter.
A timer overflow interrupt (INTTBOF0) is generated when UC0 overflow occurs.

(3) Timer registers (TB0RG0H/L, TB0RG1H/L)

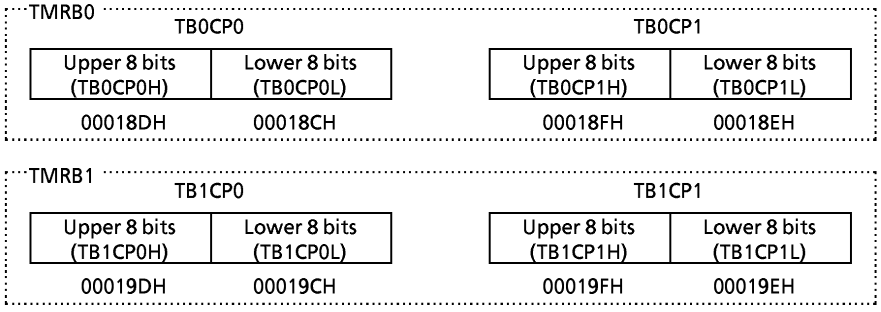
These two 16-bit registers are used to set the interval time. When the value of up counter UC0 matches the set value of this timer register, the comparator match detect signal will be active.
Setting data for both upper and lower registers is always needed. For example, either using 2-byte date transfer instruction or using 1-byte date transfer instruction twice for lower 8 bits and upper 8 bits in order.
TB0RG0 timer register is of double buffer structure, which is paired with register buffer. The timer control register TB0RUN<TB0RDE> controls whether the double buffer structure should be enabled or disabled. : disabled when <TB0RDE> = 0, while enabled when <TB0RDE> = 1.
When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up counter (UC0) and timer register TB0RG1.
After reset, TB0RG0 and TB0RG1 are undefined. To use the 16-bit timer after reset, data should be written beforehand.
When reset, it will be initialized to <TB0RDE> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <TB0RDE> = 1, and then write the following data in the register buffer.
TB0RG0 and register buffer are allocated to the same memory addresses 000188H/000189H. When <TB0RDE> = 0, same value will be written in both the timer register and register buffer. When <TB0RDE> = 1, the value is written into only the register buffer.
The address of each timer register is as following.



The timer registers are write-only registers, so they can not be read.

(4) Capture register (TB0CP0H/L, TB0CP1H/L)

These 16-bit registers are used to latch the values of the up counter.
Data in the capture registers should be read both upper and lower 16 bits. For example, using a 2-byte data load instruction or two 1-byte data load instruction, from the lower 8 bits followed by the upper 8 bits.
The address of each capture register is as following.

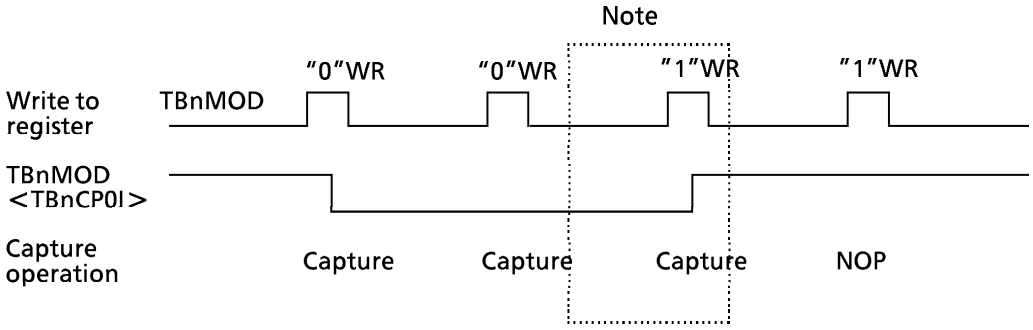


These capture registers are read-only registers, so it cannot be written.

(5) Capture input control and external interrupt control (INT5, INT6)

This circuit controls the timing to latch the value of up-counter UC0 into TB0CP0, TB0CP1 and the generating for external interrupts. The latch timing of capture register and selection of edge for external interrupt is controlled by register TB0MOD<TB0CPM1:0>.
The edge of external interrupt INT6 is fixed to rise edge.
Besides, the value of up counter can be loaded to capture registers by software. Whenever 0 is written in TB0MOD<TB0CP0I> the current value of up-counter will be loaded to capture register TB0CP0. It is necessary to keep the prescaler in RUN mode (TB0RUN<TB0PRUN> to be 1).

Note: As described above, whenever 0 is written to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0. However, note that the current value in the up counter is also loaded into capture register TB0CP0 when 1 is written to TB0MOD<TB0CP0I> while this bit is holding 0.



(6) Comparator (CP0, CP1)

These are 16-bit comparators which compare the up counter UC0 value with the set value of (TB0RG0, TB0RG1) to detect the match. When a match is detected, the comparators generate an interrupt (INTTB00, INTTB01) respectively.

(7) Timer flip-flop (TB0FF0, TB0FF1)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>. After reset, the value of TB0FF0 is undefined. TB0FF0 will be inverted when 00 is written in TB0FFCR<TB0FF0C1:0>, <TB0FF1C1:0>. Also it is set to 1 when 01 is written, and set to 0 when 10 is written. The value of TB0FF0, TB0FF1 can be output to the timer output pin TB0OUT0 (shared by P82), TB0OUT1 (shared by P83). Timer output should be specified by the function register of port 8.

3.8.3 SFR

TMRB0 Run Register

TB0RUN (0180H)

	7	6	5	4	3	2	1	0
Bit symbol	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After reset	0	0			0	0		0
Function	Double buffer 0: Disable 1: Enable	Write 0			IDLE2 0: Stop 1: Operate	16-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Count operation

0	Stop and clear
1	Count

I2TB0: Operation during IDLE2 mode
TB0PRUN: Operation of prescaler
TB0RUN: Operation of TMRB0

Note: The 1, 4 and 5 of TB0RUN are read as undefined value.

TMRB1 Run Register

TB1RUN (0190H)

	7	6	5	4	3	2	1	0
Bit symbol	TB1RDE	–			I2TB1	TB1PRUN		TB1RUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After reset	0	0			0	0		0
Function	Double buffer 0: Disable 1: Enable	Write 0			IDLE2 0: Stop 1: Operate	16-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Count operation

0	Stop and clear
1	Count

I2TB1: Operation during IDLE2 mode
TB1PRUN: Operation of prescaler
TB1RUN: Operation of TMRB1

Note: The 1, 4 and 5 of TB1RUN are read as undefined value.

Figure 3.8.3 The Registers for TMRB

TMRB0 Mode Register

		7	6	5	4	3	2	1	0
TB0MOD (0182H)	Bit symbol	TB0CT1	TB0ET1	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
	Read/Write	R/W		W*	R/W		R/W	R/W	
	After reset	0	0	1	0	0	0	0	0
	Function	TB0FF1 inversion trigger 0: Disable trigger 1: Enable trigger		Execute software capture 0: Execute 1: Un-defined	Capture timing 00: Disable INT5 occurs at rising edge. 01: TB0IN0 ↑ TB0IN1 ↑ INT5 occurs at rising edge. 10: TB0IN0 ↑ TB0IN0 ↓ INT5 occurs at falling edge. 11: TA0TRG ↑ TA0TRG ↓ INT5 occurs at rising edge.		Control up counter 0: Disable clearing 1: Enable clearing	TMRB0 source clock 00: TB0IN0 pin 01: ϕT1 10: ϕT4 11: ϕT16	
Read-modify-write instruction is prohibited.		Invert when the UC value is captured to TB0CP1.	Invert when the UC matches TB0RG1.						

TMRB0 source clock

00	TB0IN0 pin
01	ϕT1
10	ϕT4
11	ϕT16

Control clearing for up counter

0	Disable
1	Enable clearing by match with TB0RG1

Capture/Interrupt timing

	Capture control	INT5 control
00	Disable	INT5 occurs at the rise edge of TB0IN0
01	CAP0 at TB0IN0 rise CAP1 at TB0IN1 rise	
10	CAP0 at TB0IN0 rise CAP1 at TB0IN0 fall	INT5 occurs at the fall edge of TB0IN0
11	CAP0 at TA0TRG rise CAP1 at TA0TRG fall	INT5 occurs at the rise edge of TB0IN0

Software capture

0	The value of up counter is captured to TB0CP0
1	Undefined (Note)

Figure 3.8.4 The Registers for TMRB

Note: As described above, whenever 0 is written to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0. However, note that the current value in the up counter is also loaded into capture register TB0CP0 when 1 is written to TB0MOD<TB0CP0I> while this bit is holding 0.

TMRB1 Mode Register

		7	6	5	4	3	2	1	0
TB1MOD (0192H)	Bit symbol	TB1CT1	TB1ET1	TB1CP0I	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
	Read/Write	R/W		W*	R/W		R/W	R/W	
	After reset	0	0	1	0	0	0	0	0
	Function	TB1FF1 inversion trigger 0: Disable trigger 1: Enable trigger		Execute software capture 0: Execute 1: Un-defined	Capture timing 00: Disable INT7 occurs at rising edge. 01: TB1IN0 ↑ TB1IN1 ↑ INT5 occurs at rising edge. 10: TB1IN0 ↑ TB1IN0 ↓ INT7 occurs at falling edge. 11: TA0TRG ↑ TA0TRG ↓ INT5 occurs at rising edge.		Control up counter 0: Disable clearing 1: Enable clearing	TMRB0 source clock 00: TB1IN0 pin 01: ϕT1 10: ϕT4 11: ϕT16	
Read-modify-write instruction is prohibited.		Invert when the UC value is captured to TB1CP1.	Invert when the UC matches TB1RG1.						

TMRB1 source clock

00	TB1IN0 pin
01	ϕT1
10	ϕT4
11	ϕT16

Control clearing for up counter

0	Disable
1	Enable clearing by match with TB1RG1

Capture/Interrupt timing

	Capture control	INT7 control
00	Disable	INT7 occurs at the rise edge of TB1IN0
01	CAP2 at TB1IN0 rise CAP3 at TB1IN1 rise	
10	CAP2 at TB1IN0 rise CAP3 at TB1IN0 fall	INT7 occurs at the fall edge of TB1IN0
11	CAP2 at TA0TRG rise CAP3 at TA0TRG fall	INT7 occurs at the rise edge of TB1IN0

Software capture

0	The value of up counter is captured to TB1CP0
1	Undefined (Note)

Figure 3.8.5 The Registers for TMRB

Note: As described above, whenever 0 is written to TB1MOD<TB1CP0I>, the current value in the up counter is loaded into capture register TB1CP0. However, note that the current value in the up counter is also loaded into capture register TB1CP0 when 1 is written to TB1MOD<TB1CP0I> while this bit is holding 0.

TMRB0 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB0FFCR (0183H)	Bit symbol	TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify-write instruction is prohibited.	Function	Control TB0FF1 00 : Invert 01 : Set 10 : Clear 11 : Don't care ※ Always read as 11.		TB0FF0 inversion trigger 0: Disable trigger 1: Enable trigger				Control TB0FF0 00 : Invert 01 : Set 10 : Clear 11 : Don't care ※ Always read as 11.	
				Invert when the UC value is loaded to TB0CP1.	Invert when the UC value is loaded to TB0CP0.	Invert when the UC value is matches to TB0RG1.	Invert when the UC value is matches to TB0RG0.		

Control TB0FF0

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

Invert when the UC value is loaded to TB0CP1.

0	Disable trigger
1	Enable trigger

Invert when the UC value is loaded to TB0CP0.

0	Disable trigger
1	Enable trigger

Invert when the UC value is matches to TB0RG1.

0	Disable trigger
1	Enable trigger

Invert when the UC value is matches to TB0RG0.

0	Disable trigger
1	Enable trigger

Figure 3.8.6 The Registers for TMRB

TMRB1 Flip-flop Control Register

TB1FFCR (0193H)

	7	6	5	4	3	2	1	0
Bit symbol	TB1FF1C1	TB1FF1C0	TB1C1T1	TB1C0T1	TB1E1T1	TB1E0T1	TB1FF0C1	TB1FF0C0
Read/Write	W*		R/W				W*	
After reset	1	1	0	0	0	0	1	1
Function	Control TB1FF1 00 : Invert 01 : Set 10 : Clear 11 : Don't care ※ Always read as 11.		TB1FF0 inversion trigger 0: Disable trigger 1: Enable trigger				Control TB1FF0 00 : Invert 01 : Set 10 : Clear 11 : Don't care ※ Always read as 11.	
Read-modify-write instruction is prohibited.			Invert when the UC value is loaded to TB1CP1.	Invert when the UC value is loaded to TB1CP0.	Invert when the UC value is matches to TB1RG1.	Invert when the UC value is matches to TB1RG0.		

Control TB1FF0

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

Invert when the UC value is loaded to TB1CP1.

0	Disable trigger
1	Enable trigger

Invert when the UC value is loaded to TB1CP0.

0	Disable trigger
1	Enable trigger

Invert when the UC value is matches to TB1RG1.

0	Disable trigger
1	Enable trigger

Invert when the UC value is matches to TB1RG0.

0	Disable trigger
1	Enable trigger

Figure 3.8.7 The Registers for TMRB

3.8.4 Operation of Each Mode

(1) 16-bit timer mode

Generating interrupts at fixed intervals
In this example, the interval time is set in the timer register TB0RG1 to generate the interrupt INTTB01.

```
                7 6 5 4 3 2 1 0
TB0RUN    ← - 0 X X - - X 0     Stop TMRB0.
INTETB0   ← X 1 0 0 X 0 0 0     Enable INTTB01 and sets interrupt level 4. Disable
                                INTTB00.
TB0FFCR   ← 1 1 0 0 0 0 1 1     Disable trigger.
TB0MOD    ← 0 0 1 0 0 1 * *     Select internal clock for input and
                (**=01, 10, 11) disable the capture function.
TB0RG1    ← * * * * * * * *     Set the interval time (16 bits).
                * * * * * * * *
TB0RUN    ← - 0 X X - 1 X 1     Start TMRB0.
```

X: Don't care, −: No change

(2) 16-bit event counter mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TB0IN0 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.

```
                7 6 5 4 3 2 1 0
TB0RUN    ← - 0 X X - - X 0     Stop TMRB0.
P8CR      ← - - - - - - - 0     Set P80 to input mode for TB0IN0
P8FC      ← - - - - - - - 1
INTETB0   ← X 1 0 0 X 0 0 0     Enable INTTB01 and sets interrupt level 4, while
                                disables INTTB00.
TB0FFCR   ← 1 1 0 0 0 0 1 1     Disable trigger.
TB0MOD    ← 0 0 1 0 0 1 0 0     Select TB0IN0 as the input clock.
TB0RG1    ← * * * * * * * *     Set the number of counts (16 bits).
TB0RUN    ← - 0 X X - 1 X 1     Start TMRB0.
```

X: Don't care, −: No change
When used as an event counter, set the prescaler in RUN mode.
(TB0RUN<TB0PRUN>=1)

(3) 16-bit programmable pulse generation (PPG) output mode

Square wave pulse can be generated at any frequency and duty. The output pulse may be either low active or high active.
The PPG mode is obtained by inversion of the timer flip-flop TB0FF0 that is to be enabled by the match of the up counter UC0 with the timer register TB0RG0 or 1 and to be output to TB0OUT0. In this mode, the following conditions must be satisfied.

(Set value of TB0RG0) < (Set value of TB0RG1)

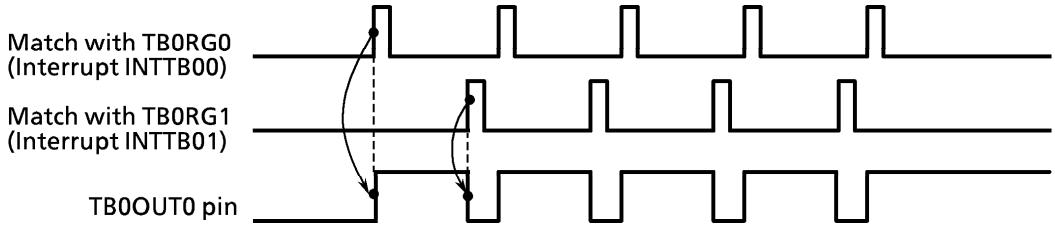


Figure 3.8.8 Programmable Pulse Generation (PPG) Output Waveforms

When the double buffer of TB0RG0 is enabled in this mode, the value of register buffer 0 will be shifted in TB0RG0 at match with TB0RG1. This feature makes easy the handling of low duty waves.

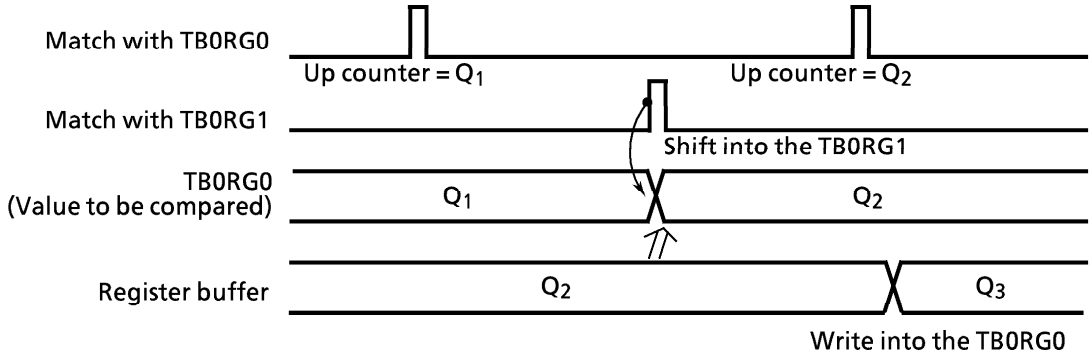


Figure 3.8.9 Operation of Register Buffer

Shows the block diagram of this mode.

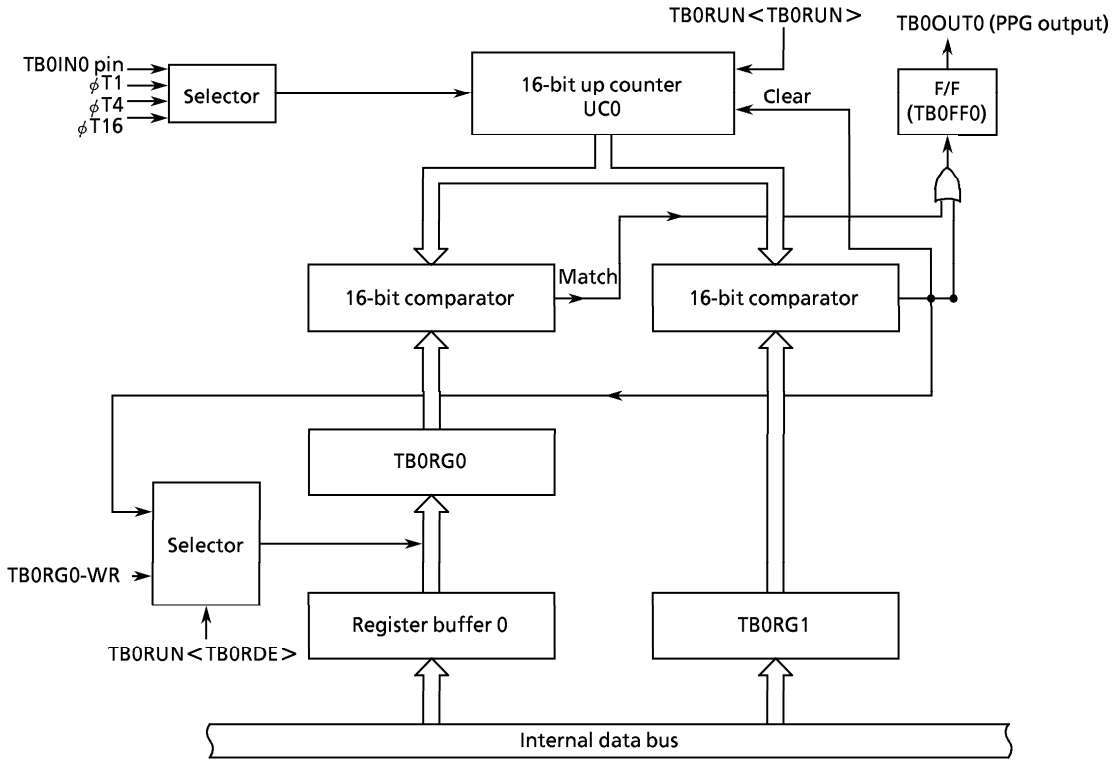


Figure 3.8.10 Block Diagram of 16-Bit PPG Mode

Setting example of 16-bit PPG output mode is as following.

```
                    7 6 5 4 3 2 1 0
TB0RUN    ← 0 0 X X - - X 0    Double Buffer of TB0RG0 disable and Stop TMRB0.
TB0RG0    ← * * * * * * * *    Set the duty (16 bits).
TB0RG1    ← * * * * * * * *    Set the cycle (16 bits).
TB0RUN    ← 1 0 X X - 0 X 0    Double Buffer of TB0RG0 enable
                               (Change the duty and cycle at the interrupt INTTB01)
TB0FFCR   ← X X 0 0 1 1 1 0    Set the mode to invert TB0FF0 at the match with
                               TB0RG0/TB0RG1, and also set the TB0FF0 to 0.
TB0MOD    ← 0 0 1 0 0 1 * *    Select the internal clock for the input, and disable
              (** = 01, 10, 11) the capture function.
P8CR      ← - - - - - 1 - -  }
T8FC      ← - - - - - 1 - -  }  Assign P82 as TB0OUT0.
TB0RUN    ← 1 0 X X - 1 X 1    Start TMRB0.
```

X: Don't care, −: No change

(4) Application examples of capture function

Used capture function, they can be applied in many ways, for example:

① One-shot pulse output from external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot pulse output from external trigger pulse

Set the up counter UC0 in free-running mode with the internal input clock, input the external trigger pulse from TB0IN0 pin, and load the value of up-counter into capture register TB0CP0 at the rise edge of the TB0IN0 pin.
When the interrupt INT5 is generated at the rise edge of TB0IN0 input, set the TB0CP0 value (c) plus a delay time (d) to TB0RG0 (=c+d), and set the above set value (c+d) plus a one-shot pulse width (p) to TB0RG1 (=c+d+p). When the interrupt INT5 occurs the TB0FFCR<TB0E1T1, TB0E0T1>register should be set 11 and that the TB0FF0 inversion is enabled only when the up counter value matches TB0RG0 or TB0RG1. When interrupt INTTB01 occurs, this inversion will be disabled after one-shot pulse is output.
The (c), (d) and (p) correspond to c, d and p in figure 3.8.11.

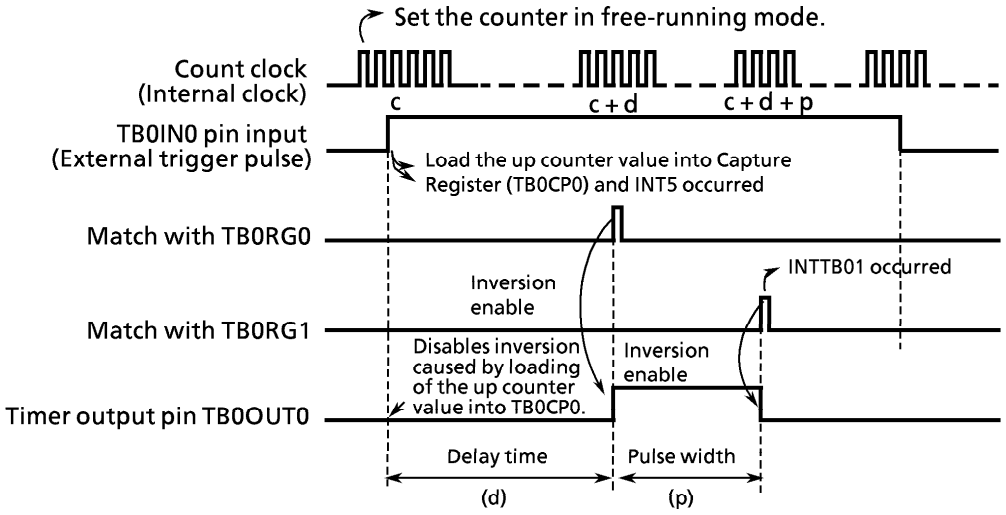


Figure 3.8.11 One-shot Pulse Output (with delay)

(Setting example)

To output 2 ms one-shot pulse with 3 ms delay to the external trigger pulse to TB0IN0 pin

※ Clock condition
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Main setting

```
TB0MOD   ←  X X 1 0 1 0 0 1   Keep counting (Free runnig)
                              Count with φT1.
TB0FFCR  ←  X X 0 0 0 0 1 0   Load the up counter value into TB0CP0 at the rise
                              edge of TB0IN0 pin input.
                              Clear TB0FF0 to 0.
                              Disable TB0FF0 inversion.
P8CR     ←  - - - - - - 1 -   Select P82 as the TB0OUT0 pin.
T8FC     ←  - - - - - - 1 -
INTE56   ←  X - - - X 1 0 0   Enable INT5, and disable INTTB00 and INTTB01.
INTETB0  ←  X 0 0 0 X 0 0 0
TB0RUN   ←  - 0 X X - 1 X 1   Start TMRB0.
```

Setting of INT5

```
TB0RG0   ←  TB0CP0+3ms/φT1
TB0RG1   ←  TB0RG0+2ms/φT1
TB0FFCR  ←  X X - - 1 1 - -   Enable TB0FF0 inversion when the up counter value
                              matches TB0RG0 or 1.
INTETB0  ←  X 1 0 0 X - - -   Enable INTTB01.
```

Setting of INTTB01

```
TB0FFCR  ←  X X - - 0 0 - -   Disable TB0FF0 inversion when the up counter value
                              matches TB0RG0 or 1.
INTETB0  ←  X 0 0 0 X - - -   Disable INTTB01.
```

X: Don't care, –: No change

When delay time is unnecessary, invert timer flip-flop TB0FF0 when the up counter value is loaded into capture register (TB0CP0), and set the TB0CP0 value (c) plus the one-shot pulse width (p) to TB0RG1 when the interrupt INT5 occurs. The TB0FF0 inversion should be enabled when the up counter (UC0) value matches TB0RG1, and disabled when generating the interrupt INTTB01.

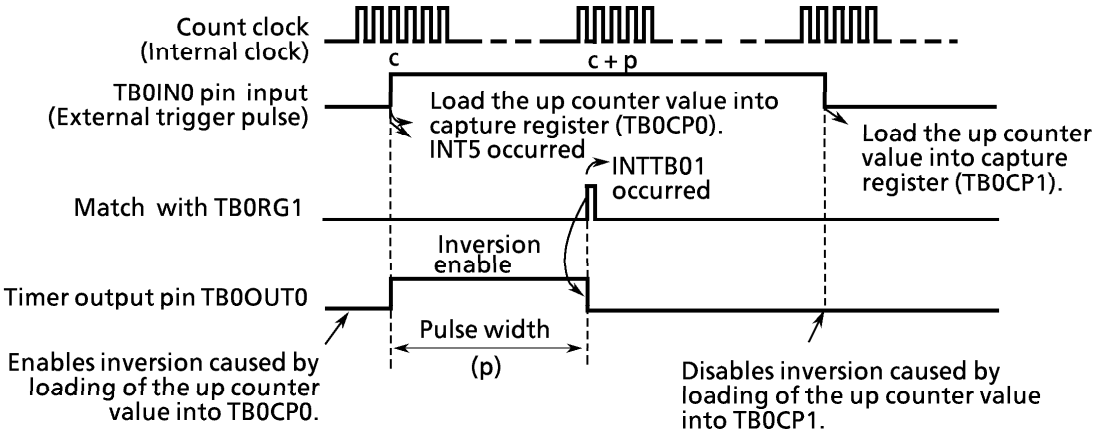


Figure 3.8.12 One-shot Pulse Output (without Delay)

② Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TB0IN0 pin, and its frequency is measured by the 8-bit timers TMRA01 and the 16-bit timer/event counter (TMRB0).

The TB0IN0 pin input should be selected for the input clock of TMRB0. Set to TB0MOD<TB0CPM1: 0>=11. The value of the up-counter is loaded into the capture register TB0CP0 at the rise edge of the timer flip-flop TA1FF of 8-bit timers (TMRA01), and into TB0CP1 at its fall edge.

The frequency is calculated by the difference between the loaded values in TB0CP0 and TB0CP1 when the interrupt (INTTA0 or INTTA1) is generated by either 8-bit timer.

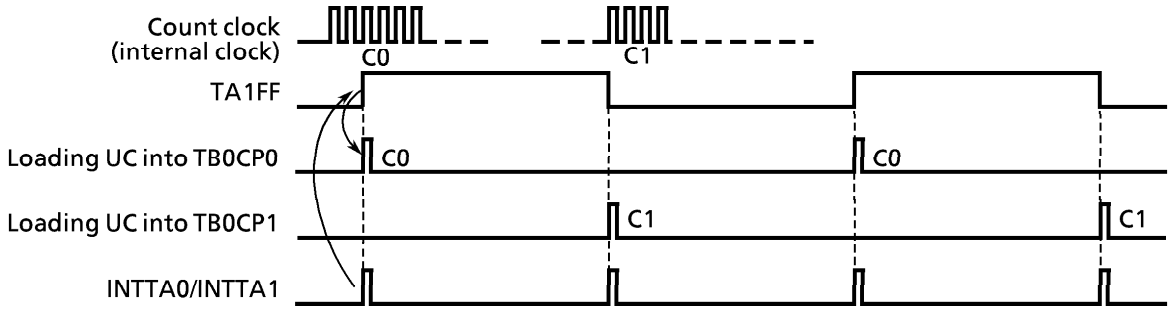


Figure 3.8.13 Frequency Measurement

For example, if the value for the level 1 width of TA1FF of the 8-bit timer is set to 0.5 [s] and the difference between TB0CP0 and TB0CP1 is 100, the frequency will be 100 ÷ 0.5 [s] = 200[Hz].

③ Pulse width measurement

This mode allows to measure the H level width of an external pulse. While keeping the 16-bit timer/event counter counting (Free running) with the internal clock input, the external pulse is input through the TB0IN0 pin. Then the capture function is used to load the UC0 values into TB0CP0 and TB0CP1 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT5 occurs at the falling edge of TB0IN0.
The pulse width is obtained from the difference between the values of TB0CP0 and TB0CP1 and the internal clock cycle.
For example, if the internal clock is 0.8 [μs] and the difference between TB0CP0 and TB0CP1 is 100, the pulse width will be 100 × 0.8 μs = 80 μs.
Additionally, the pulse width which is over the UC0 maximum count time specified by the clock source can be measured by changing software.

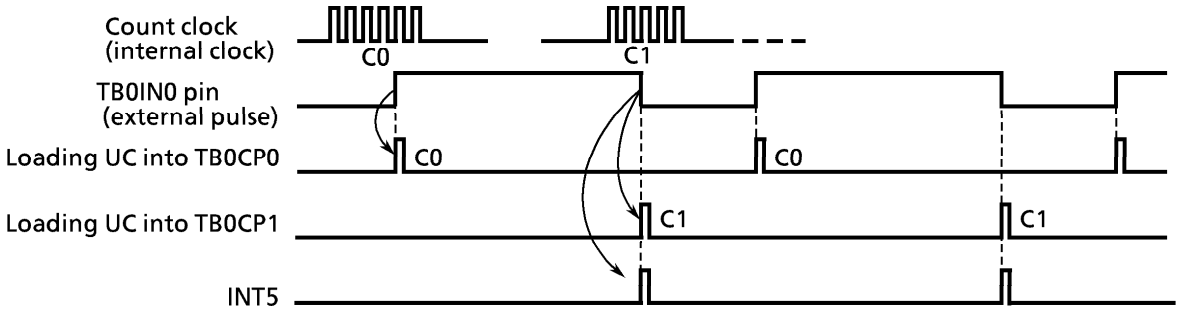


Figure 3.8.14 Pulse Width Measurement

Note: Only in this pulse width measuring mode (TB0MOD<TB0CPM1:0> = 10), external interrupt INT5 occurs at the falling edge of TB0IN0 pin input. In other modes, it occurs at the rising edge.

The width of L level can be measured from the difference between the first C1 and the second C0 at the second INT5 interrupt.
The width of L level can be measured by multiplying the difference between the first C1 and the second C0 at the second INT5 interrupt and the internal clock cycle together.

④ Time difference measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through TB0IN0 and TB0IN1.
Keep the 16-bit timer/event counter (TMRB0) counting (Free running) with the internal clock, and load the UC0 value into TB0CP0 at the rising edge of the input pulse to TB0IN0. Then the interrupt INT5 is generated.
Similarly, the UC0 value is loaded into TB0CP1 at the rising edge of the input pulse to TB0IN1, generating the interrupt INT6.
The time difference between these pulses can be obtained from the difference between the time counts at which loading the up counter value into TB0CP0 and TB0CP1 has been done.
The time difference between these pulses can be obtained by multiplying the value subtracted TB0CP0 from TB0CP1 and the internal clock cycle together at which loading the up counter value into TB0CP0 and TB0CP1 has been done.

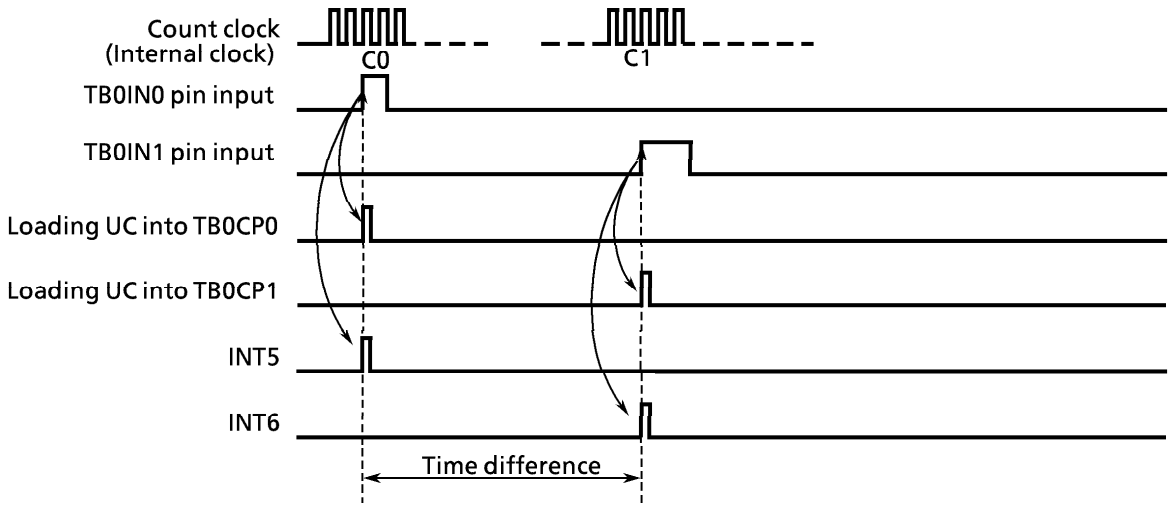


Figure 3.8.15 Time Difference Measurement

3.9 Serial Channel

TMP91CW12 contains 2 serial I/O channels. Both channels select UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission).

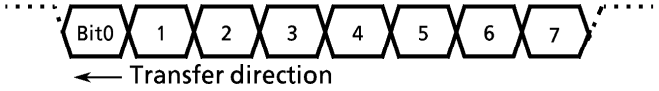
- I/O interface mode
  - Mode 0: To transmit and receive I/O data using the synchronizing signal SCLK for extending I/O.
- UART mode
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2, a parity bit can be added. Mode 3 has a wakeup function for making the master controller start slave controllers in a serial link (Multi-controller) system.
Figure 3.9.2, 3.9.3 show block diagram for each channel.
Serial channels 0 to 1 can be used independently.
All channels have the same operations except the following points, thus only the operation of channel 0 will be explained below.

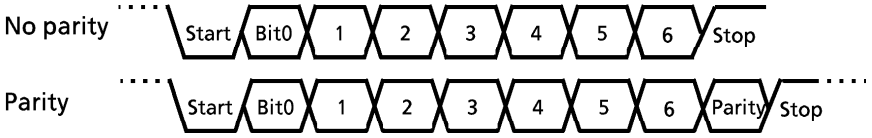
Table 3.9.1 Different Points between Channel 0 and 1

	Channel 0	Channel 1
Pin name	TXD0 (P90) RXD0 (P91) $\overline{\text{CTS0}}$/SCLK0 (P92)	TXD1 (P93) RXD1 (P94) $\overline{\text{CTS1}}$/SCLK1 (P95)
IrDA mode	Yes	No

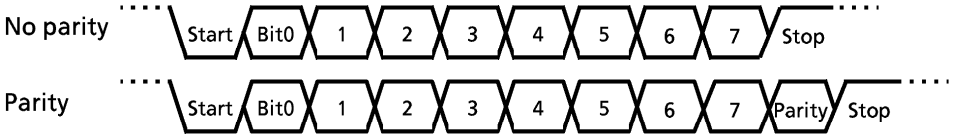
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

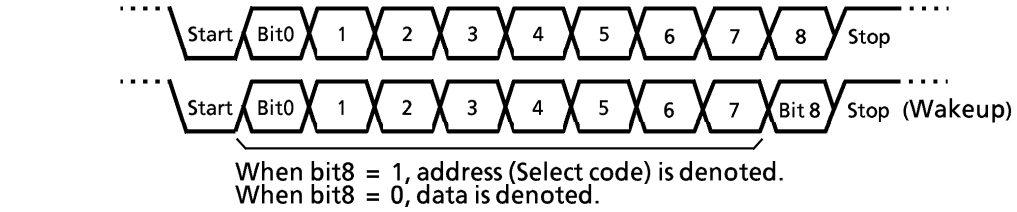


Figure 3.9.1 Data Formats

3.9.1 Block Diagram

Figure 3.9.2 shows the block diagram of the serial channel 0.

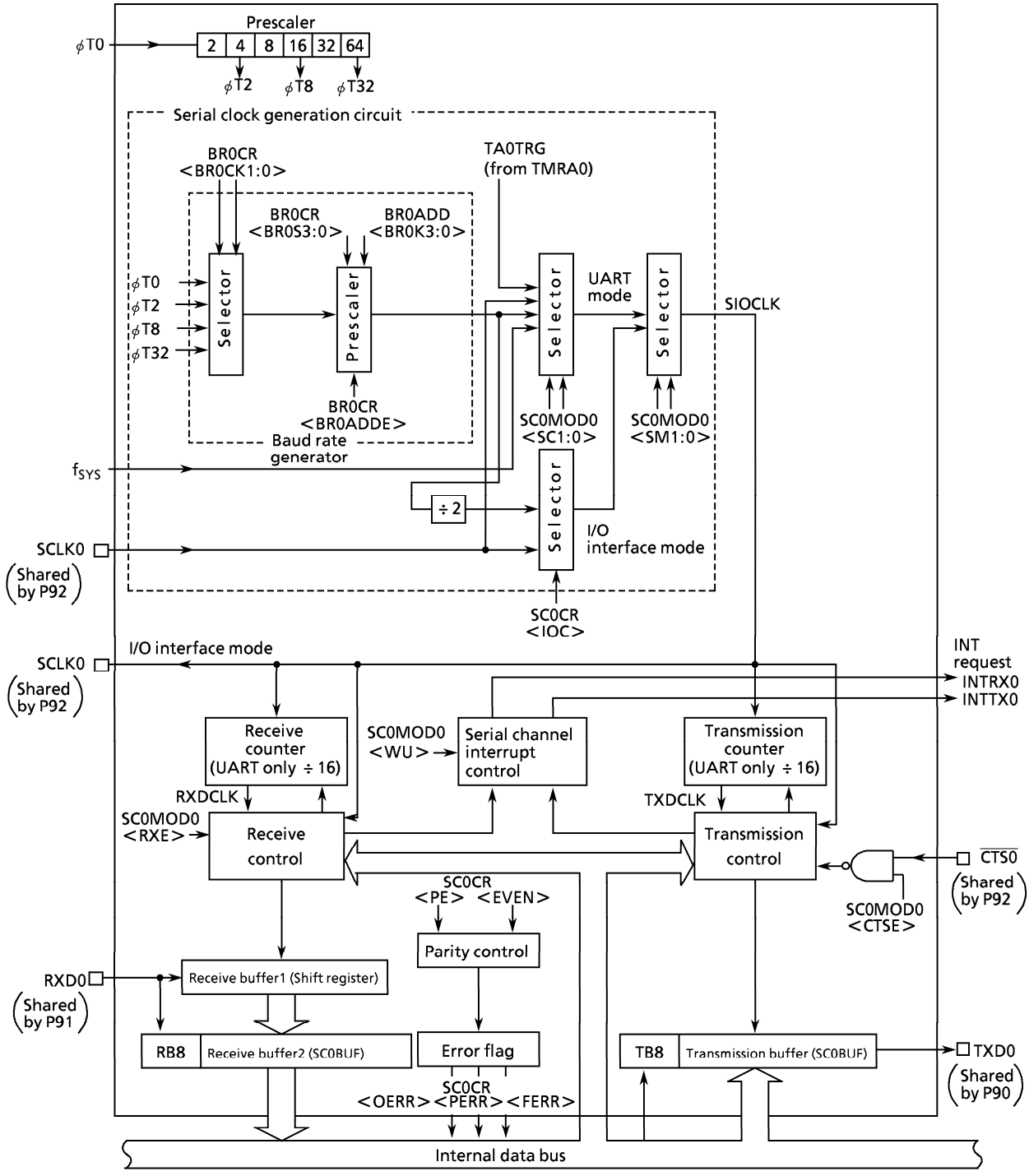


Figure 3.9.2 Block Diagram of the Serial Channel 0

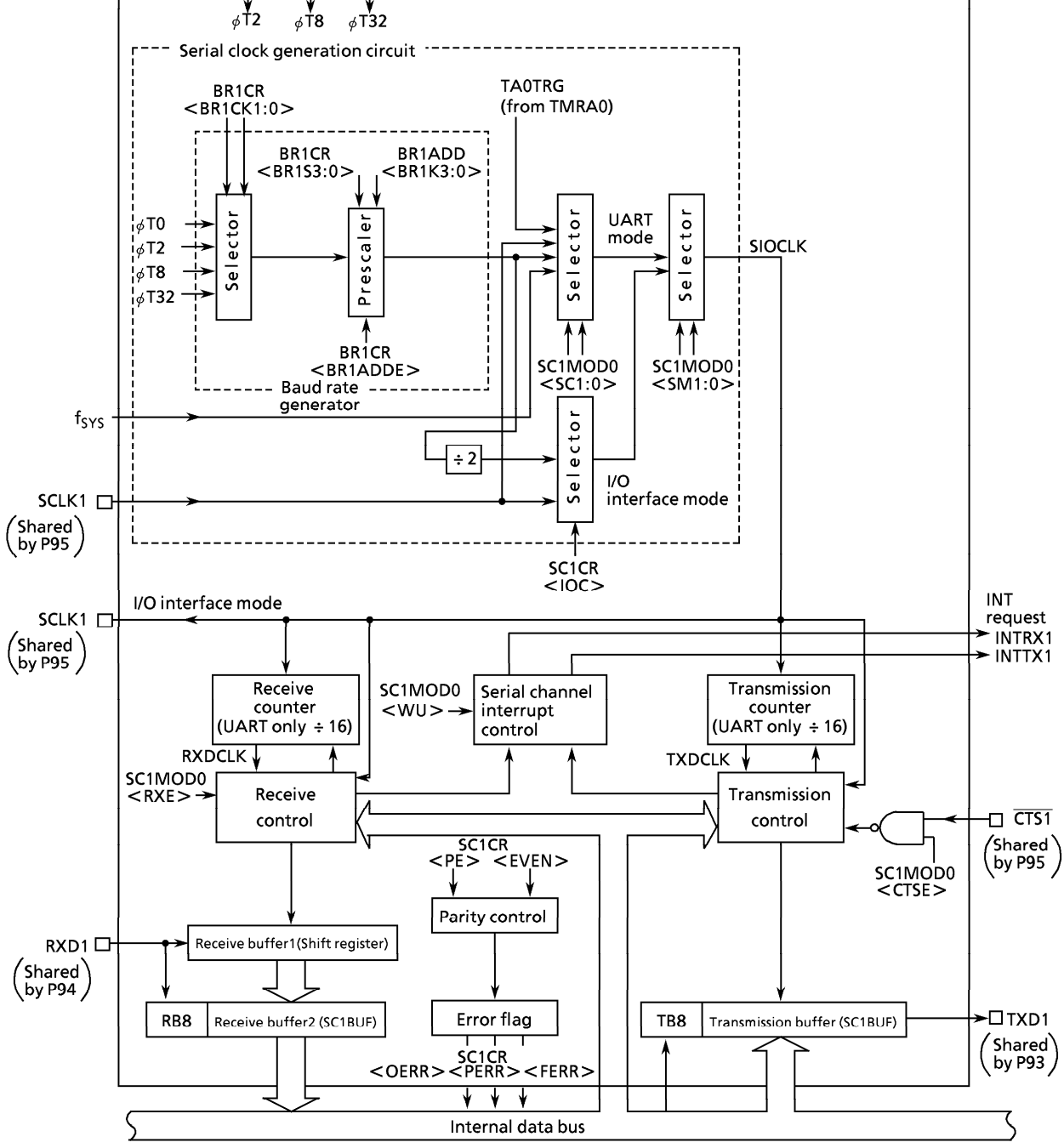


Figure 3.9.3 Block Diagram of the Serial Channel 1

3.9.2 Operation for Each Circuit

(1) Prescaler, Prescaler clock select

There is 6-bit prescaler for waking serial clock. The clock selected by prescaler clock selection register SYSCR<PRCK1:0> is divided by 4 and input to this prescaler as ϕT0. This prescaler can be run by selected baud rate generator for waking serial clock.
Table 3.9.2 shows prescaler clock resolution into the baud rate generator.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Output Clock Resolution			
			ϕT0	ϕT2	ϕT8	ϕT32
1 (fs)	00 (f_{FPH})	XXX	fs/2^2	fs/2^4	fs/2^6	fs/2^8
0 (fc)		000 (fc)	fc/2^2	fc/2^4	fc/2^6	fc/2^8
		001 (fc/2)	fc/2^3	fc/2^5	fc/2^7	fc/2^9
		010 (fc/4)	fc/2^4	fc/2^6	fc/2^8	fc/2^{10}
		011 (fc/8)	fc/2^5	fc/2^7	fc/2^9	fc/2^{11}
		100 (fc/16)	fc/2^6	fc/2^8	fc/2^{10}	fc/2^{12}
	10 (fc/16 clock)	XXX	—	fc/2^8	fc/2^{10}	fc/2^{12}

XXX: Don't care, – : Can not use

The baud rate generator selects between 4 clock inputs : ϕT0, ϕT2, ϕT8, and ϕT32 among the prescaler outputs.

(2) Baud rate generator

The baud rate generator is a circuit that generates transmission and receiving clocks to determine the transfer rate of the serial channel.
The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8, or ϕT32, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate generator control register BR0CR<BR0CK1:0>.
The baud rate generator includes a frequency divider, which divides frequency by 1, N+(16−K)/16 and 16 values to determine the transfer rate.
The transfer rate is determined by setting BR0CR<BR0ADDE, BR0S3:0> and BR0ADD<BR0K3:0>.

- UART mode

  (1) BR0CR<BR0ADDE> = 0
  Setting BR0ADD<BR0K3:0> is ignored. The baud rate generator divides the selected prescaler clock by N which is set to BR0CK<BR0S3:0>. (N = 1, 2, 3 ··· 16)

  (2) BR0CR<BR0ADDE> = 1
  N+(16−K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N+(16−K)/16 according to N set to BR0CR<BR0S3 to 0> (N = 2, 3 ··· 15) and K set to BR0ADD<BR0K3:0> (K = 1, 2, 3 ··· 15)

  Note : At N = 1 or 16, N+(16−K)/16 division function is disabled. Set BR0CR<BR0ADDE> to 0.

- I/O interface mode

  N+(16−K)/16 division function is not available in I/O interface mode. Set BR0CR<BR0ADDE> to 0 before dividing by N.

How to calculate a transfer rate when the baud rate generator is used is explained below.

- UART mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divider of baud rate generator}} \div 16$$

- I/O interface mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divider of baud rate generator}} \div 2$$

- Integer divider (N divider)

  For example, when the source clock (fc) is 12.288 MHz, the input clock is ϕT2 (fc/16), and frequency divider is N(BR0CR<BR0S3:0>)=5, BR0CR<BR0ADDE> =0, the baud rate in UART mode becomes as follows:

  ※ Clock condition
  - System clock: High frequency (fc)
  - Clock gear: 1 (fc)
  - Prescaler clock: System clock

$$\text{Baud rate} = \frac{\text{fc}/16}{5} \div 16$$
$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

  Note: +(16−K)/16 division function is disabled, and setting BR0ADD<BR0K3:0> is invalid.

- N+(16−K)/16 divider (only in UART mode)

  Accordingly, when source clock (fc) is 4.8 MHz, the input clock is ϕT0 , and frequency divider is N (BR0CR<BR0S3:0>)=7, K (BR0ADD<BR0K3:0>)=3, BR0CR<BR0ADDE>=1, the baud rate in UART mode becomes as follows:

  ※ Clock condition
  - System clock: High frequency (fc)
  - Clock gear: 1 (fc)
  - Prescaler clock: System clock

$$\text{Baud rate} = \frac{\text{fc}/4}{7+(16-3)/16} \div 16$$
$$= 4.8 \times 10^6 \div 4 \div (7+13/16) \div 16 = 9600 \text{ (bps)}$$

  Table 3.9.3, 3.9.4 show an example of the transfer rate in UART mode.
  Additionally, the external clock input is available in the serial clock. (Serial channel 0, 1). How to calculate a baud rate is explained below.

- UART mode

  Baud rate = External clock input ÷16
  It is necessary to satisfy (External clock input cycle)>=4/fc

- I/O interface mode

  Baud rate = External clock input
  It is necessary to satisfy (External clock input cycle) >=16/fc

Table 3.9.3 Selection of Transfer Rate
(when baud rate generator is used and BR0CR<BR0ADDE> = 0)

Unit (kbps)

fc [MHz]	Input Clock / Frequency Divider	φT0	φT2	φT8	φT32
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.745600	2	115.200			
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300

Note 1: Transfer rates in I/O interface mode are 8 times faster than the values given in the above table.
Note 2: This table is calculated when fc is selected as a system clock, the clock gear is set for fc, and the system clock as the prescaler clock input.

Table 3.9.4 Selection of Transfer Rate
(when TMRA0 with input clock φT1 is used)

Unit (kbps)

TA0REG0 \ fc	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

How to calculate the transfer rate (when TMRA0 is used):

$$\text{Transfer rate} = \frac{\text{The clock frequency selected by the register SYSCR0<PRCK1:0>}}{\text{TA0REG} \times \underline{8} \times 16}$$

(when TMRA0 (input clock φT1) is used)

Note 1: TMRA0 match detect signal cannot be used as the transfer clock in I/O interface mode.
Note 2: This table is calculated when fc is selected as a system clock, the clock gear is set for fc, and system clock as the prescaler clock input.

(3) Serial clock generation circuit

This circuit generates the basic clock for transmitting and receiving data.

- I/O interface mode

  When in SCLK output mode with the setting of SC0CR<IOC> = 0, the basic clock will be generated by dividing the output of the baud rate generator by 2 as described before.
  When in SCLK input mode with the setting of SC0CR<IOC> = 1, the rising edge or falling edge will be detected according to the setting of the SC0CR<SCLKS> register to generate the basic clock.

- UART mode

  The setting of SC0MOD0<SC1:0>, will select between the baud rate generator clock, internal system clock f_{SYS}, the match detect signal from timer TMRA0 or the external clock (SCLK0) to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART mode and counts up according to the SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 bit of data, and the data bit is sampled three times at the 7th, 8th and 9th clock (This is the case of except when SIOCLK is set to f_{SYS}.).
With these three samples, the received data bit is evaluated by the majority rule.
For example, if the sampled data bit is 1, 0 and 1 at 7th, 8th and 9th clock respectively, the received data is evaluated as 1. The sampled data 0, 0 and 1 is evaluated such that the received data bit is determined to be 0.

(5) Receiving control

- I/O interface mode

  When in SCLK output mode with the setting of SC0CR<IOC> = 0, the RXD0 signal will be sampled at the rising edge of the shift clock which is output to the SCLK0 pin.
  When in SCLK input mode with the setting of SC0CR<IOC> = 1, the RXD0 signal will be sampled at the rising edge or falling edge of the SCLK0 input according to the setting of the SC0CR<SCLKS> register.

- UART mode

  The receiving control block has a circuit for detecting the start bit by the rule of majority. When two or more 0 are detected during the 3 samples, it is recognized as start bit and the receiving operation is started.
  The data being received is also evaluated by the majority rule.

(6) Receiving buffer

To prevent an overrun error, the receiving buffer has a double buffer structure.
Received data is stored bit by bit in receiving buffer 1 (Shift register type). When 7 bits or 8 bits of data are stored in receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC0BUF) generating an interrupt INTRX0. The CPU reads only receiving buffer 2 (SC0BUF). Even before the CPU reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR<RB8> are still preserved.
The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SC0CR<RB8>.
When in 9-bit UART mode, the wake-up function of the slave controller is enabled by setting SC0MOD0<WU> to 1, and interrupt INTRX0 occurs only when SC0CR<RB8> is set to 1.

(7) Transmission counter

The transmission counter is a 4-bit binary counter which is used in UART mode and, like a receiving counter, counts by the SIOCLK clock which generates TXDCLK every 16 clock pulses.

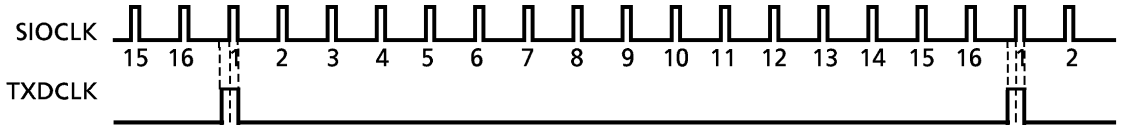


Figure 3.9.4 Generation of Transmission Clock

(8) Transmission controller

- I/O interface mode

  In SCLK output mode with the setting of SC0CR<IOC> = 0, the data in the transmission buffer is output bit by bit to the TXD0 pin at the rising edge of the shift clock which is output from the SCLK0 pin.
  In SCLK input mode with the setting of SC0CR<IOC> = 1, the data in the transmission buffer is output bit by bit to the TXD0 pin at the rising edge or falling edge of the SCLK0 input according to the setting of the SC0CR<SCLKS> register.

- UART mode

  When transmission data is written to the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TXDCLK, generating a transmission shift clock TXDSFT.

Handshake function

Serial channel 0, 1 have a $\overline{CTS}$ pin. Using this pin, data can be sent in units of one frame; thus, overrun errors can be avoided. The handshake function is enabled/ disabled by SC0MOD0<CTSE>.
When the $\overline{CTS0}$ pin goes high, after completion of the current data send, data send is halted until the $\overline{CTS0}$ pin goes low again. However the INTTX0 Interrupt is generated, it requests the next data send to the CPU. The next data is written in the transmission buffer and data send is halted.
Though there is no $\overline{RTS}$ pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{RTS}$ function. The $\overline{RTS}$ should be output high to request send data halt after data receive is completed by software in the RXD interrupt routine.

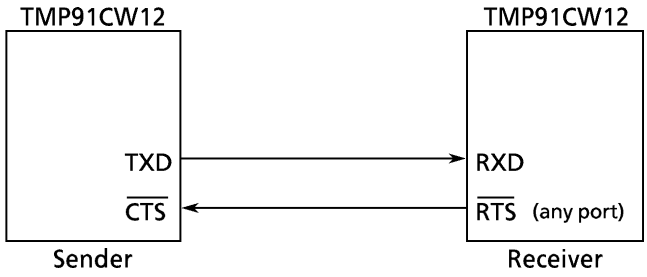
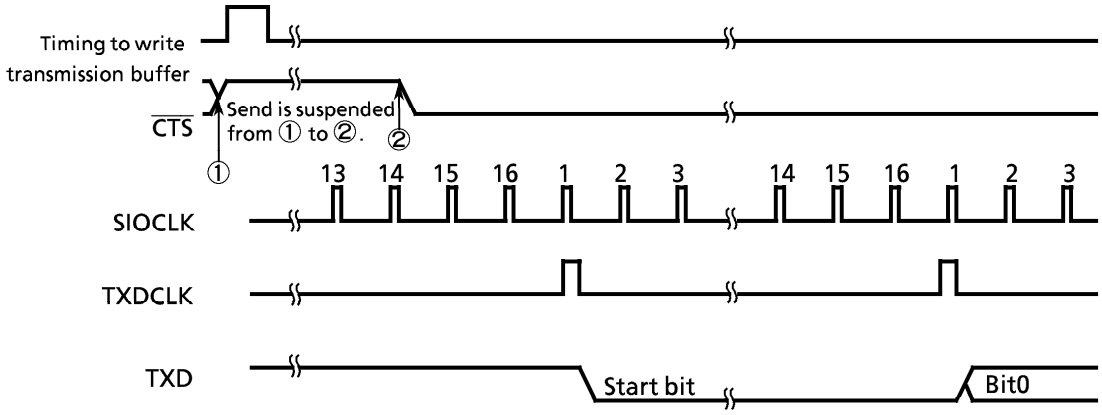


Figure 3.9.5 Handshake Function



Note 1: If the $\overline{CTS}$ signal rises during transmission, the next data is not sent after the completion of the current transmission.
Note 2: Transmission starts at the first TXDCLK clock falling edge after the $\overline{CTS}$ signal falls.

Figure 3.9.6 Timing of $\overline{CTS}$ (Clear to send)

(9) Transmission buffer

The transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0 interrupt.

(10) Parity control circuit

When the serial channel control register SC0CR<PE>is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART modes. With SC0CR <EVEN> register, even or odd parity can be selected.
For transmission, parity is automatically generated according to the data written in the transmission buffer SC0BUF. The data is transmitted after the parity bit is stored in SC0BUF<TB7> when in 7-bit UART mode or in SC0MOD0<TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before the transmission data is written to the transmission buffer.
For receiving, data are shifted in the receiving buffer 1, the parity is added after the data is transferred to receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> when in 7-bit UART mode and with SC0CR<RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs and SC0CR<PERR> flag is set.

(11) Error flag

Three error flags are provided to increase the reliability of receiving data.

1. Overrun error <OERR>

If all bits of the next data are received in receiving buffer 1 while valid data is stored in receiving buffer 2 (SC0BUF), an overrun error will occur.

The below is a processing example of when overrun-error is occurred.
(INTRX routine)
1) Read received-buffer
2) Read error-flag
3) if <OERR> = 1
then
a) Disable receiving (Write 0 to <RXE>)
b) Wait for terminating current frame
c) Read received-buffer
d) Read error-flag
e) Enable receiving (Write 1 to <RXE>)
f) Request to resend
4) Process other job

2. Parity error <PERR>

The parity generated for the data shifted in receiving buffer 2 (SC0BUF) is compared with the parity bit received from RXD pin. If they are not equal, a parity error occurs.

- Limitation to use parity-error flag (only for TMP91CW12 and TMP91PW12)

The parity-error flag: SC0CR<PERR> can not be used in the case of UART with parity and full-duplex usage.
When the above usage is needed, execute error-judgement by using parity flag of CPU based on received data like following example.

(1) The case of "8 bits + Odd-parity"

```
ld      wa, (sc0buf)      ; Read received data and parity-bit
ld      b, w              ; Copy other error flag
and     wa, 80ffh         ; Mask except other flag and
                          ; Set parity flag
jr      pe, parity_err    ; Jump to error routine if parity flag = 1
```

(2) The case of "7 bit + Odd-parity"

```
ld    a, (sc0buf)      ; Read received data and parity-bit
and   a, 0ffh          ; Set parity flag
jr    pe, parity_err   ; Jump to error routine if parity flag = 1
```

3. Framing error <FERR>

The stop bit of received data is sampled three times around the center. If the majority is 0, a framing error occurs.

(12) Generating timing

1) UART mode

Receiving

Mode	9 Bits (Note)	8 Bits + Parity (Note)	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit
Framing error timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error timing	—	Center of last bit (Parity bit)	Center of last bit (Parity bit)
Overrun error timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit

Note: In 9 bits and 8 bits + party modes, interrupts coincide with the ninth bit pulse. Thus, when servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transferred) to allow checking for a framing error.

Transmitting

Mode	9 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Just before stop bit is transmitted.	Just before stop bit is transmitted.	Just before stop bit is transmitted.

2) I/O interface

Transmission interrupt timing	SCLK output mode	Immediately after rise of last SCLK signal. (See Figure 3.9.19.)
	SCLK input mode	Immediately after rise of last SCLK signal (Rising mode), or immediately after fall in falling mode. (See Figure 3.9.20.)
Receiving interrupt timing	SCLK output mode	Timing used to transfer received data to data receive buffer 2 (SC0BUF) (that is, immediately after last SCLK). (See Figure 3.9.21.)
	SCLK input mode	Timing used to transfer received data to data receive buffer 2 (SC0BUF) (that is, immediately after last SCLK). (See Figure 3.9.22.)

3.9.3 SFR

SC0MOD0 (0202H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Hand shake 0: CTS disable 1: CTS enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TMRA0 trigger 01: baud rate generator 10: Internal clock f_{SYS} 11: External clock (SCLK0 input)	

Serial transmission clock source (UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK0 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC0CR).

Serial transmission mode

00	I/O Interface mode	
01	UART	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving function

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS}$ pin)

0	Disable (Always transferable)
1	Enable

Transmission data bit8

Figure 3.9.7 Serial Mode Control Register (Channel 0, SC0MOD0)

SC1MOD0 (020AH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Hand shake 0: CTS disable 1: CTS enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: Reserved 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock (SCLK1 input)	

Serial transmission clock source (for UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK1 input)

Serial transmission mode

00	I/O Interface mode	
01	UART	7-bit length
10		8-bit length
11		9-bit length

Wakeup Function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving control

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS}$ pin)

0	Disable (Always transferable)
1	Enable

Transmission data bit8

Figure 3.9.8 Serial Mode Control Register (Channel 1, SC1MOD0)

SC0CR (0201H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK0 (↑) 1: SCLK0 (↓)	0:Baud rate generator 1: SCLK0 Pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generater
1	SCLK0 Pin input

Edge selection for SCLK pin

0	Transmits and receives data at rise edge of SCLK0
1	Transmits and receives data at fall edge of SCLK0

Framing error flag
Parity error flag
Overrun error flag
→ Cleared to 0 when read.

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receving data bit8

Note: As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.9.9 Serial Control Register (Channel 0, SC0CR)

SC1CR (0209H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK1 (↑) 1: SCLK1 (↓)	0:Baud rate generator 1: SCLK1 Pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generater
1	SCLK1 Pin input

Edge selection for SCLK pin

0	Transmits and receives data at rise edge of SCLK1
1	Transmits and receives data at falling edge of SCLK1

Framing error flag
Parity error flag
Overrun error flag
→ Cleared to 0 when read.

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receving data bit8

Note: As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.9.10 Serial Control Register (Channel 1, SC1CR)

BR0CR (0203H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	(Note) Always write 0	+ (16 – K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the divided frequency			

+ (16 – K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR0ADD (0204H)

	7	6	5	4	3	2	1	0
Bit symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor K (Divided by N = (16 – K)/16)			

Sets baud rate generator frequency divisor

BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 – K)/16	

Note: When N + (16 – K)/16 division function is used or not

N	UART Mode	I/O Mode
2 to 15	○	×
1, 16	×	×

Set BR0CR<BR0ADDE> to 1 after setting K (K = 1 to 15) to BR0ADD<BR0K3:0> when + (16 – K)/16 division function is used.

Figure 3.9.11 Baud Rate Generator Control (Channel 0, BR0CR, BR0ADD)

BR1CR (020BH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	(Note) Always write 0	+ (16 – K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the divided frequency			

+ (16 – K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR1ADD (020CH)

	7	6	5	4	3	2	1	0
Bit symbol					BR1K3	BR1K2	BR1K1	BR1K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor K (divided by N = (16 – K)/16)			

Sets baud rate generator frequency divisor

BR1ADD <BR1K3:0> \ BR1CR <BR1S3:0>	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 – K) /16	

Note: When N + (16 – K)/16 division function is used or not

N	UART Mode	I/O Mode
2 to 15	○	×
1, 16	×	×

Set BR1CR<BR1ADDE> to 1 after setting K (K = 1 to 15) to BR1ADD<BR1K3:0> when + (16 – K)/16 division function is used.

Figure 3.9.12 Baud Rate Generator Control (Channel 1, BR1CR, BR1ADD)

SC0BUF (0200H)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read-modify-write for SC0BUF.

Figure 3.9.13 Serial Transmission/Receiving Buffer Registers (Channel 0, SC0BUF)

SC0MOD1 (0205H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S0	FDPX0						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.9.14 Serial Mode Control Register 1 (Channel 0, SC0MOD1)

SC1BUF (0208H)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note : Prohibit read-modify-write for SC1BUF.

Figure 3.9.15 Serial Transmission/Receiving Buffer Registers (Channel 1, SC1BUF)

SC1MOD1 (020DH)

	7	6	5	4	3	2	1	0
Bit symbol	I2S1	FDPX1						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.9.16 Serial Mode Control Register 1 (Channel 1, SC1MOD1)

3.9.4 Operation for Each Mode

(1) Mode 0 (I/O interface mode)

This mode is used to increase the number I/O pins for transmitting or receiving data to or from an external shifter register.
This mode includes the SCLK output mode to output synchronous clock SCLK and SCLK input mode to input external synchronous clock SCLK.

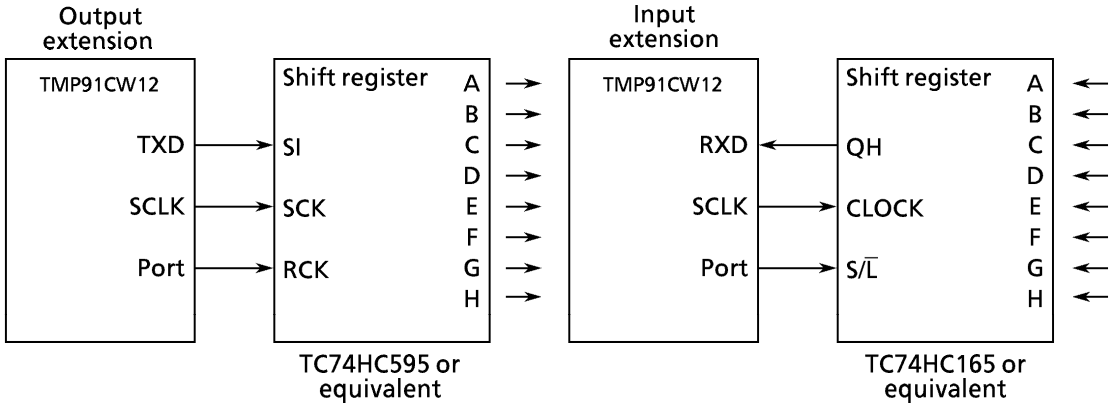


Figure 3.9.17 Example of SCLK Output Mode Connection

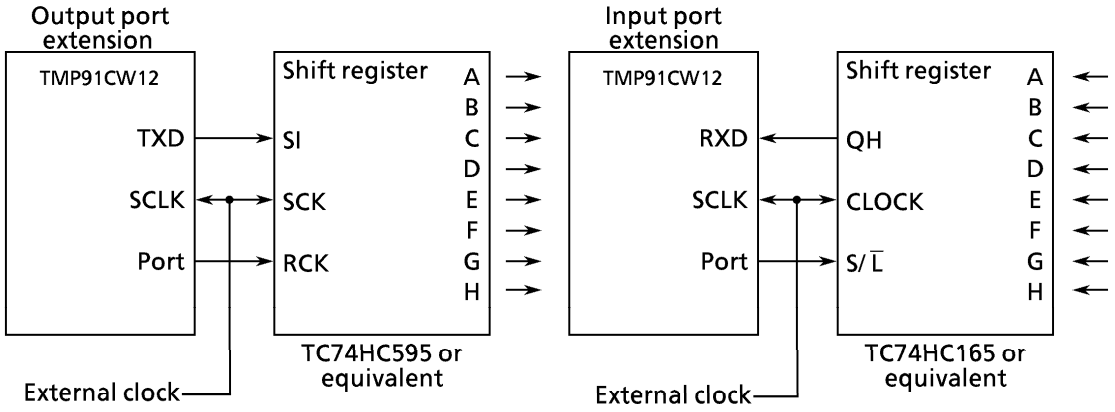


Figure 3.9.18 Example of SCLK Input Mode Connection

① Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TXD0 pin and SCLK0 pin respectively, each time the CPU writes data to the transmission buffer. When all data is output, INTES0<ITX0C> will be set to generate the INTTX0 interrupt.

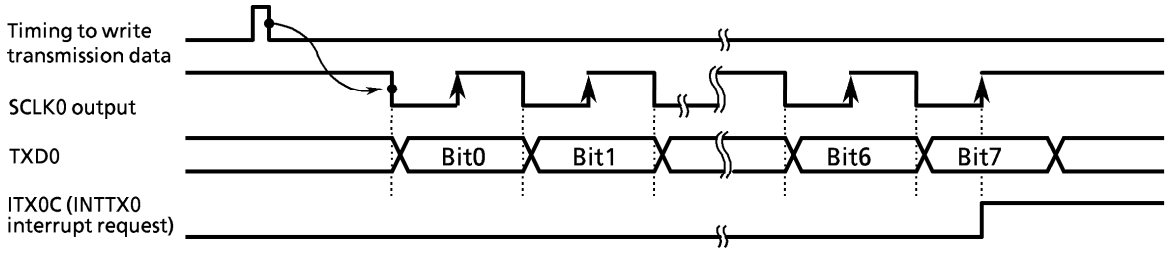


Figure 3.9.19 Transmitting Operation in I/O Interface Mode (SCLK0 output mode) (Channel 0)

In SCLK input mode, 8-bit data is output from TXD0 pin when SCLK0 input becomes active after data are written to the transmission buffer by CPU.
When all data is output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

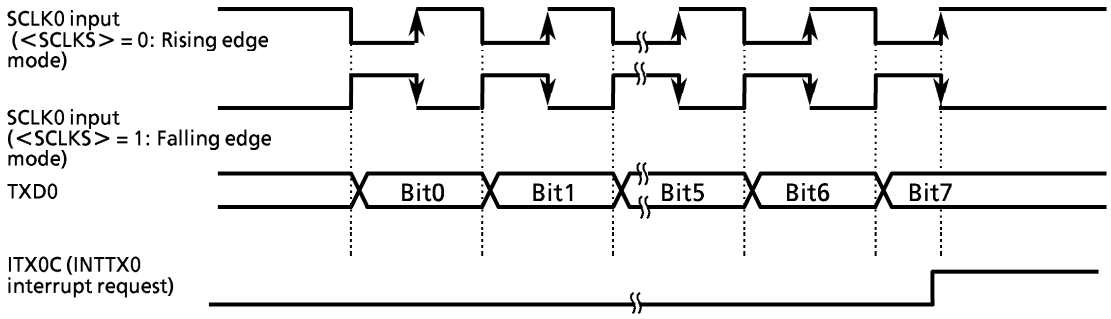


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 input mode) (Channel 0)

② Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK0 pin and the data is shifted to receiving buffer 1. This starts when the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.
The outputting for the first SCLK0 starts by setting SC0MOD0<RXE>to 1.

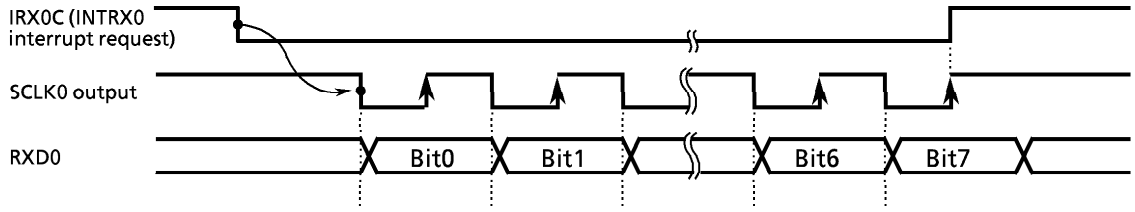


Figure 3.9.21 Receiving Operation in I/O Interface Mode (SCLK0 output mode) (Channel 0)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.

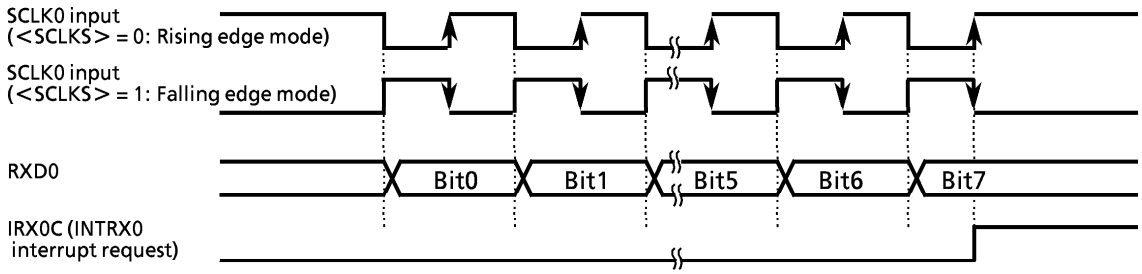


Figure 3.9.22 Receiving Operation in I/O Interface Mode (SCLK0 input mode) (Channel 0)

Note: For data receiving, the system must be placed in the receive enable state (SCMOD0<RXE> = 1).

③ Transmission and receiving (Full duplex mode)

When the full duplex mode is used, set the level of receive interrupt to 0 and set enable the level of transmit interrupt. In the transmit interrupt program, read the receiving buffer before setting the next transmit data.
The example is following.

(Setting example)

Channel 0, SCLK outputting
Baud rate = 9600 bps
at fc = 14.7456 MHz
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Main routine setting

	7 6 5 4 3 2 1 0	
INTES0	X 0 0 1 X 0 0 0	Set the level of INTTX0 to 1 and set the level of INTRX0 to 0.
P9CR	- - - - - - 0 1	Set to port P90 (TXD0), P91 (RXD0), P92 (SCLK0).
P9FC	- - - - - 1 - 1	
SC0MOD0	- - - - 0 0 - -	Set to I/O interface mode.
SC0MOD1	1 1 X X X X X X	Set to full duplex mode.
SC0CR	- - - - - - 0 -	SCLK_out, transmit on negative-edge, receive on positive-edge.
BR0CR	0 0 1 1 0 0 1 1	Baud rate = 9600 bps.
SC0MOD0	- - 1 - - - - -	Enable receiving.
SC0BUF	* * * * * * * *	Set the transmit data and start.

INTTX0 interrupt routine

Acc	SC0BUF	Read the receiving buffer.
SC0BUF	* * * * * * * *	Set the next transmit data.

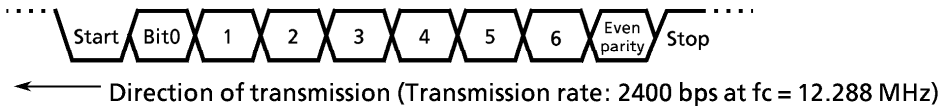
X: Don't care, –: No change

(2) Mode 1 (7-bit UART mode)

The 7-bit mode can be set by setting serial channel mode register SC0MOD0<SM1:0> to 01.
In this mode, a parity bit can be added, and the addition of the parity bit can be enabled or disabled by serial channel control register SC0CR<PE>, and even parity or odd parity is selected by SC0CR <EVEN> when <PE> is set to 1 (enable).

(Setting example)

When transmitting data with the following format, the control registers should be set as described below. Channel 0 is explained here.



※ Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

	7 6 5 4 3 2 1 0	
P9CR	← - - - - - - - 1	} Select P90 as the TXD0 pin.
P9FC	← - - - - - - - 1	
SC0MOD0	← - - - - 0 1 0 1	Set 7-bit UART mode.
SC0CR	← - 1 1 - - - - -	Add even parity.
BR0CR	← 0 0 1 0 0 1 0 1	Set transfer rate at 2400 bps.
INTES0	← X 1 0 0 - - - -	Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF	← * * * * * * * *	Set data for transmission.

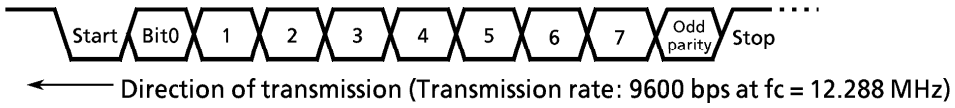
X: Don't care, –: No change

(3) Mode 2 (8-bit UART mode)

The 8-bit UART mode can be specified by setting SC0MOD0<SM1:0> to 10. In this mode, the parity bit can be added (The addition of a parity bit is enabled or disabled by SC0CR<PE>) and even parity or odd parity is selected by SC0CR<EVEN> when <PE> is set to 1 (enable).

(Setting example)

When receiving data with the following format, the control register should be set as described below.



※ Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

Main setting

```
                7 6 5 4 3 2 1 0
P9CR      ← - - - - - - 0 -      Select P91 (RXD0) as the input pin.
SC0MOD0   ← - - 1 - 1 0 0 1      Enable receiving in 8-bit UART mode.
SC0CR     ← - 0 1 X X X 0 0      Add odd parity.
BR0CR     ← 0 0 0 1 0 1 0 1      Set transfer rate at 9600 bps.
INTES0    ← X - - - X 1 0 0      Enable INTRX0 interrupt and set interrupt level 4.
```

Interrupt processing

```
Acc ← SC0CR AND 00011100   }  Check for error.
if Acc ≠ 0 then ERROR      }
Acc ← SC0BUF                  Read the received data.
```

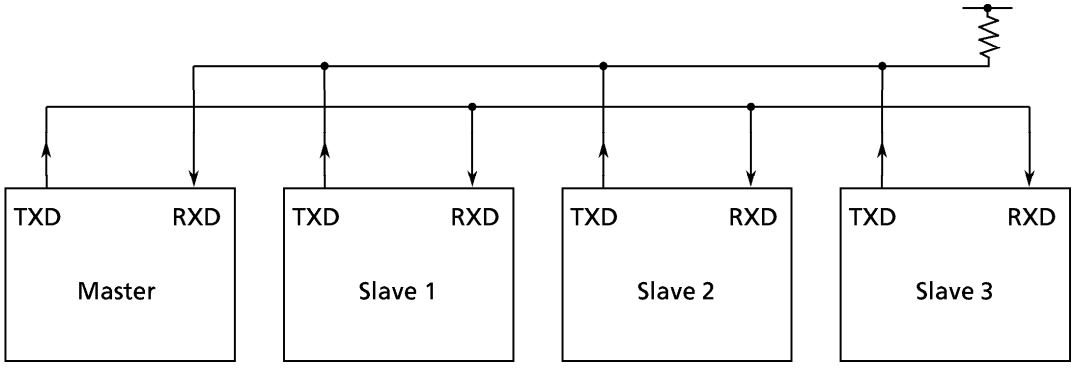
X: Don't care, —: No change

(4) Mode 3 (9-bit UART mode)

9-bit UART mode can be specified by setting SC0MOD0<SM1:0> to 11. In this mode, parity bit cannot be added.
For transmission, the MSB (9th bit) is written in SC0MOD0<TB8>. For receiving it is stored in SC0CR<RB8>. For writing and reading of the buffer, the MSB is read or written first then the rest of the data from SC0BUF.

Wakeup function

In 9-bit UART mode, the wakeup function of slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only when<RB8> = 1.



Note: TXD pin of the slave controllers must be in open drain output mode.

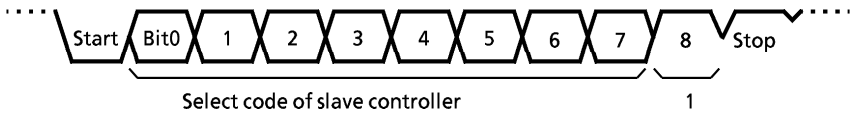
Figure 3.9.23 Serial Link Using Wakeup Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

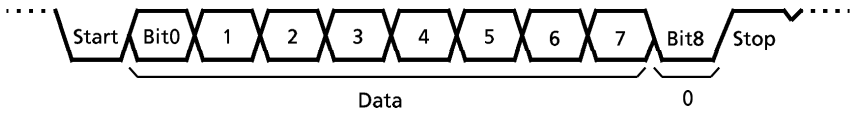
② Set SC0MOD0＜WU＞ bit of each slave controller to 1 to enable data receiving.

③ The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (Bit8)＜TB8＞ is set to 1.



④ Each slave controller receives the above frame, and clears WU bit to 0 if the above select code matches its own select code.

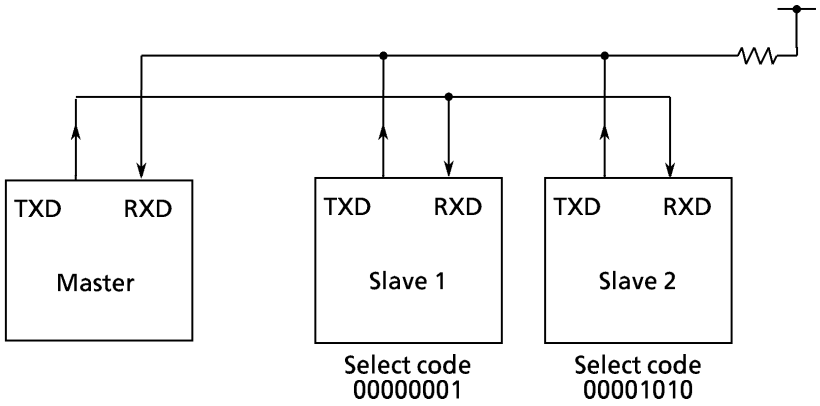
⑤ The master controller transmits data to the specified slave controller whose SC0MOD0＜WU＞ bit is cleared to 0. The MSB (Bit8)＜TB8＞ is cleared to 0.



⑥ The other slave controllers (with the ＜WU＞ bit remaining at 1) ignore the receiving data because their MSBs (Bit8 or ＜RB8＞) are set to 0 to disable the interrupt INTRX0.
The slave controllers (WU＝0) can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

(Setting example)

To link two slave controllers serially with the master controller, and use the internal clock f_{SYS} as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same way, channel 0 is used for the purposes of explanation.

- Setting the master controller

```
Main
P9CR    ← - - - - - - - 0 1 }
P9FC    ← X X - X - - X 1 }   Select P90 as TXD0 pin and P91 as RXD0 pin.
INTES0  ← X 1 0 0 X 1 0 1     Enable INTTX0 and set the interrupt level 4.
                              Enable INTRX0 and set the interrupt level 5.
SC0MOD0 ← 1 0 1 0 1 1 1 0     Set fSYS as the transmission clock in 9-bit UART mode.
SC0BUF  ← 0 0 0 0 0 0 0 1     Set the select code for slave controller 1.
INTTX0 interrupt
SC0MOD0 ← 0 - - - - - - -     Sets TB8 to 0.
SC0BUF  ← * * * * * * * *     Set data for transmission.
```

- Setting the slave controller

```
Main
P9CR    ← - - - - - - - 0 1 }
P9FC    ← X X - X - - X 1 }   Select P91 as RXD0 pin and P90 as TXD0 pin (Open-drain
ODE     ← X X X X X X - 1 }   output).
INTES0  ← X 1 0 1 X 1 1 0     Enable INTRX0 and INTTX0.
SC0MOD0 ← 0 0 1 1 1 1 1 0     Set <WU> to 1 in the 9-bit UART transmission mode with
                              transfer clock fSYS.
INTRX0 interrupt
Acc ← SC0BUF
if Acc = Select code
Then SC0MOD0 ← - - - 0 - - - -   Clear <WU> to 0.
```

3.9.5 Support for IrDA

The SIO0 has a function which supports the IrDA 1.0 specification of the infrared data communication. Figure 3.9.24 shows the block diagram.

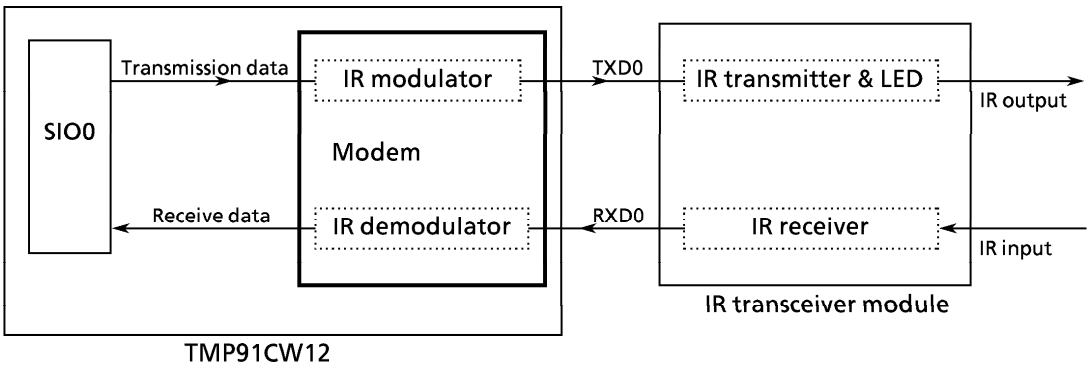


Figure 3.9.24 Block Diagram

(1) Modulation for the transmit data

When the transmit data is 0, the modem outputs 1 to TXD0 pin with either 3/16 or 1/16 times for width of baud-rate. The pulse width is selected by SIRCR<PLSEL>. When the transmit data is 1, the modem outputs 0.

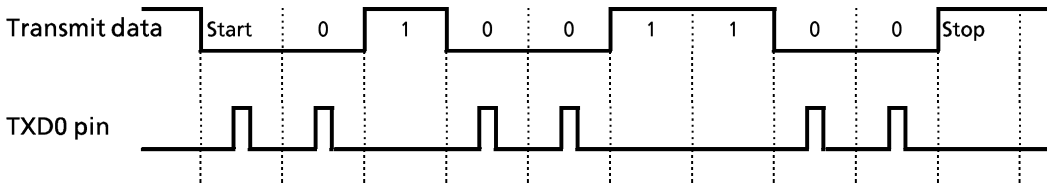


Figure 3.9.25 The Example for Transmission

(2) Demodulation for the receive data

When the receive data is the effective width of 1 pulse, the modem outputs 0 to SIO0. Otherwise the modem outputs 1 to SIO0. The effective pulse width is selected by SIRCR<SIRWD3:0>.

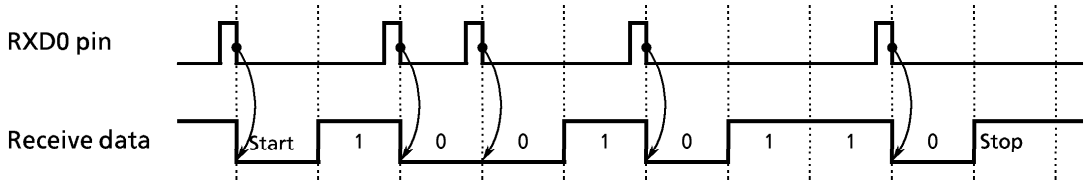


Figure 3.9.26 The Example Receiving

(3) Data format

The data format is fixed following

- Data length: 8 bits
- Parity bit: None
- Stop bit: 1 bit

(4) SFR

Figure 3.9.27 shows the control register SIRCR. Set the data SIRCR during SIO0 is stopping. The setting example is following

```
1) SIO setting              ;  Set SIO for UART mode.
        ↓
2) LD   (SIRCR), 07H        ;  Set to 3/16 pulse width for receive data.
3) LD   (SIRCR), 37H        ;  TXEN, RXEN enable the transmission and receiving.
        ↓
4) Start transmission       ;  The modem operates by following action.
   and receiving for SIO0       • SIO0 starts transmission.
                                • IR receiver starts receiving.
```

(5) Notes

1. Baud rate for IrDA
   When IrDA is operated, set 01 to SC0MOD0<SC1:0> to generate baud rate.
   The setting except above (TA0TRG, f_{SYS} and SCLK0 input) can not be used.
2. The pulse width for transmittion
   The IrDA 1.0 specification is defined in Table 3.9.5.

Table 3.9.5 Baud Rate and Pulse Width Specifications

Baud Rate	Modulation	Rate Tolerance (% of rate)	Pulse Width (Minimum)	Pulse Width (Typical)	Pulse Width (Maximum)
2.4 kbps	RZI	±0.87	1.41 μs	78.13 μs	88.55 μs
9.6 kbps	RZI	±0.87	1.41 μs	19.53 μs	22.13 μs
19.2 kbps	RZI	±0.87	1.41 μs	9.77 μs	11.07 μs
38.4 kbps	RZI	±0.87	1.41 μs	4.88 μs	5.96 μs
57.6 kbps	RZI	±0.87	1.41 μs	3.26 μs	4.34 μs
115.2 kbps	RZI	±0.87	1.41 μs	1.63 μs	2.23 μs

The pulse width is defined either baud rate T × 3/16 or 1.6 μs (1.6 μs is equal to 3/16 pulse width when baud rate is 115.2 kbps).
The TMP91CW12 has the function selects the pulse width of transmission either 3/16 or 1/16. But 1/16 pulse width can be selected when the baud rate is equal or less than 38.4 kbps.
As the same reason, +(16−K)/16 division function in the baud rate generator of SIO0 can not be used to generate 115.2 kbps baud rate except a special condition which is explained (6) Using IrDA 115.2 kbps with USB .
Also when the 38.4 kbps and 1/16 pulse width, +(16−K)/16 division function can not be used.

Table 3.9.6 Baud Rate and Pulse Width With (16 – k)/16 Division Function

Pulse Width	Baud Rate 115.2 kbps	57.6 kbps	38.4 kbps	19.2 kbps	9.6 kbps	2.4 kbps
T × 3/16	×	○	○	○	○	○
T × 1/16	–	–	×	○	○	○

○: Can be used (16 – K)/16 division function
×: Can not be used (16 – K)/16 division function
–: Can not be set to 1/16 pulse width

SIRCR (0207H)

	7	6	5	4	3	2	1	0
Bit symbol	PLSEL	–	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
Read/Write	R/W							
Reset	0	0	0	0	0	0	0	0
Function	Select transmit pulse width 0: 3/16 1: 1/16	Write 0	Transmit 0: Disable 1: Enable	Receive 0: Disable 1: Enable	Select receive pulse width Set effective pulse width for equal or more than 2x × x(value + 1) Can be set: 1 to 14 Can not be set: 0, 15			

Select receive pulse width
Formula: Effective pulse width ≧ 2x × (value + 1)
$x = 1/f_{FPH}$

0000	Can not be set
0001	Equal or more than 4x + 100 ns
to	
1110	Equal or more than 30x + 100 ns
1111	Can not be set

Receive operation

0	Disable
1	Enable

Transmit operation

0	Disable
1	Enable

Select transmit pulse width

0	3/16
1	1/16

Figure 3.9.27 IrDA Control Register

3.10 Serial Bus Interface (SBI)

The TMP91CW12 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I²C bus mode.

The serial bus interface is connected to an external device through P61 (SDA) and P62 (SCL) in the I²C bus mode; and through P60 (SCK), P61 (SO), and P62 (SI) in the clocked-synchronous 8-bit SIO mode. Each pin is specified as follows.

	ODE<ODE62:61>	P6CR<P62C, P61C, P60C>	P6FC<P62F, P61F, P60F>
I²C bus mode	11	11X	11X
Clock-synchronous 8-bit SIO mode	XX	011 010	X11

X: Don't care

3.10.1 Configuration

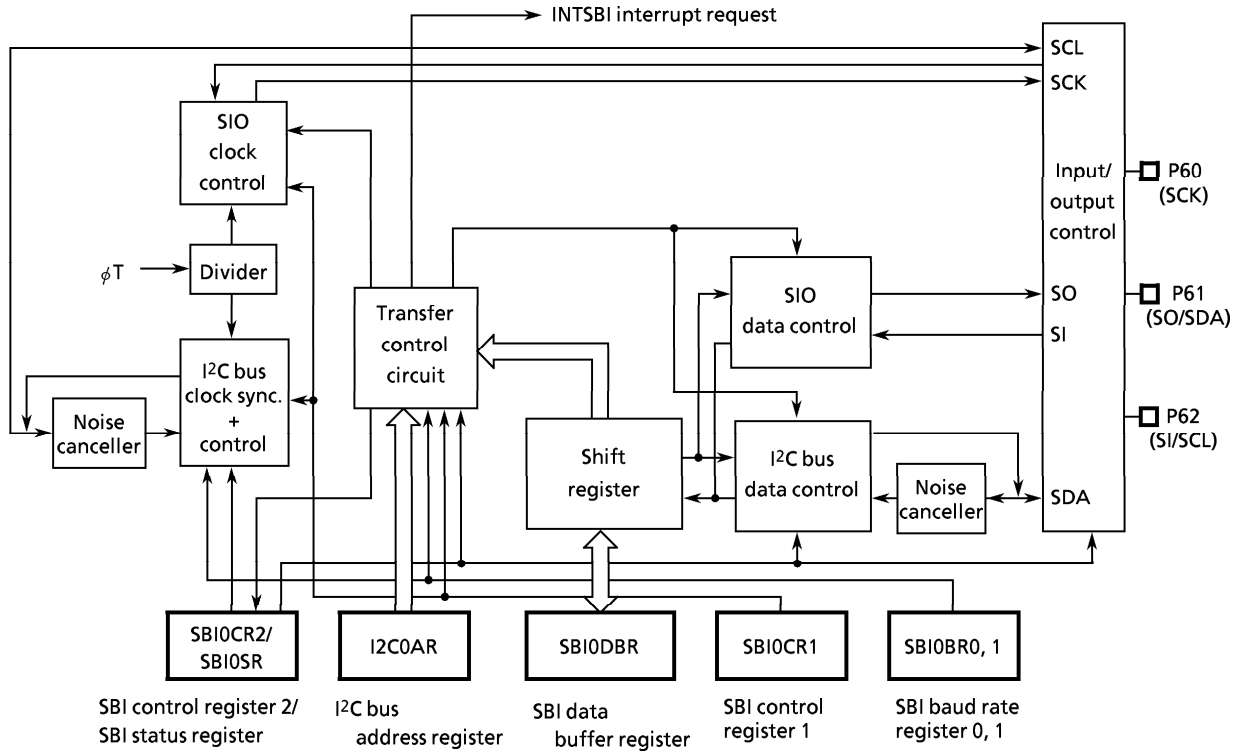


Figure 3.10.1 Serial Bus Interface (SBI)

3.10.2 Serial Bus Interface (SBI) Control

The following registers are used to control the serial bus interface and monitor the operation status.

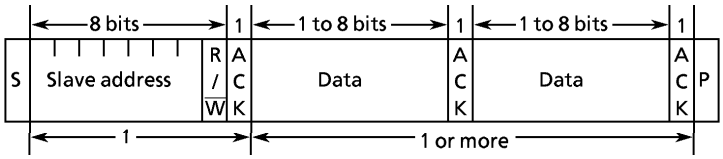
- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface data buffer register (SBI0DBR)
- I2C bus address register (I2C0AR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0BR0)
- Serial bus interface baud rate register 1 (SBI0BR1)

The above registers differ depending on a mode to be used.
Refer to section 3.10.4 "I2C bus Mode Control" and 3.10.7 "Clocked-synchronous 8-Bit SIO Mode Control".

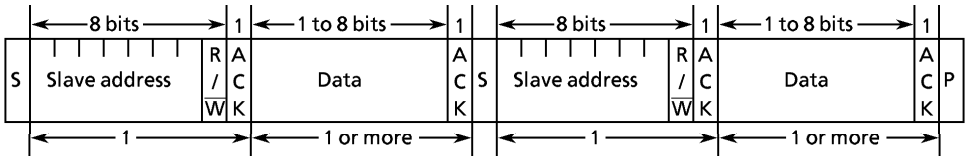
3.10.3 The Data Formats in the I2C Bus Mode

The data formats in the I2C bus mode are shown below.

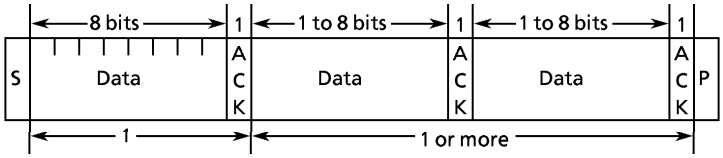
(a) Addressing format



(b) Addressing format (with restart)



(c) Free data format (Data transferred from master device to slave device)



S: Start condition
R/$\overline{W}$: Direction bit
ACK: Acknowledge bit
P: Stop condition

Figure 3.10.2 Data Format in the I2C Bus Mode

3.10.4 I²C Bus Mode Control

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the I²C bus mode.

Serial Bus Interface Control Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (0240H)	Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After reset	0	0	0	0		0	0	0/1 (Note 3)
Prohibit read-modify-write	Function	Number of transferred bits (Note 1)			Acknowledge mode specification generate 0: Not generate 1 : Generate		Internal serial clock selection and software reset monitor (Note 2)		

Internal serial clock selection <SCK2:0> at write

000	n = 5	– kHz	System clock: fc Clock gear: fc/1 fc = 16 MHz (Internal SCL output) $fscl = \frac{fc}{2^n + 8}$ [Hz]
001	n = 6	– kHz	
010	n = 7	– kHz	
011	n = 8	60.6 kHz	
100	n = 9	30.8 kHz	
101	n = 10	15.5 kHz	
110	n = 11	7.78 kHz	
111	(Reserved)	(Reserved)	

Software reset state monitor <SWRMON> at read

0	During software reset
1	Initial data

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse for acknowledge signal

Number of transferred bits

<BC2:0>	<ACK> = 0		<ACK> = 1	
	Number of Clock	Bit	Number of Clock	Bit
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

Note 1: Set the <BC2:0> to 000 before switching to a clock-synchronous 8-bit SIO mode.
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) "Serial clock".
Note 3: Initial data of SCK0 is "0", SWRMON is "1".
Note 4: This I²C bus circuit does not support high-speed mode, it supports standard mode only.

Figure 3.10.3 (a) Registers for the I²C Bus Mode

Serial Bus Interface Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0243H)	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note1)		W (Note1)	
	After reset	0	0	0	1	0	0	0	0
Prohibit read-modify-write	Function	Master/slave selection	Transmitter/ receiver selection	Start/stop condition generation	Cancel INTSBI interrupt request	Serial bus interface operating mode selection (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset generate write 10 and 01, then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

00	Port mode (Serial bus interface output disable)
01	Clocked-synchronous 8-bit SIO mode
10	I²C bus mode
11	(Reserved)

INTSBI interrupt request

0	–
1	Cancel interrupt request

Start/stop condition generation

0	Generate the stop condition
1	Generate the start condition

Transmitter/receiver selection

0	Receiver
1	Transmitter

Master/slave selection

0	Slave
1	Master

Note 1: Reading this register functions as SBI0SR register.
Note 2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I2C bus mode and clocked-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.10.3 (b) Registers for the I2C Bus Mode

Serial Bus Interface Status Register

SBI0SR (0243H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/slave status monitor	Transmitter/ receiver status monitor	I2C bus status monitor	INTSBI interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 1: Detected	Last received bit monitor 0: 0 1: 1

Last received bit monitor

0	Last received bit0
1	Last received bit1

GENERAL CALL detection monitor

0	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	–
1	Arbitration lost detected

INTSBI interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I2C bus status monitor

0	Free
1	Busy

Transmitter/receiver status monitor

0	Receiver
1	Transmitter

Master/slave status monitor

0	Slave
1	Master

Note: Writing in this register functions as SBI0CR2.

Figure 3.10.3 (c) Registers for the I2C Bus Mode

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (0244H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Write 0	IDLE2 0 : Stop 1 : Run						

Operation during IDLE2 mode

0	Stop
1	Operate

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0245H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write 0						

Control baud rate clock

0	Stop
1	Operate

Serial Bus Interface Data Buffer Register

SBI0DBR (0241H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transfer)							
After reset	Undefined							

Note 1: When writing transmitted data, start from the MSB (Bit7). Receiving data is placed from LSB (Bit0).

Note 2: SBI0DBR can't be read the written data. Therefore read-modify-write instruction (e.g., "BIT" instruction) is prohibitted.

Note 3: Written data in SBI0DBR is cleared by INTSBI signal.

I2C Bus Address Register

I2C0AR (0242H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Slave address selection when operating as slave device							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10.3 (d) Registers for the I2C Bus Mode

3.10.5 Control in I²C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to 1 for operation in the acknowledge mode. The TMP91CW12 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.
Clear the <ACK> to 0 for operation in the non-acknowledge mode. The TMP91CW12 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

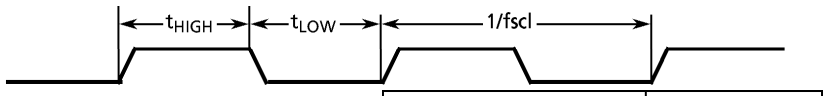
(2) Number of transfer bits

The SBI0CR1<BC2:0> is used to select a number of bits for next transmitting and receiving data.
Since the <BC2:0> is cleared to 000 as a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the <BC2:0> retains a specified value.

(3) Serial clock

① Clock source

The SBI0CR1<SCK2:0> is used to select a maximum transfer frequency outputed on the SCL pin in the master mode. Set the baud rates, which have been calculated according to the formula below, to meet the specifications of the I²C bus, such as the smallest pulse width of t_{LOW}.



$t_{LOW} = 2^{n-1}/f_{SBI}$

$t_{HIGH} = 2^{n-1}/f_{SBI} + 8/f_{SBI}$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \dfrac{f_{SBI}}{2^n + 8}$

SBI0CR1<SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Note 1: f_{SBI} is the clock f_{FPH}.
Note 2: It's prohibit to use fc/16 prescaler clock when using SBI block (I²C bus and clock synchronous).

Figure 3.10.4 Clock Source

② Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which pulls down a clock line to low-level, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.
The TMP91CW12 has a clock synchronization function for normal data transfer even when more than one master exists on a bus.
The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

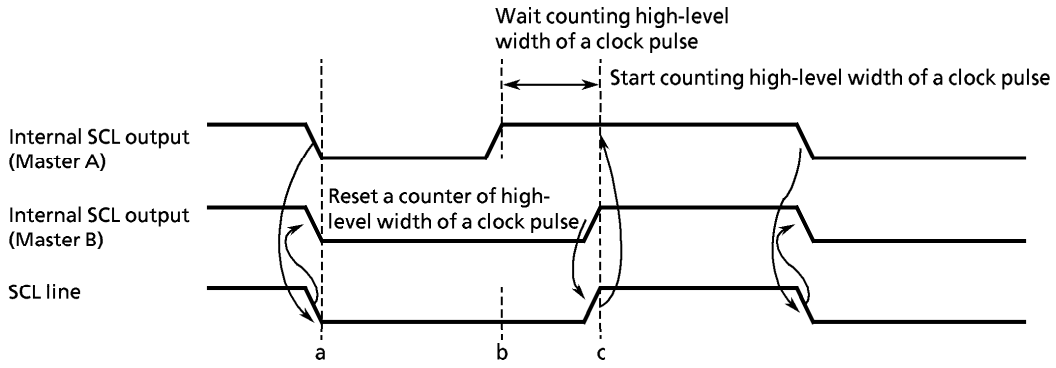


Figure 3.10.5 Clock Synchronization

As master A pulls down the internal SCL output to the low level at point a, the SCL line of the bus becomes the low level. After detecting this situation, master B resets a counter of high-level width of an own clock pulse and sets the internal SCL output to the low level.
Master A finishes counting low-level width of an own clock pulse at point b and sets the internal SCL output to the high level. Since master B holds the SCL line of the bus at the low level, master A waits for counting high-level width of an own clock pulse. After master B finishes counting low-level width of an own clock pulse at point c and master A detects the SCL line of the bus at the high level, and starts counting high level of an own clock pulse.
The clock pulse on the bus is determinded by the master device with the shortest high-level width and the master device with the longest low-level width from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When the TMP91CW12 is used as a slave device, set the slave address <SA6:0> and <ALS> to the I2C0AR. Clear the <ALS> to 0 for the address recognition mode.

(5) Master/slave selection

Set the SBI0CR2<MST> to 1 for operating the TMP91CW12 as a master device. Clear the <MST> to 0 for operation as a slave device. The <MST> is cleared to 0 by the hardware after a stop condition on a bus is detected or arbitration is lost.

(6) Transmitter/receiver selection

Set the SBI0CR2<TRX> to 1 for operating the TMP91CW12 as a transmitter. Clear the <TRX> to 0 for operation as a receiver. When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2C0AR or a GENERAL CALL is received (All 8-bit data are 0 after a start condition), the <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 by the hardware if the bit is 0. In the master mode, after an acknowledge signal is returned from the slave device, the <TRX> is cleared to 0 by the hardware if a transmitted direction bit is 1, and is set to 1 by the hardware if it is 0. When an acknowledge signal is not returned, the current condition is maintained.
The <TRX> is cleared to 0 by the hardware after a stop condition on the I2C bus is detected or arbitration is lost.

(7) Start/stop condition generation

When the SBI0SR<BB> is 0, slave address and direction bit which are set to the SBI0DBR are output on a bus after generating a start condition by writing 1 to the SBI0CR2<MST, TRX, BB, and PIN>. It is necessary to set transmitted data to the data buffer register (SBI0DBR) and set 1 to <ACK> beforehand.

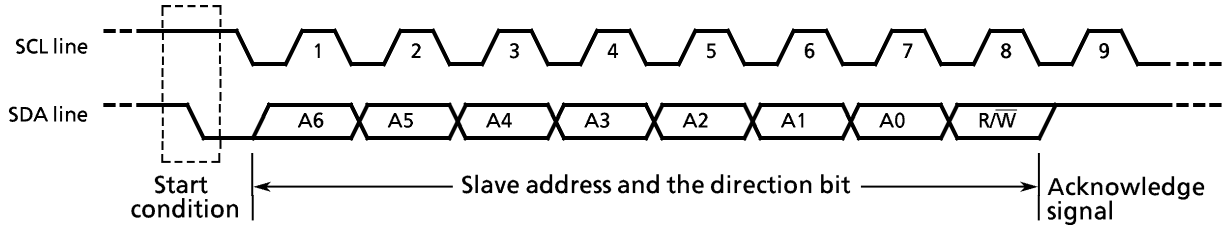


Figure 3.10.6 Start Condition Generation and Slave Address Generation

When the <BB> is 1, a sequence of generating a stop condition is started by writing 1 to the <MST, TRX, and PIN>, and 0 to the <BB>. Do not modify the contents of <MST, TRX, BB and PIN> until a stop condition is generated on a bus.

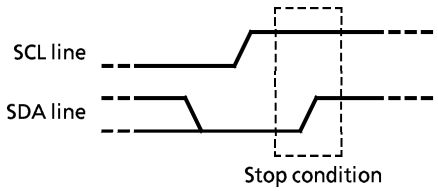


figure 3.10.7 Stop Condition Generation

The bus condition can be indicated by reading the contents of the SBI0SR<BB>. The <BB> is set to 1 when a start condition on a bus is detected,and is cleared to 0 when a stop condition is detected on a bus.

(8) Interrupt service request and cancel

When a serial bus interface interrupt request (INTSBI) occurs, the SBI0CR2<PIN> is cleared to 0. During the time that the <PIN> is 0, the SCL line is pulled down to the low level.
The <PIN> is cleared to 0 when 1 word of data is transmitted or received. Either writing/reading data to/from the SBI0DBR sets the <PIN> to 1.
The time from the <PIN> being set to 1 until the SCL line is released takes t_{LOW}.
In the address recognition mode (<ALS>=0), the <PIN> is cleared to 0 when the received slave address is the same as the value set at the I2C0AR or when a GENERAL CALL is received (All 8-bit data are 0 after a start condition). Although the SBI0CR2<PIN> can be set to 1 by the program, the <PIN> is not cleared to 0 when it is written 0.

(9) Serial bus interface operation mode selection

The SBI0CR2<SBIM1:0> is used to specify the serial bus interface operation mode. Set the <SBIM1:0> to 10 when used in the I²C bus mode after confirming pin condition of serial bus interface to "H".
Switch a mode to port after confirming a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus in the I²C bus mode, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.
A data on the SDA line is used for bus arbitration of the I²C bus.
The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on the bus. Master A and master B output the same data until point a. After master A outputs L and master B, H, the SDA line of the bus is wire-AND and the SDA line is pulled down to the low level by master A. When the SCL line of the bus is pulled up at point b, the slave device reads data on the SDA line, that is, data in master A. A data transmitted from master B becomes invalid. The state in master B is called "ARBITRATION LOST". Master B device which loses arbitration releases the internal SDA output 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.

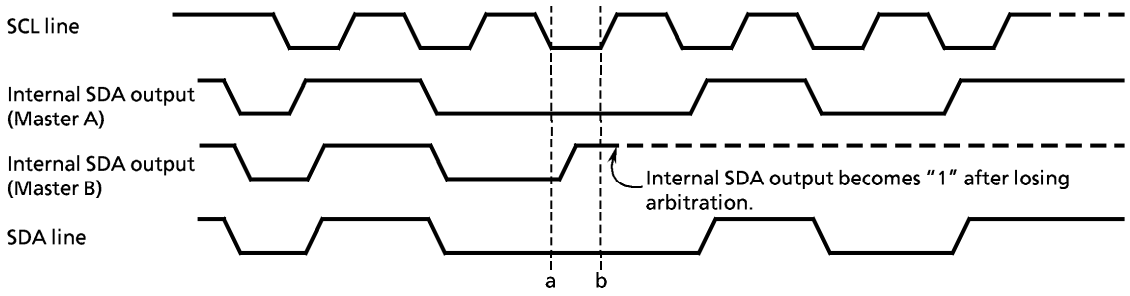


Figure 3.10.8 Arbitration Lost

The TMP91CW12 compares levels of the SDA line on a bus with those of the internal SDA output at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the SBI0SR<AL> is set to 1.
When the <AL> is set to 1, the SBI0SR<MST, TRX> are cleared to 0 and the mode is switched to a slave receiver mode. Thus, clock output is stopped in data transfer after setting <AL> = "1".
The <AL> is cleared to 0 by writing/reading data to/from the SBI0DBR or writing data to the SBI0CR2.

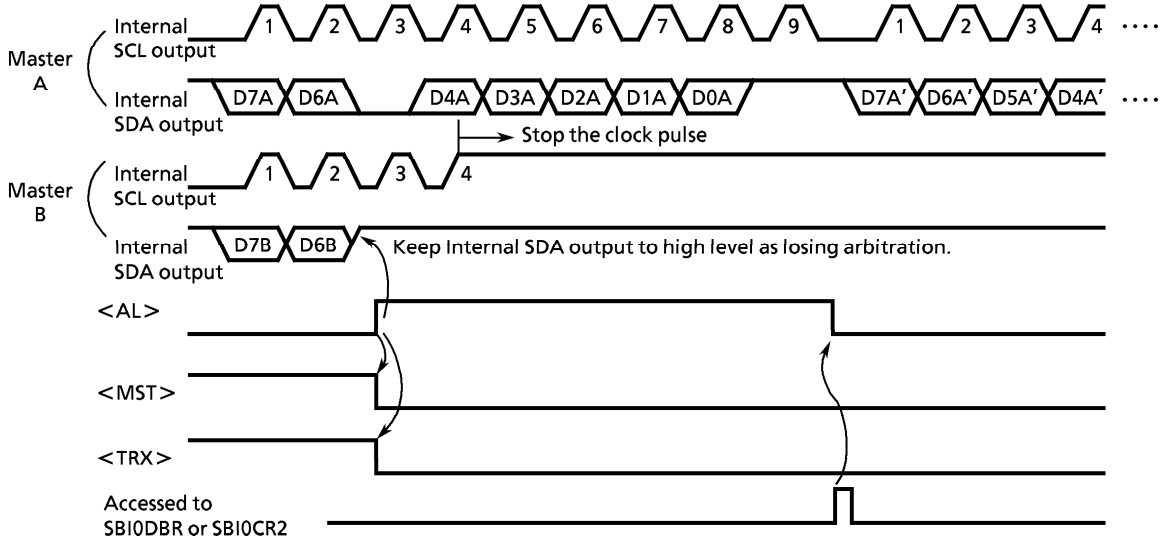


Figure 3.10.9 Example of when TMP91CW12 is a Master Device B (D7A = D7B and D6A = D6B)

(11) Slave address match detection monitor

The SBI0SR<AAS> is set to 1 in the slave mode, in the address recognition mode (I2C0AR<ALS> =0), when receiving GENERAL CALL or a slave address with the same value that is set to the I2C0AR. When the <ALS> is 1, the <AAS> is set to 1 after receiving the first 1-word of data. The <AAS> is cleared to 0 by writing/reading data to/from a data buffer register SBI0DBR.

(12) GENERAL CALL detection monitor

The SBI0SR<AD0> is set to 1 in the slave mode, when GENERAL CALL is received (All 8-bit received data is 0, after a start condition). The AD0 is cleared to 0 when a start or stop condition is detected on a bus.

(13) Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to the SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the SBI0SR<LRB>.

(14) Software reset function

Software reset function is used to initialize SBI circuit, when SBI is rocked by external noises, etc.
An internal reset signal pulse is generated by setting SBI0CR2<SWRST1:0> to 10 and 01. The internal state of SBI circuit is initialized by it. All command except SBI0CR2<SBIM1:0> registers and status registers are initialized too.
The SBI0CR1<SWRMON> is automatically set to "1" after SBI circuit has been initialized.

(15) Serial bus interface data buffer register (SBI0DBR)

The received data can be read and the transferred data can be written by reading or writing the SBI0DBR.
In the master mode, the start condition is generated after the slave address and the direction bit are set to this register.

(16) I2C bus address register (I2C0AR)

I2C0AR<SA6:0> is used to set the slave address when the TMP91CW12 functions as the slave device. The slave address output from the master device is recognized by setting I2C0AR<ALS> to 0. The data format is the addressing format. When the slave address is not recognized at the <ALS> = 1, the data format is the free data format.

(17) Baud rate register (SBI0BR1)

Write 1 to SBI0BR1<P4EN> before hands.

(18) Setting register for operation during IDLE2 mode (SBI0BR0)

SBI0BR0<I2SBI0> is the register setting operation/stop during IDLE2 mode. Therefore, setting <I2SBI0> is necessary before the HALT instruction is executed.

3.10.6 Data Transfer in I2C Bus Mode

(1) Device initialization

Set the SBI0BR1<P4EN>, SBI0CR1<ACK, SCK2:0>. Set SBI0BR1 to 1 and clear bit7 to 5 and 3 in the SBI0CR1 to 0.
Set a slave address<SA6:0>and the <ALS> (<ALS>=0 when an addressing format) to the I2C0AR.
For specifying the default setting to a slave receiver mode, clear 0 to the <MST, TRX, BB> and set 1 to the <PIN>, 10 to the <SBIM1:0>.

(2) Start condition and slave address generation

① Master mode

In the master mode, the start condition and the slave address are generated as follows.
Check a bus free status (when <BB>=0).
Set the SBI0CR1<ACK> to 1 (Acknowledge mode) and specify a slave address and a direction bit to be transmitted to the SBI0DBR.
When the <BB> is 0, the start condition are generated by writing 1111 to the SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the start condition, nine clocks are output from the SCL pin. While eight clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock, the SDA line is released and the acknowledge signal is received from the slave device.
An INTSBI interrupt request occurs at the falling edge of the 9th clock. The <PIN> is cleared to 0. In the master mode, the SCL pin is pulled down to the low level while the <PIN> is 0. When an interrupt request occurs, the <TRX> changes according to the direction bit only when an acknowledge signal is returned from the slave device.

② Slave mode

In the slave mode, the start condition and the slave address are received.
After the start condition is received from the master device, while eight clocks are output from the SCL pin, the slave address and the direction bit which are output from the master device are received.
When the GENERAL CALL or the same address as the slave address set to the I2C0AR is received, the SDA line is pulled down to the low level at the 9th clock, and the acknowledge signal is output.
An INTSBI interrupt request occurs at the falling edge of the 9th clock. The <PIN> is cleared to 0. In the slave mode, the SCL line is pulled down to the low level while the <PIN> is 0.

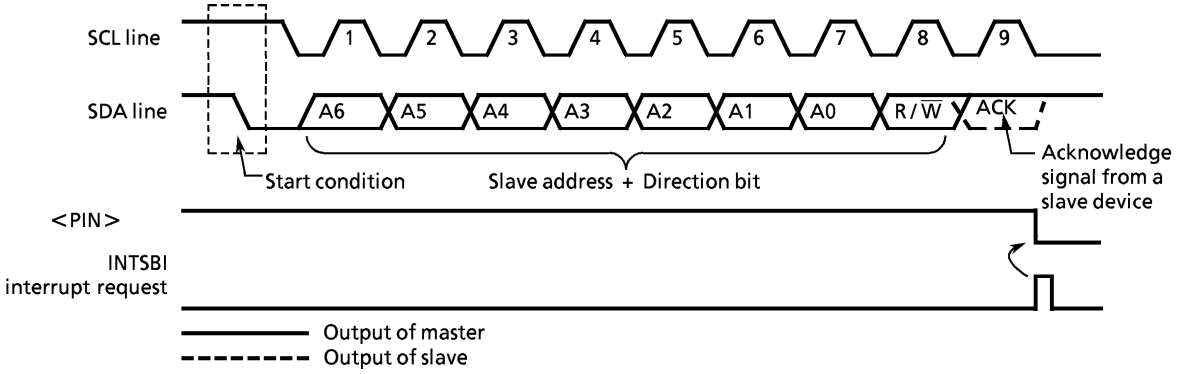


Figure 3.10.10 Start Condition Generation and Slave Address Transfer

(3) 1-word data transfer

Check the <MST> by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

① When the <MST> is 1 (Master mode)

Check the <TRX> and determine whether the mode is a transmitter or receiver.

When the <TRX> is 1 (Transmitter mode)
Check the <LRB>. When the <LRB> is 1, a receiver does not request data. Implement the process to generate a stop condition (Refer to 3.10.6 (4)) and terminate data transfer.
When the <LRB> is 0, the receiver requests new data. When the next transmitted data is 8 bits, write the transmitted data to SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2:0>, set the <ACK> to 1 and write the transmitted data to the SBI0DBR. After writing the data, the <PIN> becomes 1, a serial clock pulse is generated for transferring a new 1 word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, an interrupt request occurs. The <PIN> becomes 0 and the SCL line is pulled down to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the <LRB> checking above.

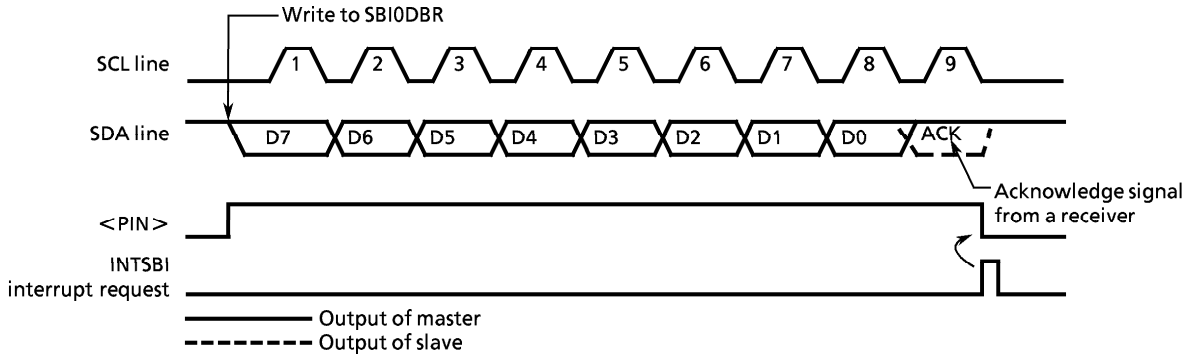


Figure 3.10.11 Example when <BC2:0> = 000, <ACK> = 1 in Transmitter Mode

When the <TRX> is 0 (Receiver mode)
When the next transmitted data is other than 8 bits, set the <BC2:0> again. Set the <ACK> to 1 and read the received data from the SBI0DBR to release the SCL line (data which is read immediately after a slave address is sent is undefined). After the data is read, the <PIN> becomes 1.
Serial clock pulse for transfering new 1 word of data is defined SCL and outputs "L" level from SDA pin with acknowledge timing.
An INTSBI interrupt request then occurs and the <PIN> becomes 0. Then the TMP91CW12 pulls down the SCL pin to the low level. The TMP91CW12 outputs a clock pulse for 1 word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.

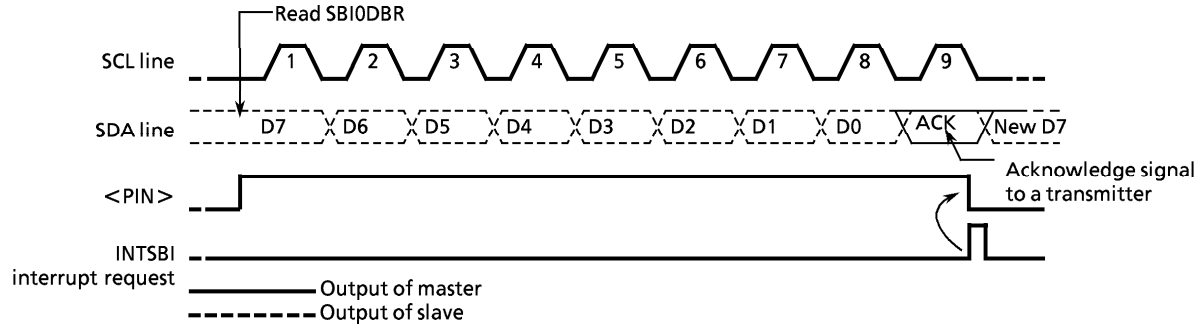


Figure 3.10.12 Example of when <BC2:0> = 000, <ACK> = 1 in Receiver Mode

In order to terminate transmitting data to a transmitter, clear the <ACK> to 0 before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set the <BC2:0> to 001 and read the data. The TMP91CW12 generates a clock pulse for a single bit data transfer. 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 an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, the TMP91CW12 generates a stop condition (Refer to 3.10.6 (4)) and terminates data transfer.

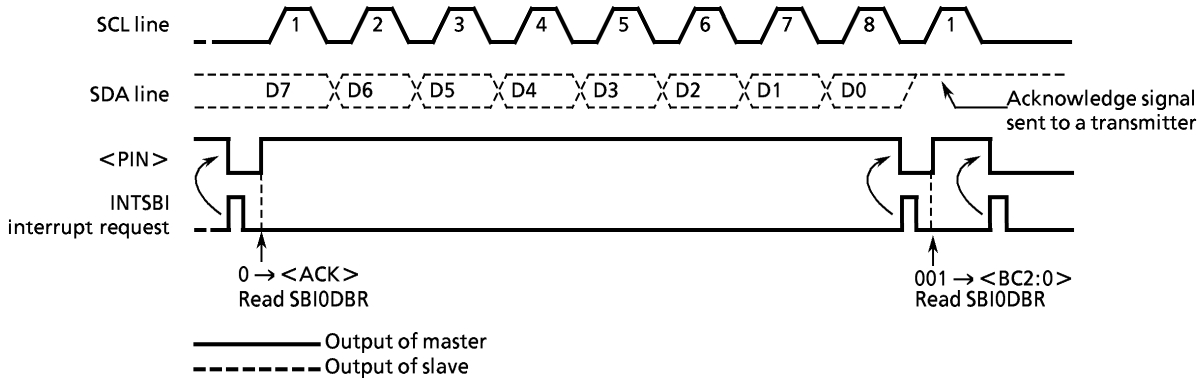


Figure 3.10.13 Termination of Data Transfer in Master Receiver Mode

② When the <MST> is 0 (Slave mode)

In the slave mode, the TMP91CW12 operates either in normal slave mode or in slave mode after losing arbitration.
In the slave mode, an INTSBI interrupt request occurs when the TMP91CW12 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete, or after matching a received slave address. In the master mode, the TMP91CW12 operates in a slave mode if it is losing arbitration. An INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBI interrupt request occurs, the <PIN> is cleared to 0, and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBI0DBR or setting the <PIN> to 1 releases the SCL pin after taking t_{LOW} time.
Check the SBI0SR<AL>, <TRX>, <AAS>, and <AD0> and implements processes according to conditions listed in the next table.

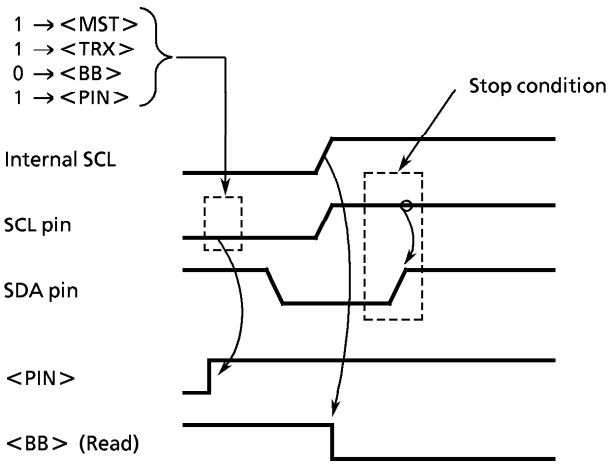
Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP91CW12 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 the <BC2:0> and write transmitted data to the SBI0DBR.
	0	1	0	In the slave receiver mode, the TMP91CW12 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, 1-word data is transmitted.	Check the <LRB>. If the <LRB> is set to 1, set the <PIN> to 1 since the receiver does not request next data. Then, clear the <TRX> to 0 release the bus. If the <LRB> is cleared to 0, set the number of bits in a word to the <BC2:0> and write transmitted data to the SBI0DBR since the receiver requests next data.
0	1	1	1/0	The TMP91CW12 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL of which the value of the direction bit sent from another master is 0.	Read the SBI0DBR for setting the <PIN> to 1 (Reading dummy data) or set the <PIN> to 1.
		0	0	The TMP91CW12 loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, the TMP91CW12 receives a slave address or GENERAL CALL of which the value of the direction bit sent from the master is 0.	
		0	1/0	In the slave receiver mode, the TMP91CW12 terminates receiving of 1-word data.	Set the number of bits in a word to the <BC2:0> and read received data from the SBI0DBR.

(4) Stop condition generation

When the SBI0SR<BB> is 1, a sequence of generating a stop condition is started by setting 1 to the SBI0CR2<MST, TRX, PIN>, and 0 to the <BB>. Do not modify the contents of the <MST, TRX, BB, PIN> until a stop condition is generated on a bus. When a SCL line of bus is pulled down by other devices, the TMP91CW12 generates a stop condition after they release a SCL line.
When SBI0CR2<MST, TRX, PIN> are written 1 and <BB> is written to 0, <BB> changes to 0 by internal SCL changes to 1, without waiting stop condition.
To check whether SCL and SDA pin are 1 by sensing their ports is needed to detect bus free condition.

(Single master)



(Multi master)

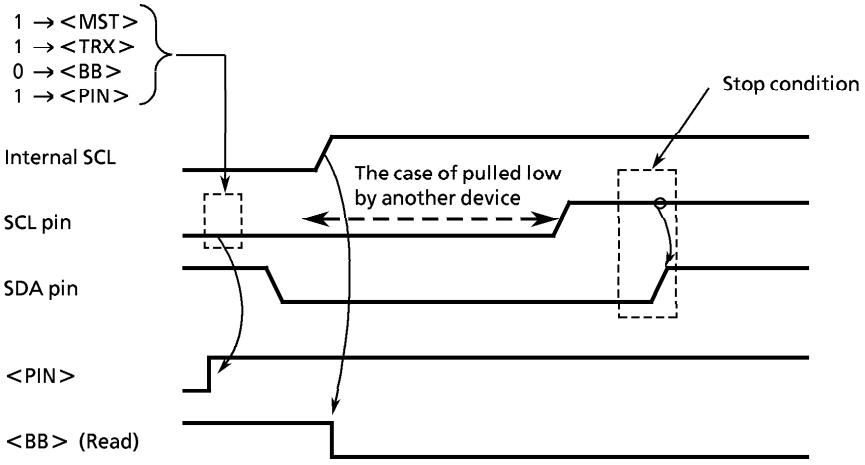


Figure 3.10.14 Stop Condition Generation

(5) Restart

Restart is used during data transfer between a master device and a slave device to change the data transfer direction.
The following description explains how to restart when the TMP91CW12 is in master mode.
Clear SBI0CR2<MST, TRX, BB> to 0 and set SBI0CR2<PIN> to 1 to release the bus. The SDA line remains high and the SCL pin is released. Since a stop condition has not been generated on the bus, other devices assume the bus to be in busy state.
And confirm SCL pin, that SCL pin is released and become bus-free state by SBI0SR <BB> = "0" or signal level "1" of SCL pin in port mode. Check the <LRB> until it becomes 1 to check that the SCL line on a bus is not pulled down to the low level by other devices. After confirming that the bus remains in a free state, generate a start condition using the procedure described in 3.10.15.
In order to satisfy the setup time requirements when restarting, take at least 4.7 μs of waiting time by software from the time of restarting to confirm that the bus is free until the time to generate the start condition.

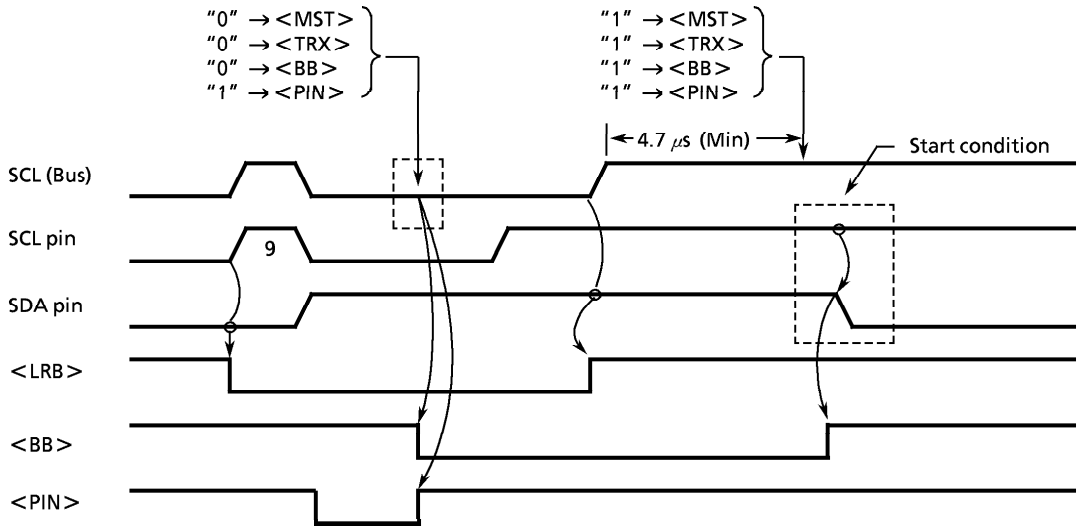


Figure 3.10.15 Timing Chart for Generate Restart

3.10.7 Clocked-synchronous 8-Bit SIO Mode Control

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the clocked-synchronous 8-bit SIO mode.

Serial Bus Interface Control Register 1

SBI0CR1 (0240H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	W					W		W
After reset	0	0	0	0		0	0	0
Function	Indicate transfer start/stop 0 : Stop 1 : Start	Continue/ abort transfer 0 : Continue transfer 1 : Abort transfer	Transfer mode select 00 : Transmit mode 01 : (Reserved) 10 : Transmit/receive mode 11 : Receive mode			Serial clock select and reset monitor		

Serial clock select <SCK2:0> at write

000	n = 4	1 MHz	System clock: fc Clock gear: fc/1 fc = 16 MHz (output to SCK pin) $fscl = \frac{fc}{2^n}$ [Hz]
001	n = 5	500 kHz	
010	n = 6	250 kHz	
011	n = 7	125 kHz	
100	n = 8	62.5 kHz	
101	n = 9	31.25 kHz	
110	n = 10	15.625 kHz	
111	–	External clock (Inputted from SCK pin)	

Transfer mode select

00	8-bit transmit mode
01	(Reserved)
10	8-bit transmit/receive mode
11	8-bit receive mode

Continue/abort transfer

0	Continue transfer
1	Abort transfer (Automatically cleared after avort)

Indicate transfer start/stop

0	Stop
1	Start

Note : Set the tranfer mode and the serial clock after setting <SIOS> to 0 and <SIOINH> to 1.

Serial Bus Interface Data Buffer Register

SBI0DBR (0241H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transfer)							
After reset	Undefined							

Figure 3.10.16 (a) Register for the SIO Mode

Serial Bus Interface Control Register 2

SBI0CR2 (0243H)

	7	6	5	4	3	2	1	0
Bit symbol					SBIM1	SBIM0	–	–
Read/Write					W		W	W
After reset					0	0	0	0
Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		(Note 2)	(Note 2)

Prohibit read-modify-write

Serial bus interface operation mode selection

00	Port mode (Serial bus interface output disable)
01	Clocked-synchronous 8-bit SIO mode
10	I²C bus mode
11	(Reserved)

Note 1: Set the SBI0CR1<BC2:0> to 000 before switching to a clocked-synchronous 8-bit SIO mode.
Note 2: Please always write "00" to SBICR2<1:0>.

Serial Bus Interface Status Register

SBI0SR (0243H)

	7	6	5	4	3	2	1	0
Bit symbol					SIOF	SEF		
Read/Write					R			
After reset					0	0		
Function					Serial transfer operating status monitor	Shift operating status monitor		

Shift operating status monitor

0	Shift operation terminated
1	Shift operation in process

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in process

Figure 3.10.16 (b) Registers for the SIO Mode

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (0244H)

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Write 0	IDLE2 0 : Stop 1 : Run						

Prohibit read-modify-write

Operation during IDLE2 mode

0	Stop
1	Operate

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0245H)

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write 0						

Prohibit read-modify-write

Control baud rate clock

0	Stop
1	Operate

Figure 3.10.16 (c) Registers for the SIO Mode

(1) Serial clock

① Clock source

The SBI0CR1<SCK2:0> is used to select the following functions.

Internal clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the SCK pin. The SCK pin becomes a high level when data transfer starts. When writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is complete.

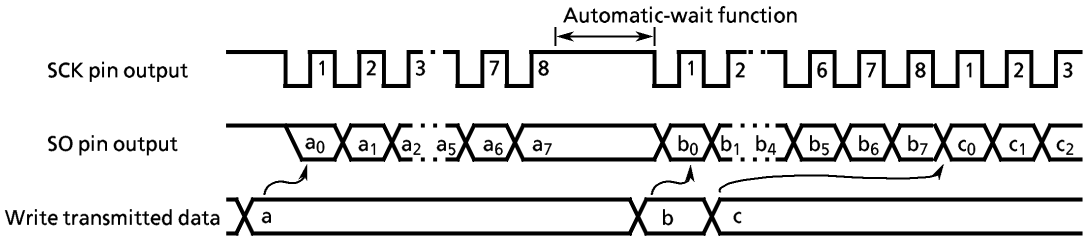


Figure 3.10.17 Automatic-wait Function

External clock (<SCK2:0> = 111)

An external clock supplied to the SCK pin is used as the serial clock. In order to ensure shift operation, a pulse width mentioned below is required for both high-level and low-level in the serial clock. The maximum data transfer frequency is 1 MHz (when fc = 16 MHz).

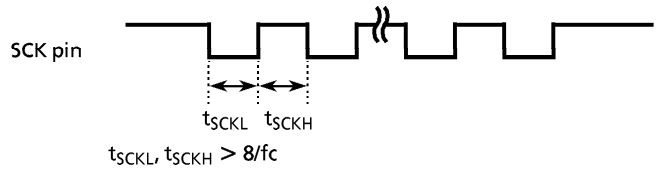


Figure 3.10.18 Maximum Data Transfer Frequency when External Clock Input

② Shift edge

The leading edge is used to transmit data, and the trailing edge is used to receive data.

Leading edge shift
Data is shifted on the leading edge of the serial clock (at a falling edge of the SCK pin input/output).

Trailing edge shift
Data is shifted on the trailing edge of the serial clock (at a rising edge of the SCK pin input/output).

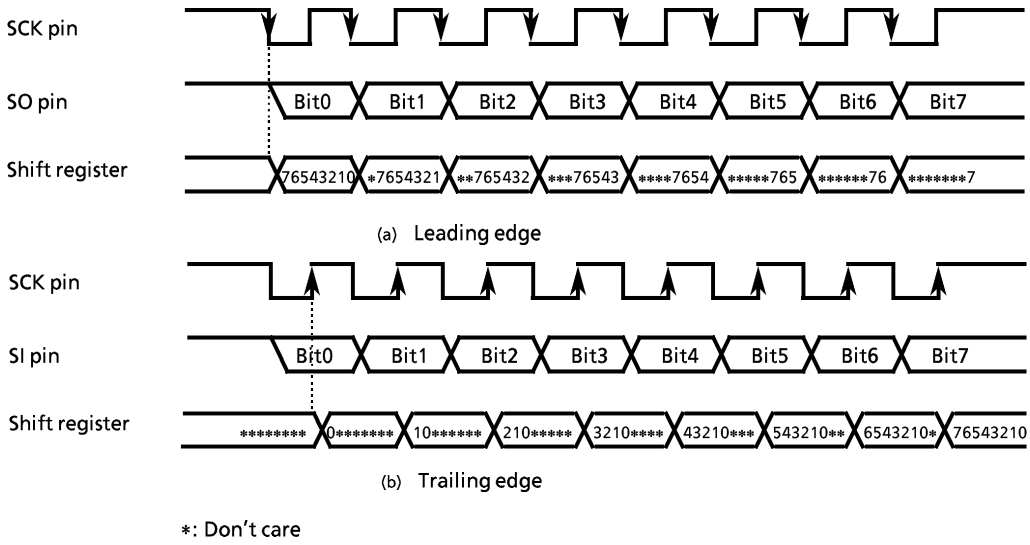


Figure 3.10.19 Shift Edge

(2) Transfer mode

The SBI0CR1<SIOM1:0> is used to select a transmit, receive, or transmit/receive mode.

① 8-bit transmit mode

Set a control register to a transmit mode and write transmit data to the SBI0DBR.
After the transmit data is written, set the SBI0CR1<SIOS> to 1 to start data transfer. The transmitted data is transferred from the SBI0DBR to the shift register and output to the SO pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the transmit data is transferred to the shift register, the SBI0DBR becomes empty. The INTSBI (Buffer empty) interrupt request is generated to request new data.
When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new transmit data is written, automatic-wait function is canceled.
When the external clock is used, data should be written to the SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBI0DBR by the interrupt service program.
When the transmit is started, after the SBI0SR<SIOF> goes 1 output from the SO pin holds final bit of the last data until falling edge of the SCK.
Transmitting data is ended by clearing the <SIOS> to 0 by the buffer empty interrupt service program or setting the <SIOINH> to 1. When the <SIOS> is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the <SIOF> (Bit3 in the SBI0SR) to be sensed. The SBI0SR<SIOF>is cleared to 0 when transmitting is complete. When the <SIOINH> is set to 1, transmitting data stops. The <SIOF> turns 0.
When the external clock is used, it is also necessary to clear the <SIOS> to 0 before new data is shifted; otherwise, dummy data is transmitted and operation ends.

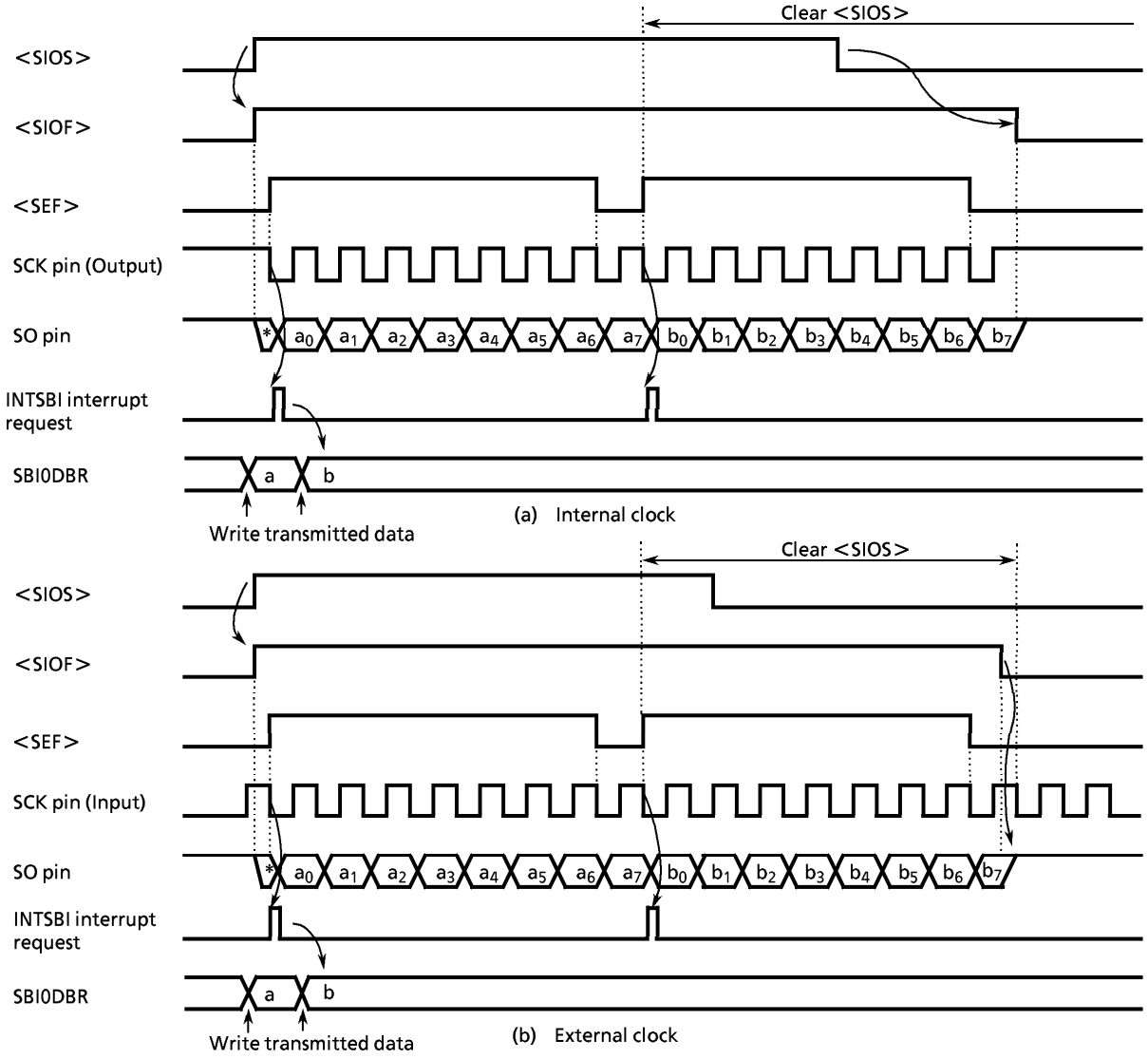


Figure 3.10.20 Transfer Mode

Example: Program to stop transmitting data (when external clock is used).

```
STEST1 :  BIT   SEF, (SBI0SR)          ;  If <SEF> = 1  then  loop
          JR    NZ , STEST1

STEST2 :  BIT   0, (P6)                ;  If SCK = 0    then  loop
          JR    Z , STEST2
          LD    (SBI0CR1) , 00000111B  ;  <SIOS> ← 0
```

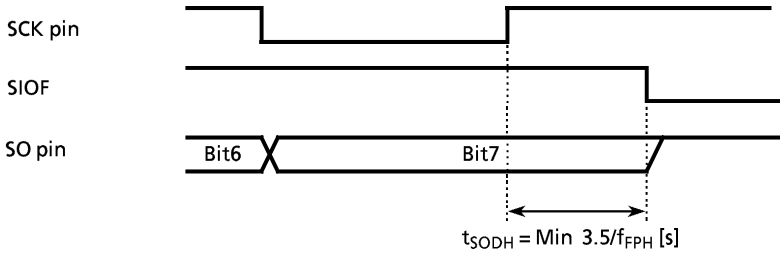


Figure 3.10.21 Transmitted Data Hold Time at End of Transmit

② 8-bit receive mode

Set the control register to receive mode and the SBI0CR1<SIOS> to 1 for switching to receive mode. Data is received from the SI pin to the shift register in synchronous with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBI0DBR. The INTSBI (Buffer full) interrupt request is generated to request of reading the received data. The data is then read from the SBI0DBR by the interrupt service program.
When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated until the received data is read from the SBI0DBR.
When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read from the SBI0DBR before next serial clock is input. If the received data is not read, further data to be received is canceled. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read.
Receiving data is ended by clearing the <SIOS> to 0 by the buffer full interrupt service program or setting the <SIOINH> to 1. When the <SIOS> is cleared, received data is transferred to the SBI0DBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set the SBI0SR<SIOF> to be sensed. The <SIOF> is cleared to 0 when receiving is complete. After confirming that receiving has ended, the last data is read. When the <SIOINH> is set to 1, receiving data stops. The <SIOF> turns 0 (The received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is switched, the SBI0DBR contents are lost. In case that the mode needs to be switched, receiving data is concluded by clearing the <SIOS> to 0, read the last data, and then switch the mode.

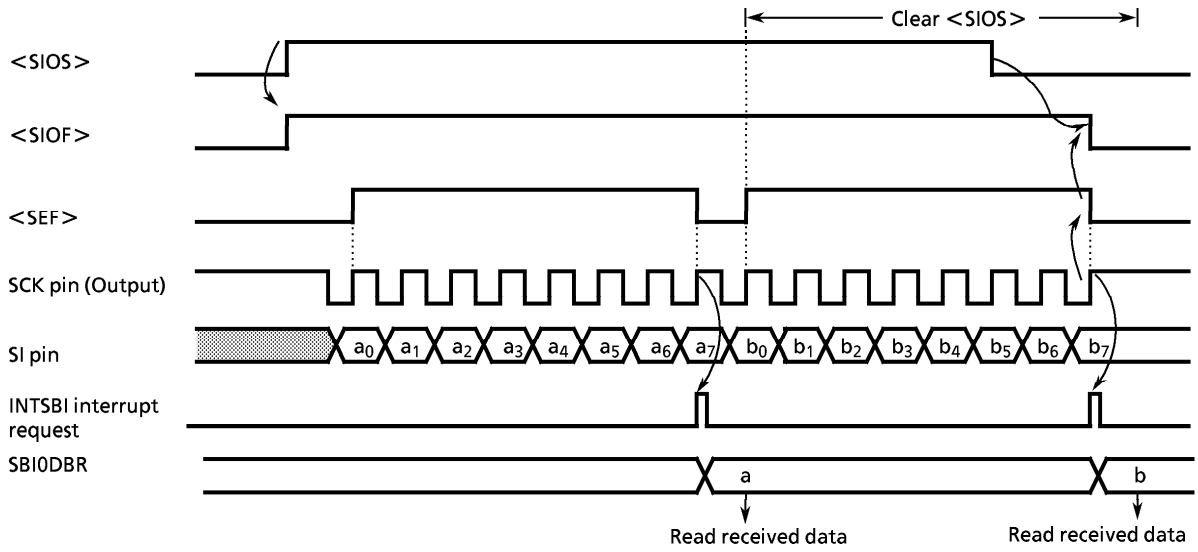


Figure 3.10.22 Receive Mode (Example: Internal clock)

③ 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to the SBI0DBR. After the data is written, set the SBI0CR<SIOS> to 1 to start transmitting/receiving. When transmitting, the data is output from the SO pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI pin on the trailing edges of the serial clock. 8-bit data is transferred from the shift register to the SBI0DBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, automatic-wait function is initiated until received data is read and next data is written.

When the external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before new shift operation is executed. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read and transmitted data is written.

When the transmit is started, after the SBI0SR<SIOF> goes 1 output from the SO pin holds final bit of the last data until falling edge of the SCK.

Transmitting/receiving data is ended by clearing the <SIOS> to 0 by the INTSBI interrupt service program or setting the SBI0CR1<SIOINH> to 1. When the <SIOS> is cleared, received data is transferred to the SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted/received by the program, set the SBI0SR to be sensed. The <SIOF> becomes 0 after transmitting/receiving is complete. When the <SIOINH> is set, transmitting/receiving data stops. The <SIOF> turns 0.

Note: When the transfer mode is switched, the SBI0DBR contents are lost. In case that the mode needs to be switched, conclude transmitting/receiving data by clearing the <SIOS> to 0, read the last data, and then switch the transfer mode.

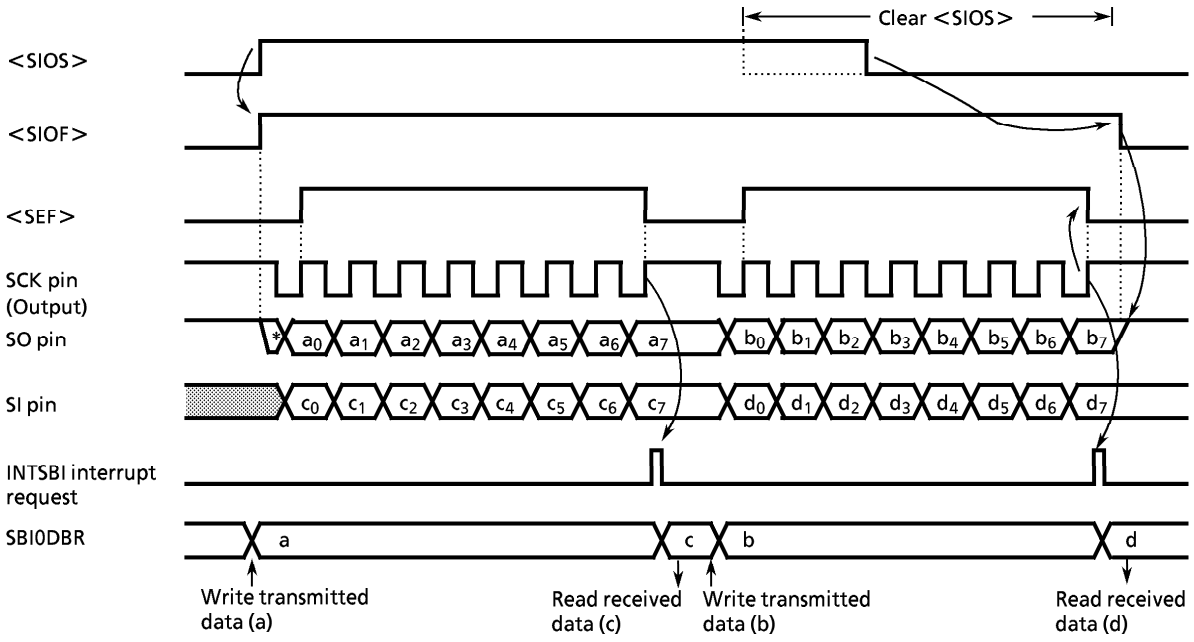


Figure 3.10.23 Transmit/Receive Mode (Example: Internal clock)

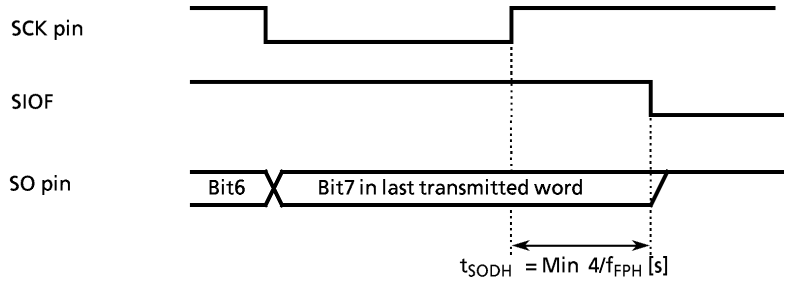


Figure 3.10.24 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Analog/Digital Converter

TMP91CW12 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 8-channel analog input.
Figure 3.11.1 is a block diagram of the AD converter. The 8-channel analog input pins (AN0 to AN7) are shared by input-only port A and can thus be used as an input port.

Note : When the power is reduced by setting IDLE2, IDLE1, or STOP mode, with some timings, the system may enter standby mode even though the internal comparator is still enabled. Therefore, be sure to check that AD converter operations are halted before executing a HALT instruction.

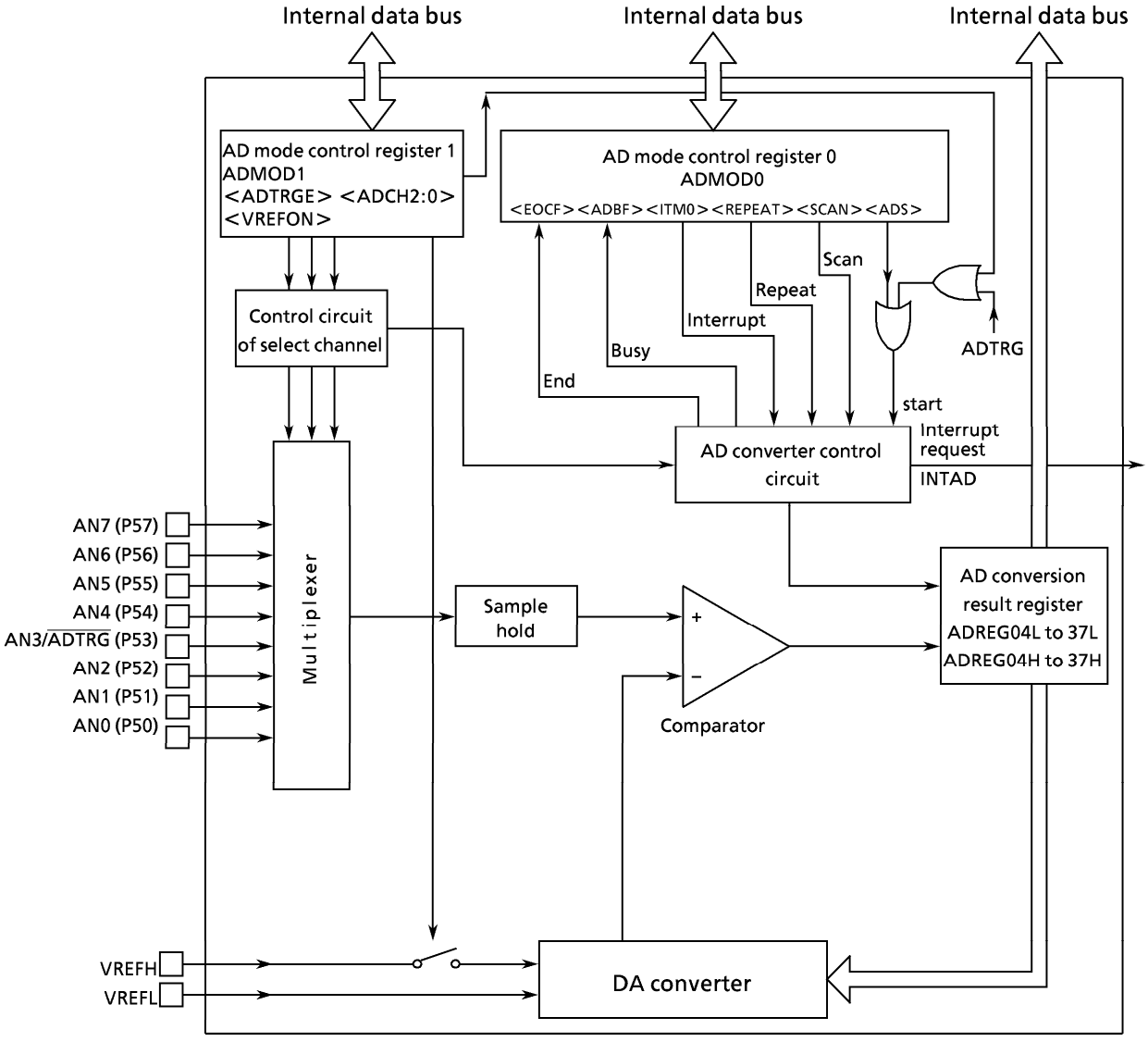


Figure 3.11.1 AD Converter Block Diagram

3.11.1 Analog/Digital Converter Registers

The AD converter is controlled by two AD mode control registers: ADMOD0 and ADMOD1. Eight AD conversion data upper and lower registers (ADREG04H/L, ADREG15H/L, ADREG26H/L, and ADREG37H/L) store the AD conversion results.
Figures 3.11.2 shows registers related to the AD converter.

AD Mode Control Register 0

ADMOD0 (02B0H)

	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	AD conversion end flag 0: Conversion in progress 1: Conversion complete	AD conversion busy flag 0: Conversion stopped 1: Conversion in progress	Always write 0.	Always write 0.	Interrupt specification in conversion channel fixed repeat mode 0: Every conversion 1: Every fourth conversion	Repeat mode specification 0 : Single conversion mode 1 : Repeat conversion mode	Scan mode specification 0: Conversion channel fixed mode 1: Conversion channel scan mode	AD conversion start 0: Don't care 1: Start conversion Always read as 0.

AD conversion start

0	Don't care
1	Start AD conversion

Note: Always read as 0.

AD scan mode setting

0	AD conversion channel fixed mode
1	AD conversion channel scan mode

AD repeat mode setting

0	AD single conversion mode
1	AD repeat conversion mode

Specify AD conversion interrupt for channel fixed repeat conversion mode

	Channel fixed repeat conversion mode <SCAN> = 0, <REPEAT> = 1
0	Generate interrupt every conversion
1	Generate interrupt every fourth conversion

AD conversion busy flag

0	AD Conversion stopped
1	AD conversion in progress

AD conversion end flag

0	Before or during AD conversion
1	AD conversion complete

Figure 3.11.2 (a) AD Converter Related Register

AD Mode Control Register 1

ADMOD1 (02B1H)

	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
Read/Write	R/W				R/W			
After reset	0	0			0	0	0	0
Function	VREF application control 0: OFF 1: ON	IDLE2 0: Stop 1: Operation			AD external trigger start control 0: Disable 1: Enable	Analog input channel selection		

Analog input channel selection

<ADCH2:0> \ <SCAN>	0 (Channel fixed)	1 (Channel scan)
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011 (Note)	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

AD conversion start control by external trigger ($\overline{\text{ADTRG}}$ input)

0	Disable
1	Enable

IDLE2 control

0	Stop
1	Operation

Control of application of reference voltage to AD converter

0	OFF
1	ON

Before starting conversion (before writing 1 to ADMOD0<ADS>), set the <VREFON> bit to 1.

Note: As pin AN3 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set <ADCH2:0> = 011 when using $\overline{\text{ADTRG}}$ with <ADTRGE> set to 1.

Figure 3.11.2 (b) AD Converter Related Register

AD Conversion Data Lower Register 0/4

ADREG04L (02A0H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR01	ADR00						ADR0RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result							AD conversion data storage flag 1: Conversion result stored

AD Conversion Data Upper Register 0/4

ADREG04H (02A1H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Data Lower Register 1/5

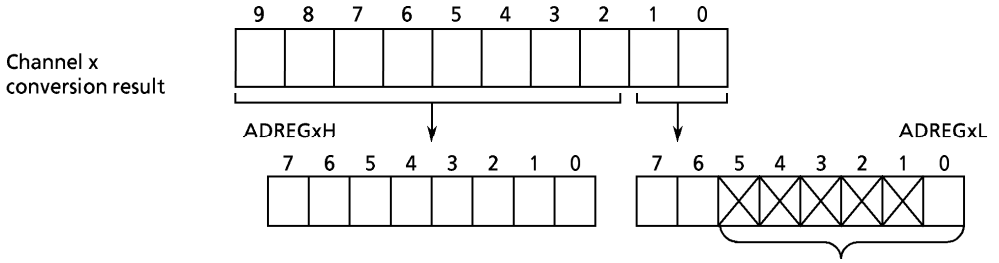
ADREG15L (02A2H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR11	ADR10						ADR1RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result							AD conversion result flag 1: Conversion result stored

AD Conversion Data Upper Register 1/5

ADREG15H (02A3H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5 to 1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.2 (c) AD Converter Related Registers

AD Conversion Result Lower Register 2/6

ADREG26L (02A4H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						ADR2RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result							AD conversion data storage flag 1: Conversion result stored

AD Conversion Data Upper Register 2/6

ADREG26H (02A5H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Data Lower Register 3/7

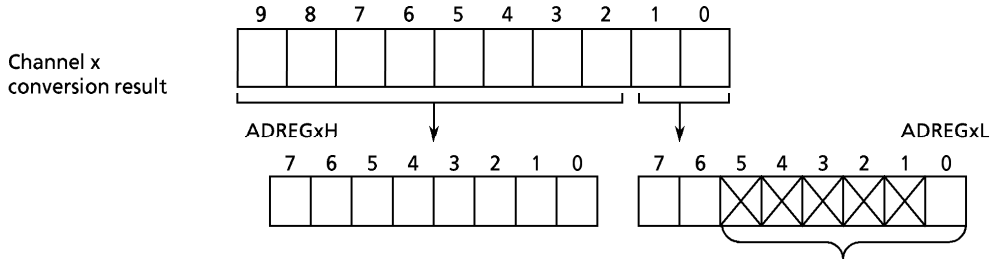
ADREG37L (02A6H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						ADR3RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result							AD conversion data storage flag 1: Conversion result stored

AD Conversion Result Upper Register 3/7

ADREG37H (02A7H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5 to 1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.2 (d) AD Converter Related Registers

3.11.2 Description of Operation

(1) Analog reference voltage

A high level analog reference voltage is applied to the VREFH pin; a low level analog reference voltage to the VREFL pin. To perform AD conversion, the reference voltage, the difference between VREFH and VREFL, is divided by 1024 using string resistance. Then, the result of the division is compared with the analog input voltage.
To turn off the switch between VREFH and VREFL, write 0 to AD mode control register 1 ADMOD1<VREFON>. To start AD conversion from the off state, first write 1 to <VREFON>, wait 3 μs until the internal reference voltage stabilizes (not related to the fc), then write 1 to AD mode register ADMOD0<ADS>.

(2) Analog input channel selection

The analog input channel selection varies according to the operating mode of the AD converter.

- In analog input channel fixed mode (ADMOD0<SCAN> = 0)
  Setting ADMOD1<ADCH2:0> selects one channel among analog input pins AN0 to AN7.
- In analog input channel scan mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH2:0> selects one scan mode from among eight scan modes.

Table 3.11.1 shows the analog input channel selection for each operating mode.

After a reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADCH2:0> is initialized to 000, thus selecting pin AN0 as the channel fixed input. Pins not used as analog input channels can be used as standard input ports.

Table 3.11.1 Analog Input Channel Selection

<ADCH2:0>	Channel fixed <SCAN> = 0	Channel scan <SCAN> = 1
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

(3) Starting AD conversion

To start AD conversion, write 1 to AD mode control register 0 ADMOD0<ADS> or AD mode control register 1 ADMOD1<ADTRGE> and input a falling edge on the $\overline{\text{ADTRG}}$ pin. When AD conversion starts, the AD conversion busy flag ADMOD0<ADBF> is set to 1, indicating AD conversion is in progress.
Writing 1 to <ADS> during AD conversion restarts conversion. At that time, to determine whether the AD conversion results are preserved, check the conversion data storage flag ADREGxL<ADRxRF>.
During AD conversion, inputting a falling edge to the $\overline{\text{ADTRG}}$ pin is ignored.

(4) AD conversion modes and AD conversion end interrupt

The four AD conversion modes are:

- Channel fixed single conversion mode
- Channel scan single conversion mode
- Channel fixed repeat conversion mode
- Channel scan repeat conversion mode

AD mode control register 0 ADMOD0<REPEAT>, <SCAN> selects the AD mode.
Completion of AD conversion triggers the AD conversion end INTAD interrupt request. Also, ADMOD0<EOCF> is set to 1 to indicate that AD conversion is complete.

① Channel fixed single conversion mode

Setting ADMOD0<REPEAT>, <SCAN> to 00 sets conversion channel fixed single conversion mode. In this mode, one specified channel is converted once only. When the conversion is complete, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

② Channel scan single conversion mode

Setting ADMOD0<REPEAT>, <SCAN> to 01 sets conversion channel scan single conversion mode. In this mode, the specified scan channels are converted once only. When scan conversion is complete, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

③ Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT>, <SCAN> to 10 sets conversion channel fixed repeat conversion mode. In this mode, one specified channel is converted repeatedly. When conversion is complete, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held at 1. The INTAD interrupt request generation timing is selected by ADMOD0<ITM0>.
Setting <ITM0> to 0 generates an interrupt request when every AD conversion completes.
Setting <ITM0> to 1 generates an interrupt request when every fourth conversion completes.

④ Channel scan repeat conversion mode

Setting ADMOD0<REPEAT>, <SCAN> to 11 sets conversion channel scan repeat conversion mode. In this mode, the specified scan channels are converted repeatedly. When each scan conversion completes, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held at 1.
To stop conversion in a repeat conversion mode (Mode ③ or ④), write 0 to ADMOD0<REPEAT>. After the current conversion is complete, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.
Switching to a halt state (IDLE2 when ADMOD1<I2AD> is cleared to 0, IDLE1, or STOP) immediately stops the AD converter even with AD conversion still in progress. In repeat conversion modes (Modes ③ and ④), after the halt is released, conversion restarts from the beginning. In single conversion modes (Modes ① and ②), conversion does not restart (The converter remains stopped).

Table 3.11.2 shows the relationship between AD conversion modes and interrupt requests.

Table 3.11.2 Relationship between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<SCAN>
Channel fixed single conversion mode	After completion of conversion	X	0	0
Channel scan single conversion mode	After completion of scan conversion	X	0	1
Channel fixed repeat conversion mode	Every conversion	0	1	0
	Every fourth conversion	1		
Channel scan repeat conversion mode	After completion of every scan conversion	X	1	1

X: Don't care

(5) AD conversion time

84 states (10.5 μs at f_{FPH} = 16MHz) are required for AD conversion of one channel.

(6) Storing and reading AD conversion result

The AD conversion data upper and lower registers (ADREG04H/L to ADREG37H/L) store the AD conversion results. (ADREG04H/L to ADRG37H/L are read-only registers.)
In channel fixed repeat conversion mode, the conversion results are stored successively in registers ADREG04H/L to ADREG37H/L. In other modes, the AN0 and AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7 conversion results are stored in ADREG04H/L, ADREG15H/L, ADREG26H/L, and ADREG37H/L respectively.
Table 3.11.3 shows the correspondence between analog input channels and AD conversion result registers.

Table 3.11.3 Correspondence between Analog Input Channels and AD Conversion Result Registers

Analog Input Channel (Port A)	AD Conversion Result Register	
	Conversion Modes Other than at Right	Channel Fixed Repeat Conversion Mode (Every 4th conversion)
AN0	ADREG04H/L	ADREG04H/L → ADREG15H/L → ADREG26H/L → ADREG37H/L → ADREG04H/L
AN1	ADREG15H/L	
AN2	ADREG26H/L	
AN3	ADREG37H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

The AD conversion data storage flag <ADRxRF> uses bit0 of the AD conversion data lower register. The storage flag indicates whether the AD conversion result register was read or not. When a conversion result is stored in the AD conversion result register the flag is set to 1. When either of the AD conversion result registers (ADREGxH or ADREGxL) is read the flag is cleared to 0.
Reading the AD conversion result also clears the AD conversion end flag ADMOD0<EOCF> to 0.

Setting example:

① Convert the analog input voltage at the AN3 pin and write the result, using the AD interrupt (INTAD) processing routine, to memory address 0800H.

Main routine setting:

```
                7 6 5 4 3 2 1 0
INTE0AD ← X 1 0 0 X - - -        Enable INTAD and set level to 4.
ADMOD1  ← 1 1 X X 0 0 1 1        Set analog input channel to pin AN3.
ADMOD0  ← X X 0 0 0 0 0 1        Start conversion in channel fixed single conversion mode.
```

Interrupt routine processing example:

```
WA      ← ADREG37                Read value of ADREG37L and ADREG37H to general-
                                 purpose register WA (16 bits).
WA      >> 6                     Shift contents read in WA six times to right and zero-fill
                                 upper bits.
(0800H) ← WA                     Write contents of WA to memory address 0800H.
```

② This example repeatedly converts the analog input voltages at the three pins AN0 to AN2, using channel scan repeat conversion mode.

```
INTE0AD ← X 0 0 0 X - - -        Disable INTAD.
ADMOD1  ← 1 X X X 0 0 1 0        Set pins AN0 to AN2 as analog input channels.
ADMOD0  ← X X 0 0 0 1 1 1        Start conversion in channel scan repeat conversion mode.
```

X: Don't care, —: No change

3.12 Watchdog Timer (Runaway Detecting Timer)

TMP91CW12 contain a watchdog timer of runaway detecting.
The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (Runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.
Connecting the watchdog timer output to the reset pin internally forces a reset.

3.12.1 Configuration

Figure 3.12.1 shows the block diagram of the watchdog timer (WDT).

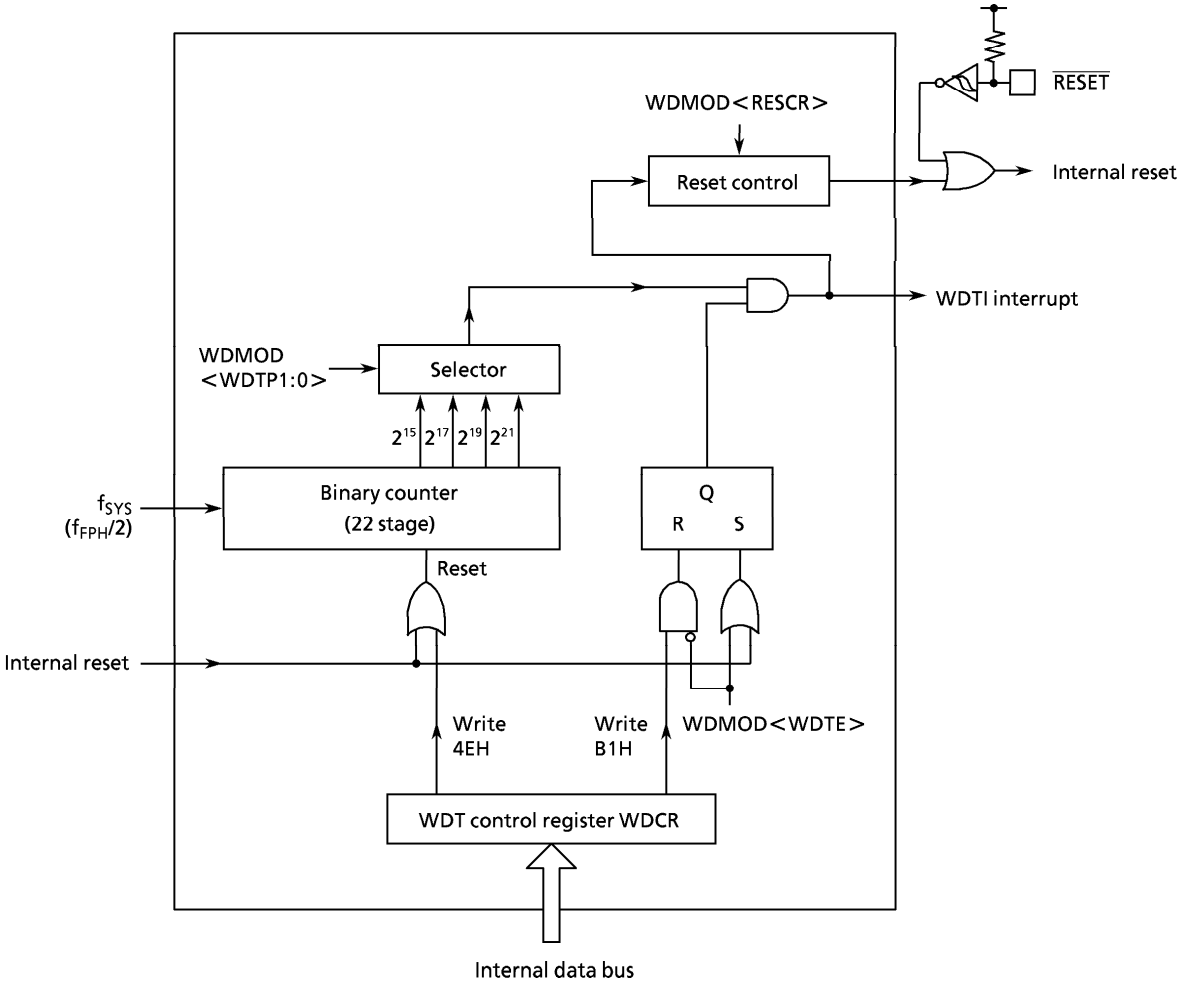


Figure 3.12.1 Block Diagram of Watchdog Timer

Note: Adequate care must be given when designing systems so as to eliminate disturbing noise. Otherwise the WDT may not exhibit its full functionality.

The watchdog timer consists of 22-stage binary counters which use system clock (f_{SYS}) as the input clock. The binary counter has $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$ output. Selecting one of the outputs with the WDMOD<WDTP1:0> register generates a watchdog interrupt and outputs watchdog timer out when an overflow occurs.

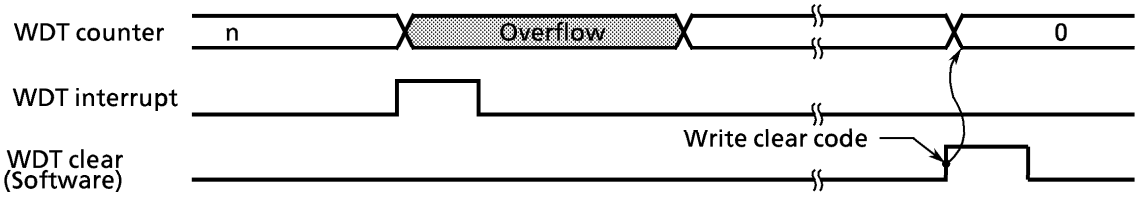


Figure 3.12.2 Normal Mode

The runaway detecting result can also be connected to the reset pin internally.
In this case, the reset time will be during 22 to 29 states shown by Figure 3.12.3.

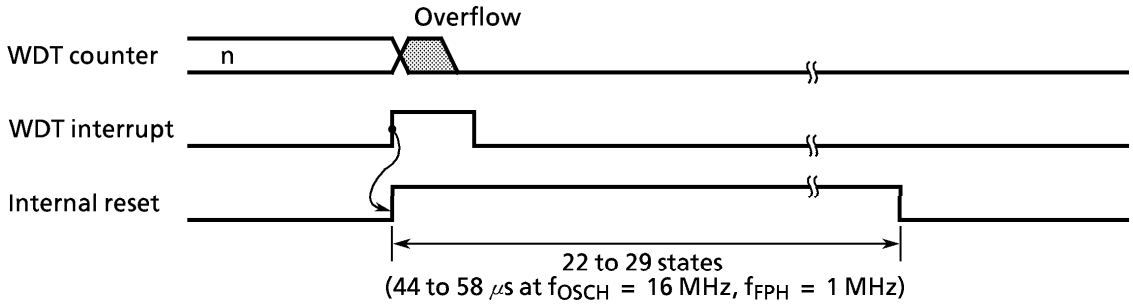


Figure 3.12.3 Reset Mode

3.12.2 Control Registers

Watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog timer mode register (WDMOD)

① Setting the detecting time of watchdog timer <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD<WDTP1:0> = 00 when reset.
The defecting time of WDT is shown Table 3.12.4.

② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD<WDTE> is initialized to 1 enable the watchdog timer.
To disable, it is necessary to set this bit to "0" and write the disable code (B1H) in the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.
However, it is possible to return from the disable state to enable state by merely setting <WDTE> to 1.

③ Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with $\overline{\text{RESET}}$ terminal, internally. Since WDMOD<RESCR>is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog timer control register (WDCR)

This register is used to disable and clear of binary counter the watchdog timer function.

- Disable control

By writting the disable code (B1H) in this WDCR register after clearing WDMOD<WDTE> to 0, the watchdog timer can be disabled.

```
WDMOD ← 0 - X X - - - 0      Clear WDMOD<WDTE>to 0.
WDCR  ← 1 0 1 1 0 0 0 1      Write the disable code (B1H).
```

- Enable control

Set WDMOD<WDTE>to 1.

- Watchdog timer clear control

The binary counter can be cleared and resume counting by writing clear code (4EH) into the WDCR register.

```
WDCR  ← 0 1 0 0 1 1 1 0      Write the clear code (4EH).
```

WDMOD (0300H)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	–
Read/Write	R/W	R/W				R/W		R/W
After reset	1	0	0			0	0	0
Function	WDT control 1: Enable	Select detecting time 00 : $2^{15}/f_{SYS}$ 01 : $2^{17}/f_{SYS}$ 10 : $2^{19}/f_{SYS}$ 11 : $2^{21}/f_{SYS}$				IDLE2 0: Stop 1: Operation	1: Internally connects WDT out to the reset pin	Allways write 0

Watchdog timer out control

0	–
1	Connects WDT out to a reset

IDLE2 control

0	Stop
1	Operation

Watchdog timer detecting time

at fc = 16 MHz, fs = 32.768 kHz

SYSCR1 System Clock Selection <SYSCK>	SYSCR1 Gear Value <GEAR2:0>	Watchdog Timer Detecting Time WDMOD<WDTP1:0> 00	01	10	11
1 (fs)	XXX	2.0 s	8.0 s	32.0 s	128.0 s
0 (fc)	000 (fc)	4.096 ms	16.384 ms	65.536 ms	262.144 ms
	001 (fc/2)	8.192 ms	32.768 ms	131.072 ms	524.288 ms
	010 (fc/4)	16.384 ms	65.536 ms	262.144 ms	1.049 s
	011 (fc/8)	32.768 ms	131.072 ms	524.288 ms	2.097 s
	100 (fc/16)	65.536 ms	262.144 ms	1.049 s	4.194 s

Watchdog timer enable/disable control

0	Disable
1	Enable

Figure 3.12.4 Watchdog Timer Mode Register

WDCR (0301H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

Disable/clear WDT

B1H	Disable code
4EH	Clear code
Others	Don't care

Figure 3.12.5 Watchdog Timer Control Register

3.12.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1:0>. The watchdog timer must be zero cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (Runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (Runaway) due to the INTWD interrupt and it is possible to return to normal operation by an anti-mulfunction program.
The watchdog timer does not operate in IDLE1 or STOP mode. The watchdog timer begins operating immediately on release of the watchdog timer reset.
As the binary counter continues counting during bus release (when $\overline{BUSAK}$ goes low).
When the IDLE2 mode is on, the operation of WDT is depended on setting WDMOD<I2WDT>. Set WDMOD<I2WDT> before IDLE2 mode is on.

Example: ① Clear the binary counter.

```
WDCR  ← 0 1 0 0 1 1 1 0        Write clear code (4EH).
```

② Set the watchdog timer detecting time to $2^{17}/f_{SYS}$.

```
WDMOD ← 1 0 1 X X - - 0
```

③ Disable the watchdog timer.

```
WDMOD ← 0 - - - - - - -        Clear WDTE to 0.
WDCR  ← 1 0 1 1 0 0 0 1        Write disable code (B1H).
```

3.13 Timer for Real Time Clock (RTC)

The TMP91CW12 includes a timer which is used for a clock operation.
An interrupt (INTRTC) can be generated each 0.0625 [s] or 0.125 [s] or 0.25 [s] or 0.50 [s] by using a low-frequency clock of 32.768 kHz. A clock function can be easily used.
A timer for real time clock can operate in all mode in which a low-frequency oscillation is operated.
In addition, INTRTC can return from each standby mode except STOP mode.

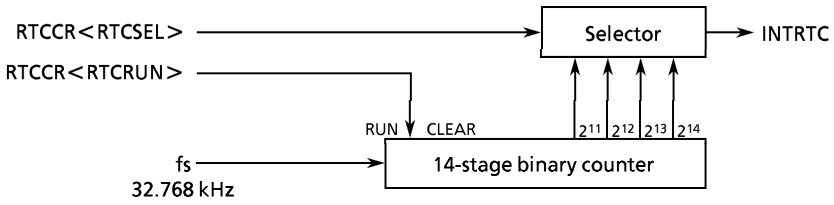


Figure 3.13.1 Block Diagram for Timer for Real Time Clock

The timer for real time clock is controlled by a timer for real time clock control register (RTCCR).

Figure 3.13.2 shows the timer for real time clock control register.

RTCCR (0310H)

	7	6	5	4	3	2	1	0
Bit symbol	–					RTCSEL1	RTCSEL0	RTCRUN
Read/Write	R/W					R/W		R/W
After reset	0					0	0	0
Function	Write 0					00 : 2^{14}/fs 01 : 2^{13}/fs 10 : 2^{12}/fs 11 : 2^{11}/fs		0: Stop & clear 1: Run

Counting operation

0	Stop & clear
1	Count

Interrupt generation cycle (fs = 32.768 kHz)

00	0.50 [s]
01	0.25 [s]
10	0.125 [s]
11	0.0625 [s]

Figure 3.13.2 Timer for Real Time Clock Control Register

4. Electrical Characteristics

4.1 Maximum Ratings

Parameter	Symbol	Rating	Unit
Power supply voltage	Vcc	– 0.5 to 6.5	V
Input voltage	V_{IN}	– 0.5 to Vcc + 0.5	
Output current	I_{OL}	2	mA
Output current	I_{OH}	– 2	
Output current (Total)	ΣI_{OL}	80	
Output current (Total)	ΣI_{OH}	– 80	
Power dissipation (Ta = 85 °C)	P_D	600	mW
Soldering temperature (10 s)	T_{SOLDER}	260	°C
Storage temperature	T_{STG}	– 65 to 150	
Operating temperature	T_{OPR}	– 40 to 85	

Note: The 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 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 maximum rating value will ever be exceeded.

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition		Min	Typ. (Note)	Max	Unit
Power supply voltage (AVcc = DVcc) (AVss = DVss = 0 V)		Vcc	fc = 2 to 16 MHz	fs = 30 to 34 kHz	2.7		5.5	V
			fc = 4 to 25 MHz		4.5			
Input low voltage	P00 to P17 (AD0 to AD15)	V_{IL}	Vcc < 4.5 V		– 0.3		0.6	V
			Vcc ≧ 4.5 V				0.8	
	P20 to PA7 (except P63)	V_{IL1}	Vcc = 2.7 to 5.5 V				0.3Vcc	
	$\overline{RESET}$, $\overline{NMI}$, P63 (INT0)	V_{IL2}					0.25Vcc	
	AM0, AM1	V_{IL3}					0.3	
	X1	V_{IL4}					0.2Vcc	
Input high voltage	P00 to P17 (AD0 to AD15)	V_{IH}	Vcc<4.5 V		2.0		Vcc + 0.3	
			Vcc≧4.5 V		2.2			
	P20 to PA7 (except P63)	V_{IH1}	Vcc = 2.7 to 5.5 V		0.7Vcc			
	$\overline{RESET}$, $\overline{NMI}$, P63 (INT0)	V_{IH2}			0.75Vcc			
	AM0, AM1	V_{IH3}			Vcc – 0.3			
	X1	V_{IH4}			0.8Vcc			
Output low voltage		V_{OL}	I_{OL} = 1.6 mA (Vcc = 2.7 to 5.5 V)				0.45	V
Output high voltage		V_{OH}	I_{OH} = – 400 μA (Vcc = 3.0 V ± 10%)		2.4			
			I_{OH} = – 400 μA (Vcc = 5.0V ± 10%)		4.2			

Note: Typical values are for Ta = 25°C and V_{CC} = 3.0 V unless otherwise noted.

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition		Min	Typ. (Note1)	Max	Unit
Input leakage current	I_{LI}	$0.0 \leqq V_{IN} \leqq Vcc$			0.02	± 5	μA
Output leakage current	I_{LO}	$0.2 \leqq V_{IN} \leqq Vcc - 0.2$			0.05	± 10	
Power down voltage (at STOP, RAM backup)	V_{STOP}	$V_{IL2} = 0.2$ Vcc, $V_{IH2} = 0.8$ Vcc		2.0		6.0	V
$\overline{RESET}$ pull-up resistor	R_{RST}	Vcc = 3 V ± 10%		100		400	kΩ
		Vcc = 5 V ± 10%		50		230	
Pin capacitance	C_{IO}	fc = 1 MHz				10	pF
Schmitt width $\overline{RESET}$, $\overline{NMI}$, INT0	V_{TH}			0.4	1.0		V
Programmable pull-up resistor	R_{KH}	Vcc = 3 V ± 10%		100		400	kΩ
		Vcc = 5 V ± 10%		50		230	
NORMAL (Note 2)	Icc	Vcc = 3 V ± 10% fc = 16 MHz			6.7	10.0	mA
IDLE2					2.4	4.0	
IDLE1					0.8	1.6	
NORMAL (Note 2)		Vcc = 5 V ± 10% fc = 25 MHz (Typ. : Vcc = 5.0 V)			20.5	35.0	
IDLE2					8.6	13.0	
IDLE1					3.5	7.0	
SLOW (Note 2)		Vcc = 3 V ± 10% fs = 32.768 kHz			16.0	35.0	μA
IDLE2					5.4	12.0	
IDLE1					3.0	8.0	
STOP		Ta ≦ 50°C	Vcc = 2.7 to 5.5 V			10	
		Ta ≦ 70°C			0.2	20	
		Ta ≦ 85°C				50	

Note 1: Typical values are for Ta = 25°C and V_{CC} = 3.0 V unless otherwise noted.

Note 2: I_{CC} measurement condition (NORMAL, SLOW):
All functions are operational; output pins are open and input pins are fixed.

4.3 AC Characteristics

(1) Vcc = 3.0 V ± 10 %

No.	Parameter	Symbol	Variable		16 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	t_{FPH}	62.5	31250	62.5		ns
2	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 26		5		ns
3	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 26		5		ns
4	ALE high width	t_{LL}	x – 52		10		ns
5	ALE fall → $\overline{RD}$/$\overline{WR}$ fall	t_{LC}	0.5x – 28		3		ns
6	$\overline{RD}$ rise → ALE rise	t_{CLR}	0.5x – 26		5		
7	$\overline{WR}$ rise → ALE rise	t_{CLW}	x – 26		36		ns
8	A0 to A15 valid → $\overline{RD}$/$\overline{WR}$ fall	t_{ACL}	x – 41		21		ns
9	A0 to A23 valid → $\overline{RD}$/$\overline{WR}$ fall	t_{ACH}	1.5x – 50		43		ns
10	$\overline{RD}$ rise→ A0 to A23 hold	t_{CAR}	0.5x – 31		0		
11	$\overline{WR}$ rise→ A0 to A23 hold	t_{CAW}	x – 31		31		ns
12	A0 to A15 valid→ D0 to D15 input	t_{ADL}		3.0x – 87		100	ns
13	A0 to A23 valid→ D0 to D15 input	t_{ADH}		3.5x – 98		120	ns
14	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.0x – 75		50	ns
15	$\overline{RD}$ low width	t_{RR}	2.0x – 40		85		ns
16	$\overline{RD}$ rise → D0 to D15 hold	t_{HR}	0		0		ns
17	$\overline{RD}$ rise → A0 to A15 output	t_{RAE}	x – 25		37		ns
18	$\overline{WR}$ low width	t_{WW}	1.5x – 55		39		ns
19	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	2.0x – 78		15		ns
20	$\overline{WR}$ rise →D0 to D15 hold	t_{WD}	x – 49		13		ns
21	A0 to A23 valid → $\overline{WAIT}$ input $\left(^{(1+N)\ WAIT}_{mode}\right)$	t_{AWH}		3.5x – 118		100	ns
22	A0 to A15 valid → $\overline{WAIT}$ input $\left(^{(1+N)\ WAIT}_{mode}\right)$	t_{AWL}		3.0x – 117		70	ns
23	$\overline{RD}$/$\overline{WR}$ fall →$\overline{WAIT}$ hold $\left(^{(1+N)\ WAIT}_{mode}\right)$	t_{CW}	2.0x + 0		125		ns
24	A0 to A23 valid → Port input	t_{APH}		3.5x – 168		50	ns
25	A0 to A23 valid → Port hold	t_{APH2}	3.5x		218		ns
26	A0 to A23 valid → Port valid	t_{AP}		3.5x + 100		319	ns

AC measuring conditions

- Output level: High 0.7 Vcc/Low 0.3 Vcc , CL = 50 pF
- Input level: High 0.9 Vcc/Low 0.1 Vcc

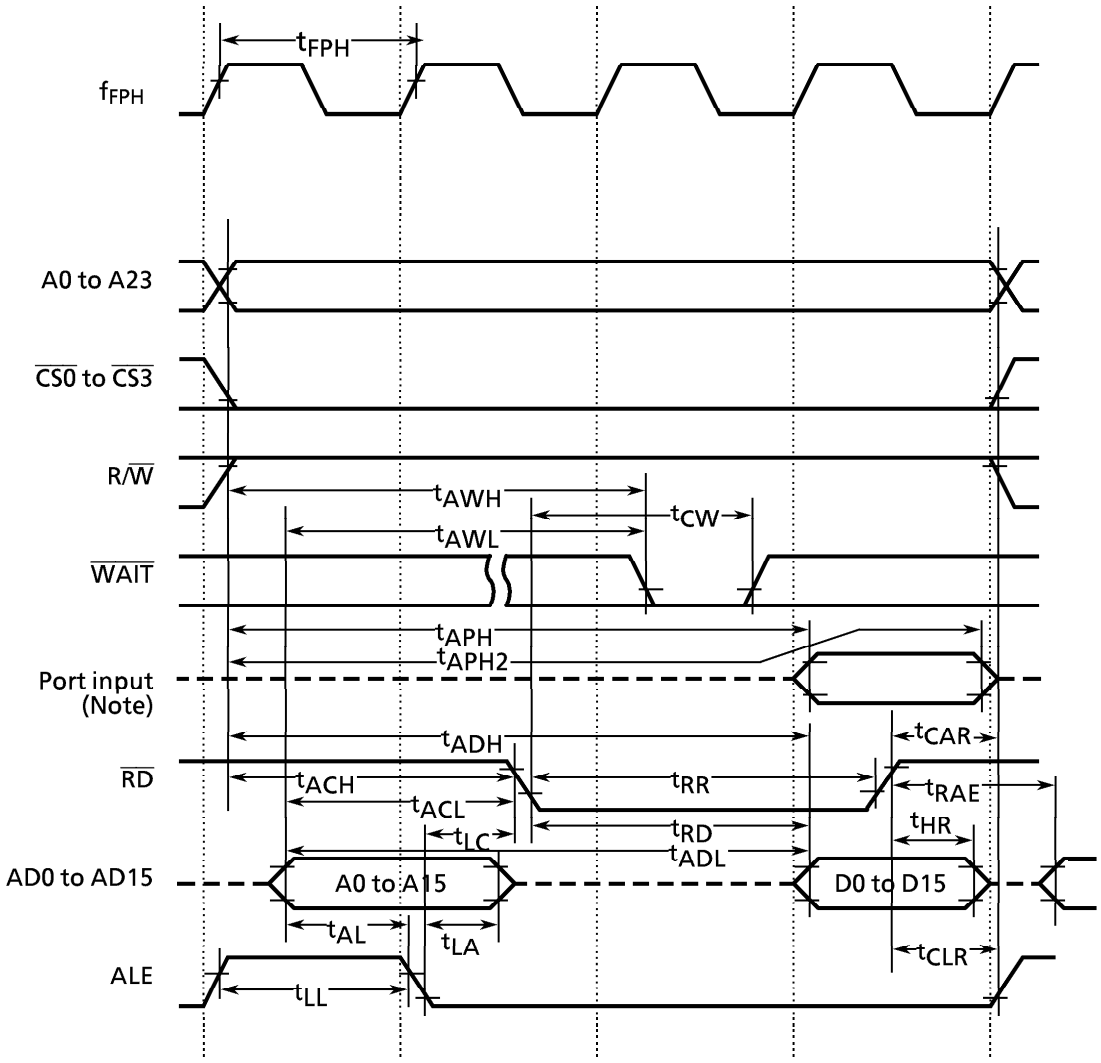
(2) Vcc = 5.0 V ± 10 %

No.	Parameter	Symbol	Variable		25 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	t_{FPH}	40	31250	40		ns
2	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 15		5		ns
3	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 15		5		ns
4	ALE high width	t_{LL}	x – 20		20		ns
5	ALE fall → $\overline{RD}/\overline{WR}$ fall	t_{LC}	0.5x – 20		0		ns
6	$\overline{RD}$ rise → ALE rise	t_{CLR}	0.5x – 15		5		
7	$\overline{WR}$ rise → ALE rise	t_{CLW}	x – 15		25		ns
8	A0 to A15 valid → $\overline{RD}/\overline{WR}$ fall	t_{ACL}	x – 25		15		ns
9	A0 to A23 valid → $\overline{RD}/\overline{WR}$ fall	t_{ACH}	1.5x – 50		10		ns
10	$\overline{RD}$ rise→ A0 to A23 hold	t_{CAR}	0.5x – 20		0		
11	$\overline{WR}$ rise→ A0 to A23 hold	t_{CAW}	x – 20		20		ns
12	A0 to A15 valid→ D0 to D15 input	t_{ADL}		3.0x – 45		75	ns
13	A0 to A23 valid→ D0 to D15 input	t_{ADH}		3.5x – 35		105	ns
14	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.0x – 40		40	ns
15	$\overline{RD}$ low width	t_{RR}	2.0x – 20		60		ns
16	$\overline{RD}$ rise → D0 to D15 hold	t_{HR}	0		0		ns
17	$\overline{RD}$ rise → A0 to A15 output	t_{RAE}	x – 15		25		ns
18	$\overline{WR}$ low width	t_{WW}	1.5x – 20		40		ns
19	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	1.5x – 50		10		ns
20	$\overline{WR}$ rise →D0 to D15 hold	t_{WD}	x – 15		25		ns
21	A0 to A23 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\text{ WAIT}\\ \text{mode}\end{smallmatrix}\right)$	t_{AWH}		3.5x – 90		50	ns
22	A0 to A15 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\text{ WAIT}\\ \text{mode}\end{smallmatrix}\right)$	t_{AWL}		3.0x – 80		40	ns
23	$\overline{RD}/\overline{WR}$ fall →$\overline{WAIT}$ hold $\left(\begin{smallmatrix}(1+N)\text{ WAIT}\\ \text{mode}\end{smallmatrix}\right)$	t_{CW}	2.0x + 0		80		ns
24	A0 to A23 valid → Port input	t_{APH}		3.5x – 120		20	ns
25	A0 to A23 valid → Port hold	t_{APH2}	3.5x		140		ns
26	A0 to A23 valid → Port valid	t_{AP}		3.5x + 100		319	ns

AC measuring conditions

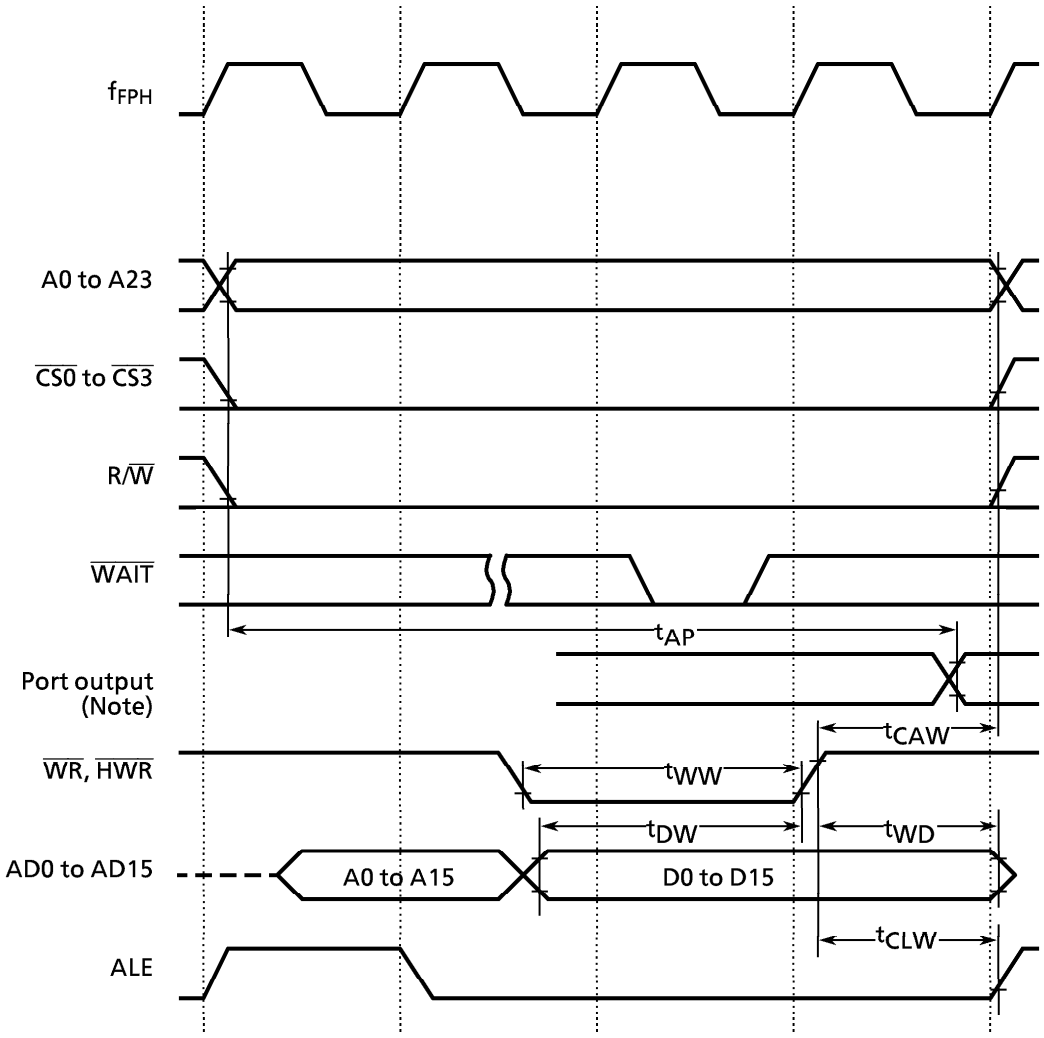
- Output level: High 2.2 V/Low 0.8 V, CL = 50 pF
- Input level: High 2.4 V/Low 0.45 V (AD0 to AD15)
  High 0.8 Vcc/Low 0.2 Vcc (except AD0 to AD15)

(1) Read cycle



Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{RD}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(2) Write cycle



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

4.4 AD Conversion Characteristics

$AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage (+)	VREFH	V_{CC} = 3 V ± 10%	V_{CC} – 0.2 V	V_{CC}	V_{CC}	V
		V_{CC} = 5 V ± 10%	V_{CC} – 1.5 V	V_{CC}	V_{CC}	
Analog reference voltage (–)	VREFL	V_{CC} = 3 V ± 10%	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
		V_{CC} = 5 V ± 10%	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
Analog input voltage range	VAIN		VREFL		VREFH	
Analog current for analog reference voltage <VREFON> = 1	IREF (VREFL = 0 V)	V_{CC} = 3 V ± 10%		0.85	1.20	mA
		V_{CC} = 5 V ± 10%		1.44	2.00	
<VREFON> = 0		V_{CC} = 2.7 to 5.5 V		0.02	5.0	μA
Error (not including quantizing errors)	–	V_{CC} = 3 V ± 10%		± 1.0	± 4.0	LSB
		V_{CC} = 5 V ± 10%		± 1.0	± 4.0	

Note 1: 1LSB = (VREFH – VREFL)/1024 [V]
Note 2: The operation above is guaranteed for $f_{FPH} \geqq 4$ MHz.
Note 3: The value I_{CC} includes the current which flows through the AVCC pin.

4.5 Serial Channel Timing (I/O internal mode)

(1) SCLK input mode

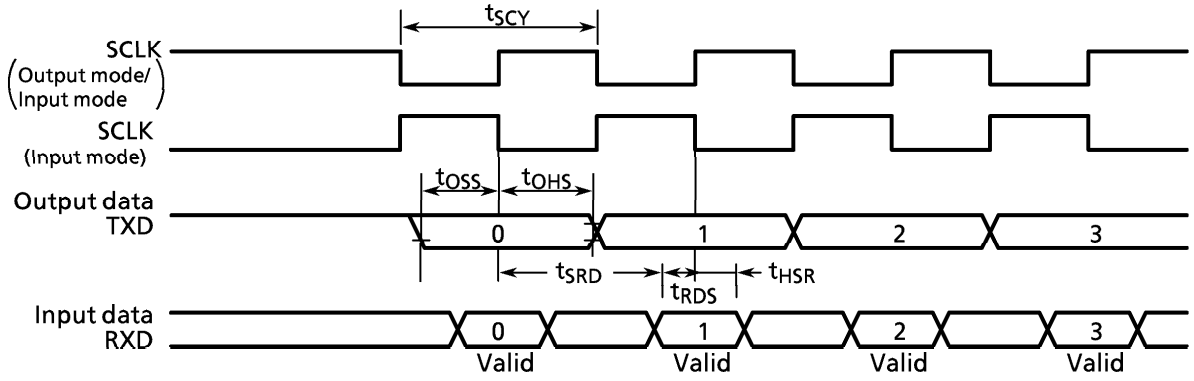
Parameter	Symbol	Variable		25 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period	t_{SCY}	16X		0.64		1.0		μs
Output data → SCLK rising/falling edge*	t_{OSS}	$t_{SCY}/2 - 4X - 85$ (V_{CC} = 5 V ± 10%)		75		165		ns
		$t_{SCY}/2 - 4X - 130$ (V_{CC} = 3 V ± 10%)		—		120		
SCLK rising/falling edge* → Output data hold	t_{OHS}	$t_{SCY}/2 + 2X + 0$		400		625		ns
SCLK rising/falling edge* → Input data hold	t_{HSR}	3X + 10		130		198		ns
SCLK rising/falling edge* → Valid data input	t_{SRD}		$t_{SCY} - 0$		640		1000	ns
Valid data input → SCLK rising/falling edge*	t_{RDS}	0		0		0		ns

*) SCLK rising/falling edge: The rising edge is used in SCLK rising mode.
The falling edge is used in SCLK falling mode.

Note: 25 MHz and 16 MHz values are calculated from $t_{SCY} = 16 \times$ Case.

(2) SCLK Output Mode

Parameter	Symbol	Variable		25 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period (Programable)	t_{SCY}	16X	8192X	0.64	327	1.0	512	μs
Output data → SCLK rising/falling edge	t_{OSS}	$t_{SCY}/2 - 40$		280		460		ns
SCLK rising/falling edge → Output data hold	t_{OHS}	$t_{SCY}/2 - 40$		280		460		ns
SCLK rising/falling edge → Input data hold	t_{HSR}	0		0		0		ns
SCLK rising/falling edge → Valid data input	t_{SRD}		$t_{SCY} - 1X - 90$		510		847	ns
Valid data input → SCLK rising/falling edge	t_{RDS}	1X + 90		130		153		ns



4.6 Event Counter (TA0IN, TA4IN, TB0IN0, TB0IN1, TB1IN0, TB1IN1)

Parameter	Symbol	Variable		25 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock period	t_{VCK}	8X + 100		420		600		ns
Clock low level width	t_{VCKL}	4X + 40		200		290		ns
Clock high level width	t_{VCKH}	4X + 40		200		290		ns

4.7 Interrupt, Capture

(1) $\overline{NMI}$, INT0 to INT4 interrupts

Parameter	Symbol	Variable		25 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 to INT4 low level width	t_{INTAL}	4X + 40		200		290		ns
$\overline{NMI}$, INT0 to INT4 high level width	t_{INTAH}	4X + 40		200		290		ns

(2) INT5 to INT8 interrupt, capture

The INT5 to INT8 input width depends on the system clock select mode, prescaler clock mode.

System Clock Selected <SYSCK>	Prescaler Clock Selected <PRCK1:0>	t_{INTBL} (INT5 to INT8 low level width)		t_{INTBH} (INT5 to INT8 high level width)		Unit
		Variable	25 MHz	Variable	25 MHz	
		Min	Min	Min	Min	
0 (fc)	00 (f_{FPH})	8X + 100	420	8X + 100	420	ns
	10 (fc/16)	128Xc + 0.1	5.22	128Xc + 0.1	5.22	μs
1 (fs)	00 (f_{FPH})	8X + 0.1	244.3	8X + 0.1	244.3	

Note: Xc = Period of clock fc

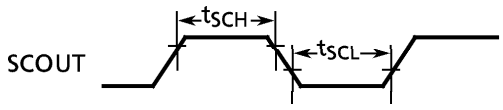
4.8 SCOUT pin AC characteristics

Parameter	Symbol	Condition	Variable		25 MHz		16 MHz		Unit
			Min	Max	Min	Max	Min	Max	
Low level width	t_{SCH}	V_{CC} = 3 V ± 10%	0.5T – 20		–		11		ns
		V_{CC} = 5 V ± 10%	0.5T – 15		5		16		
High level width	t_{SCL}	V_{CC} = 3 V ± 10%	0.5T – 20		–		11		ns
		V_{CC} = 5 V ± 10%	0.5T – 15		5		16		

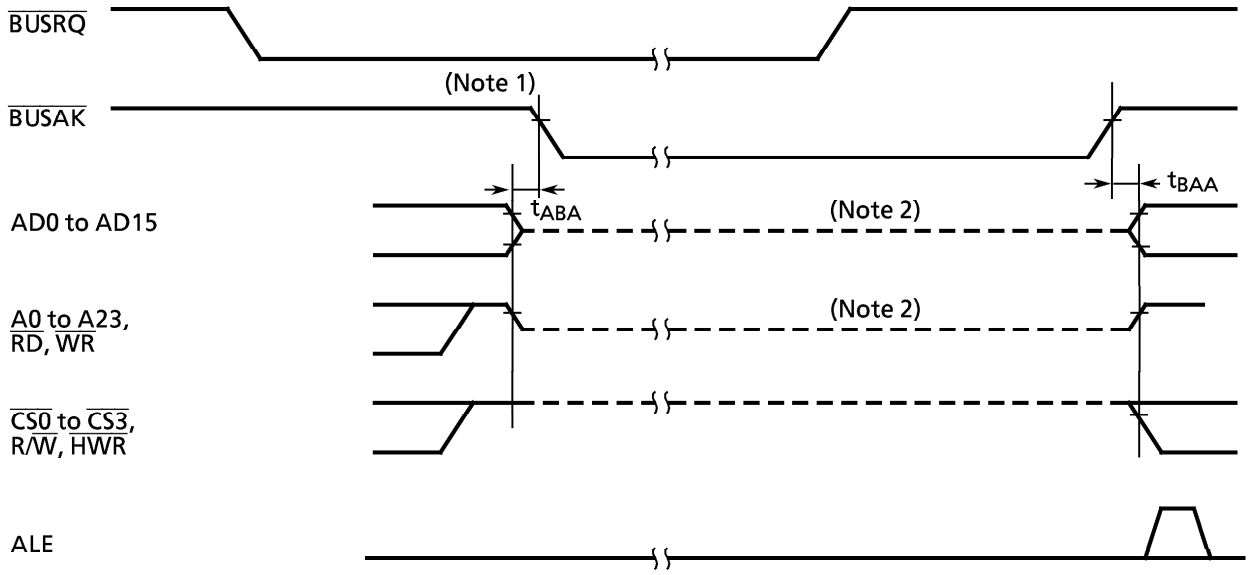
Note: T = Period of SCOUT

Measrement condition

- Output level: High 0.7 V_{CC}/low 0.3 V_{CC}, CL = 10 pF



4.9 Bus Request/Bus Acknowledge



Parameter	Symbol	Variable		25 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Output buffer off to $\overline{\text{BUSAK}}$ low	t_{ABA}	0	80	0	80	0	80	ns
$\overline{\text{BUSAK}}$ high to output buffer on	t_{BAA}	0	80	0	80	0	80	ns

Note 1: Even if the $\overline{\text{BUSRQ}}$ signal goes low, the bus will not be released while the $\overline{\text{WAIT}}$ signal is low. The bus will only be released when $\overline{\text{BUSRQ}}$ goes low while $\overline{\text{WAIT}}$ is high.

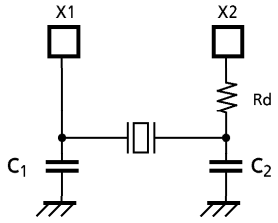
Note 2: This line shows only that the output buffer is in the off state.
It does not indicate that the signal level is fixed.
Just after the bus is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using an external resistor during bus release, careful design is necessary, as fixing of the level is delayed.
The internal programmable pull-up/pull-down resistor is switched between the active and non-active states by the internal signal.

4.10 Recommended Oscillation Circuit

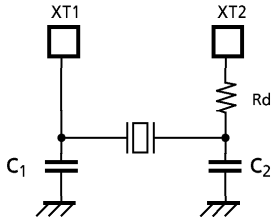
The TMP91CW12F/TMP91PW12F have been evaluated by the following resonator manufacturer. The evaluation results are shown below for your information.

Note: The load capacitance of the oscillation terminal is the sum of the load capacitances of C1 and C2 to be connected and the stray capacitance on the board. Even if the ratings of C1 and C2 are used, the load capacitance varies with each board and the oscillator may malfunction. Therefore, when designing a board, make the pattern around the oscillation circuit shortest. It is recommended that final evaluation of the resonator be performed on the board.

(1) Examples of resonator connection



High-frequency oscillator connection



Low-frequency oscillator connection

(2) Recommended ceramic resonators for the TMP91CW12F/PW12F: Murata Manufacturing Co., Ltd.

Item	Oscillation Frequency	Recommended Resonator	Recommended Rating			VCC [V]	Remarks
			C1 [pF]	C2 [pF]	Rd [kΩ]		
High-frequency oscillator	2.0	CSA2.00MG	30	30	0	2.7 to 3.3	–
		CST2.00MG	(30)	(30)			
	4.0	CSA4.00MG	30	30		2.7 to 5.5	
		CST4.00MGW	(30)	(30)			
	10.0	CSA10.0MTZ	30	30		4.5 to 5.5	
		CST10.0MTW	(30)	(30)			
		CSA10.0MTZ	30	30		2.7 to 3.3	TMP91CW12F only
		CST10.0MTW	(30)	(30)			
		CSA10.0MTZ093	30	30		2.7 to 3.3	TMP91PW12F only
		CST10.0MTW093	(30)	(30)			
	12.5	CSA12.5MTZ	30	30		4.5 to 5.5	–
		CST12.5MTW	(30)	(30)			
		CSA12.5MTZ	30	30		2.7 to 3.3	TMP91CW12F only
		CST12.5MTW	(30)	(30)			
		CSA12.5MTZ093	30	30		2.7 to 3.3	TMP91PW12F only
		CST12.5MTW093	(30)	(30)			
	16.0	CSA16.00MXZ040	5	5		4.5 to 5.5	–
		CST16.00MXW0C1	(5)	(5)			
		CSA16.00MXZ040	Open	Open		2.7 to 3.3	TMP91CW12F only
		CSA16.00MXZ046	Open	Open		2.7 to 3.3	TMP91PW12F only
	20.0	CSA20.00MXZ040	3	3		4.5 to 5.5	–
	25.0	CSA25.00MXZ040	Open	Open			

- The values enclosed in brackets in the C1 and C2 columns apply to the condenser built-in type.
- Murata Manufacturing Co., Ltd. (JAPAN)
  The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL:
  http://www.murata.co.jp/search/index.html

5. Table of SFR

The SFR (Special function register) includes the I/O ports and peripheral control registers allocated to the 4-Kbyte addresses from 000000H to 000FFFH.

(1) I/O port

(2) I/O port control

(3) Interrupt control

(4) Chip select/wait control

(5) Clock gear

(6) DFM (Clock doubler)

(7) 8-bit timer

(8) 16-bit timer

(9) UART/serial channel

(10) I2C bus/serial channel

(10) AD converter

(12) Watchdog timer

(13) RTC (Real time clock)

Configuration of the table

Symbol	Name	Address	7	6			1	0	
									→Bit symbol
									→Read/Write
									→Initial value after reset
									→Remarks

Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.
(Example) When setting only bit0 of register P0CR, "SET 0, (0002H)" cannot be used. The LD (Transfer) instruction must be used to write all eight bits.

Read/Write

R/W:	Both read and write is possible
R:	Only read is possible
W:	Only write is possible
W*:	Both read and write is possible (when this bit is read as 1)
Prohibit RMW:	Prohibit read-modify-write. (Prohibit EX, ADD, ADC, BUS, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD, RRD instruction.)
Prohibit RMW*:	Read-modify-write is prohibited when controlling the pull-up resistor.

Table 5. Address Map of SFR

[1] Port

ADDRESS	NAME
0000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	P2
7H	P3
8H	P2CR
9H	P2FC
AH	P3CR
BH	P3FC
CH	P4
DH	P5
EH	P4CR
FH	P4FC

ADDRESS	NAME
0010H	
1H	
2H	P6
3H	P7
4H	P6CR
5H	P6FC
6H	P7CR
7H	P7FC
8H	P8
9H	P9
AH	P8CR
BH	P8FC
CH	P9CR
DH	P9FC
EH	PA
FH	

ADDRESS	NAME
0020H	PACR
1H	PAFC
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	ODE

[2] INTC

ADDRESS	NAME
0080H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	
5H	
6H	
7H	
8H	INTCLR
9H	DMAR
AH	DMAB
BH	
CH	IIMC
DH	
EH	
FH	

ADDRESS	NAME
0090H	INTE0AD
1H	INTE12
2H	INTE34
3H	INTE56
4H	INTE78
5H	INTETA01
6H	INTETA23
7H	INTETA45
8H	INTETA67
9H	INTETB0
AH	INTETB1
BH	INTETB01V
CH	INTES0
DH	INTES1
EH	INTES2RTC
FH	

ADDRESS	NAME
00A0H	INTETC01
1H	INTETC23
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] CS/WAIT

ADDRESS	NAME
00C0H	B0CS
1H	B1CS
2H	B2CS
3H	B3CS
4H	
5H	
6H	
7H	BEXCS
8H	MSAR0
9H	MAMR0
AH	MSAR1
BH	MAMR1
CH	MSAR2
DH	MAMR2
EH	MSAR3
FH	MAMR3

Note: Do not access to the no-name address which is not allocated.

[4] CGEAR, DFM

ADDRESS	NAME
00E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	EMCCR0
4H	EMCCR1
5H	
6H	
7H	
8H	DFMCR0
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADDRESS	NAME
00F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] TMRA

ADDRESS	NAME
0100H	TA01RUN
1H	
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA1FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

ADDRESS	NAME
0110H	TA45RUN
1H	
2H	TA4REG
3H	TA5REG
4H	TA45MOD
5H	TA5FFCR
6H	
7H	
8H	TA67RUN
9H	
AH	TA6REG
BH	TA7REG
CH	TA67MOD
DH	TA7FFCR
EH	
FH	

[6] TMRB

ADDRESS	NAME
0180H	TB0RUN
1H	
2H	TB0MOD
3H	TB0FFCR
4H	
5H	
6H	
7H	
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADDRESS	NAME
0190H	TB1RUN
1H	
2H	TB1MOD
3H	TB1FFCR
4H	
5H	
6H	
7H	
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

[7] UART/SIO

ADDRESS	NAME
0200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

[8] I2C bus/SIO

ADDRESS	NAME
0240H	SBI0CR1
1H	SBI0DBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] 10-bit ADC

ADDRESS	NAME
02A0H	ADREG04L
1H	ADREG04H
2H	ADREG15L
3H	ADREG15H
4H	ADREG26L
5H	ADREG26H
6H	ADREG37L
7H	ADREG37H
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADDRESS	NAME
02B0H	ADMOD0
1H	ADMOD1
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] WDT

ADDRESS	NAME
0300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[11] RTC

ADDRESS	NAME
0310H	RTCCR
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

(1) I/O port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Data from external port (Output latch register is undefined)							
P1	Port 1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Data from external port (Output latch register is set to "0")							
P2	Port 2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Data from external port (Output latch register is set to "1")							
P3	Port 3	07H (Prohibit RMW)	P37	P36	P35	P34	P33	P32	P31	P30
			*R/W							
			Data from external port (Output latch register is set to "1")						1	1
			0: Pull-up resistor OFF 1: Pull-up resistor ON						—	—
P4	Port 4	0CH (Prohibit RMW)					P43	P42	P41	P40
							*R/W			
							Data from external port (Output latch register is set to "1")			
							0: Pull-up resistor OFF 1: Pull-up resistor ON			
P5	Port 5	0DH	P57	P56	P55	P54	P53	P52	P51	P50
			R							
			Data from external port							
P6	Port 6	12H		P66	P65	P64	P63	P62	P61	P60
				R/W						
				Data from external port (Output latch register is set to "1")						
P7	Port 7	13H			P75	P74	P73	P72	P71	P70
					R/W					
					Data from external port (Output latch register is set to "1")					
P8	Port 8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			Data from external port (Output latch register is set to "1")							
P9	Port 9	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W	R/W	R/W					
			1	1	Data from external port (Output latch register is set to "1")					
PA	Port A	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			Data from external port (Output latch register is set to "1")							

(2) I/O port control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	Port 0 control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1CR	Port 1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1FC	Port 1 function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00: Input, 01: Output, 10: AD8 to AD15, 11: A8 to A15							
P2CR	Port 2 control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			0 : Input 1 : Output							
P2FC	Port 2 function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00: Input, 01: Output, 10: A0 to A7, 11: A16 to A23							
P3CR	Port 3 control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C		
				W						
			0	0	0	0	0	0		
				0: Input 1: Output						
P3FC	Port 3 function	0BH (Prohibit RMW)	—	P36F	P35F	P34F		P32F	P31F	P30F
			W					W		
			0	0	0	0		0	0	0
			Always write 0	0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$
P4CR	Port 4 control	0EH (Prohibit RMW)					P43C	P42C	P41C	P40C
							W			
							0	0	0	0
							0: Input 1: Output			
P4FC	Port 4 function	0FH (Prohibit RMW)					P43F	P42F	P41F	P40F
							W			
							0	0	0	0
							0: Port 1: CS3	0: Port 1: CS2	0: Port 1: CS1	0: Port 1: CS0

Note: When P30 pin is defined as $\overline{RD}$ signal output mode (P30F = 1), clearing the output latch register P30 to 0 outputs the $\overline{RD}$ strobe from P30 pin for PSRAM, even when the internal address is accessed. If the output latch register P30 remains 1, the $\overline{RD}$ strobe is output only when the external address is accessed.

I/O port control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	Port 6 control	14H (Prohibit RMW)		P66C	P65C	P64C	P63C	P62C	P61C	P60C
				W						
				0	0	0	0	0	0	0
				0: Input 1: Output						
P6FC	Port 6 function	15H (Prohibit RMW)				P64F	P63F	P62F	P61F	P60F
						W	W	W	W	W
						0	0	0	0	0
						0: Port 1: SCOUT	0: Port 1: INT0	0: Port 1: SCL/SI	0: Port 1: SDA/SO	0: Port 1: SCK output
P7CR	Port 7 control	16H (Prohibit RMW)			P75C	P74C	P73C	P72C	P71C	P70C
					W					
					0	0	0	0	0	0
					0: Input 1: Output					
P7FC	Port 7 function	17H (Prohibit RMW)			P75F	P74F		P72F	P71F	
					W	W		W	W	
					0	0		0	0	
					0: Port 1: TA7OUT	0: Port 1: TA5OUT		0: Port 1: TA3OUT	0: Port 1: TA1OUT	
P8CR	Port 8 control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P8FC	Port 8 function	1BH (Prohibit RMW)	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
			W							
			0	0	0	0	0	0	0	0
			0: Port 1: TB1OUT1	0: Port 1: TB1OUT0	0: Port 1:INT8/TB1IN1	0: Port 1:INT7/TB1IN0	0: Port 1: TB0OUT1	0: Port 1: TB0OUT0	0: Port 1:INT6/TB0IN0	0: Port 1:INT5/TB0IN0
P9CR	Port 9 control	1CH (Prohibit RMW)	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W	W	W					
			1	1	0	0	0	0	0	0
			0 : Input 1 : Output							
P9FC	Port 9 function	1DH (Prohibit RMW)			P95F		P93F	P92F		P90F
					W		W	W		W
					0		0	0		0
					0 : Port 1 : SCLK1		0 : Port 1 : TXD1	0 : Port 1 : SCLK0		0 : Port 1 : TXD0
PACR	Port A control	20H (Prohibit RMW)	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
PAFC	Port A function	21H (Prohibit RMW)	–	–	–	–	PA3F	PA2F	PA1F	PA0F
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Write "0"				INT4 to INT1 Input enable			
ODE	Serial open-drain enable	2FH					ODE62	ODE61	ODE93	ODE90
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: P62ODE	1: P61ODE	1: P93ODE	1: P90ODE

Note: 1. External interrupt INT0
The input enable is set by P6FC<P63F>. The selection either level or edge and selection either rise or fall edge are set by IIMC<I0LE, I0EDGE>.

2. External interrupt INT1 to INT4
The input enable is set by PAFC<PA3F to PA0F>. The selection either rise or fall edge is set by IIMC<I4EDGE to I1EDGE>.

3. External interrupt INT5 to INT8
The input enable is set by P8FC<P85F, P84F, P81F, P80F>. The setting of edge is set by TB0MOD, TB1MOD.

(3) Interrupt control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	Interrupt enable 0 & AD	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTAD	Interrupt level			1: INT0	Interrupt level		
INTE12	Interrupt enable 2/1	91H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT2	Interrupt level			1: INT1	Interrupt level		
INTE34	Interrupt enable 4/3	92H	INT4				INT3			
			I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT4	Interrupt level			1: INT3	Interrupt level		
INTE56	Interrupt enable 6/5	93H	INT6				INT5			
			I6C	I6M2	I6M1	I6M0	I5C	I5M2	I3M1	I5M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT6	Interrupt level			1: INT5	Interrupt level		
INTE78	Interrupt enable 8/7	94H	INT8				INT7			
			I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT8	Interrupt level			1: INT7	Interrupt level		
INTETA01	Interrupt enable timer A 1/0	95H	INTTA1 (TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA1	Interrupt level			1: INTTA0	Interrupt level		
INTETA23	Interrupt enable timer A 3/2	96H	INTTA3 (TMRA3)				INTTA2 (TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA3	Interrupt level			1: INTTA2	Interrupt level		
INTETA45	Interrupt enable timer A 5/4	97H	INTTA5 (TMRA5)				INTTA4 (TMRA4)			
			ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA5	Interrupt level			1: INTTA4	Interrupt level		
INTETA67	Interrupt enable timer A 7/6	98H	INTTA7 (TMRA7)				INTTA6 (TMRA6)			
			ITA7C	ITA7M2	ITA7M1	ITA7M0	ITA6C	ITA6M2	ITA6M1	ITA6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA7	Interrupt level			1: INTTA6	Interrupt level		

Interrupt control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTETB0	Interrupt enable TMRB0	99H	INTTB01 (TMRB0)				INTTB00 (TMRB0)			
			ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTB01	Interrupt level			1: INTTB00	Interrupt level		
INTETB1	Interrupt enable TMRB1	9AH	INTTB11 (TMRB1)				INTTB10 (TMRB1)			
			ITB11C	ITB11M2	ITB11M1	ITB11M0	ITB10C	ITB10M2	ITB10M1	ITB10M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTB11	Interrupt level			1: INTTB10	Interrupt level		
INTETB01V	Interrupt enable TMRB0/1 (Overflow)	9BH	INTTBOF1 (TMRB1 Overflow)				INTTBOF0 (TMRB0 Overflow)			
			ITF1C	ITF1M2	ITF1M1	ITF1M0	ITF0C	ITF0M2	ITF0M1	ITF0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTBOF1	Interrupt level			1: INTTBOF0	Interrupt level		
INTES0	Interrupt enable serial 0	9CH	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX0	Interrupt level			1: INTRX0	Interrupt level		
INTES1	Interrupt enable serial 1	9DH	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX1	Interrupt level			1: INTSBI	Interrupt level		
INTES2RTC	Interrupt enable SBI /RTC	9EH	INTRTC				INTSBI			
			IRTCC	IRTCM2	IRTCM1	IRTCM0	ISBIC	ISBIM2	ISBIM1	ISBIM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTRTC	Interrupt level			1: INTSBI	Interrupt level		
INTETC01	Interrupt enable TC0/1	A0H	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	Interrupt enable TC2/3	A1H	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Interrupt control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request vector	80H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 Start vector					
DMA1V	DMA 1 request vector	81H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 Start vector					
DMA2V	DMA 2 request vector	82H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 Start vector					
DMA3V	DMA 3 request vector	83H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 Start vector					
INTCLR	Interrupt clear control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					0	0	0	0	0	0
					Clears interrupt request flag by writing DMA start vector					
DMAR	DMA software request register	89H					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA request by software			
DMAB	DMA burst request register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA request on burst-mode			
IIMC	Interrupt input mode contorol	8CH (Prohibit RMW)	–	I4EDGE	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Always write "0"	INT4 edge 0: Rising 1: Falling	INT3 edge 0: Rising 1: Falling	INT2 edge 0: Rising 1: Falling	INT1 edge 0: Rising 1: Falling	INT0 edge 0: Rising 1: Falling	INT0 0: edge 1: Level	1 : Operation even at $\overline{\text{NMI}}$ rising edge

(4) Chip Select/Wait Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	C0H (Prohibit RMW)	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 1xx: Reserved 001: 1 wait 010: (1 + N) waits 011: 0 waits		
B1CS	Block 1 CS/WAIT control register	C1H (Prohibit RMW)	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 1xx: Reserved 001: 1 wait 010: (1 + N) waits 011: 0 waits		
B2CS	Block 2 CS/WAIT control register	C2H (Prohibit RMW)	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
			W	W	W	W	W	W	W	W
			1	0	0	0	0	0	0	0
			0: Disable 1: Enable	0: 16 MB area 1: Address specification area	00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 1xx: Reserved 001: 1 wait 010: (1 + N) waits 011: 0 waits		
B3CS	Block 3 CS/WAIT control register	C3H (Prohibit RMW)	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 1xx: Reserved 001: 1 wait 010: (1 + N) waits 011: 0 waits		
BEXCS	External CS/WAIT control register	C7H (Prohibit RMW)					BEXBUS	BEXW2	BEXW1	BEXW0
							W	W	W	W
							0	0	0	0
							Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 1xx: Reserved 001: 1 wait 010: (1 + N) waits 011: 0 waits		
MSAR0	Memory start address register 0	C8H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR0	Memory address mask register 0	C9H	V20	V19	V18	V17	V16	V15	V14 to V9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0 : Enable to compare address							
MSAR1	Memory start address register 1	CAH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR1	Memory Address mask register 1	CBH	V21	V20	V19	V18	V17	V16	V15 to V9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS1 area size 0: Enable to compare address							

Chip select/wait control (2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MSAR2	Memory start address register 2	CCH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR2	Memory address mask register 2	CDH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS2 area size 0: Enable to compare address							
MSAR3	Memory start address register 3	CEH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR3	Memory address mask register 3	CFH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS3 area size 0: Enable to compare address							

(5) Clock gear

Symbol	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System clock control register 0	E0H	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation	High-frequency oscillator (fc) after released STOP mode 0: Stop 1: Oscillation	Low-frequency oscillator (fs) after released STOP mode 0: Stop 1: Oscillation	Slect clock after released STOP mode 0: fc 1: fs	Warm-up timer 0 Write: don't care 1 Write: Start timer 0 Read: End warm-up 1 Read: Not end warm-up	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	
SYSCR1	System clock control register 1	E1H					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							Select system clock 0: fc 1: fs (Note 2)	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		
SYSCR2	System clock control register 2	E2H		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
				R/W	R/W	R/W	R/W	R/W		R/W
				0	1	0	1	1		0
				0: fs 1: f_{FPH}	Warm-up time 00: Reserved 01: 2^8/inputted frequency 10: 2^{14} 11: 2^{16}		00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			1 : Drive the pin in STOP mode
EMCCR0	EMC control register 0	E3H	PROTECT	–	–	–	ALEEN	EXTIN	DRVOSCH	DRVOSCL
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	1	0	0	0	1	1
			Protect flag 0 : OFF 1 : ON	Always write 0	Always write 1	Always write 0	1 : ALE output	1 : fc is external clock	fc oscillator drivability 1 : Normal 0 : Weak	fs oscillator drivability 1 : Normal 0 : Weak
EMCCR1	EMC control register 1	E4H	The protect is OFF by writing 1FH The protect is ON by writing except 1FH							

Note: EMCCR1

If the protect is ON, the write operation to the following SFR can not be possible.

1. CS/WAIT control
   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0, MSAR1, MSAR2, MSAR3,
   MAMR0, MAMR1, MAMR2, MAMR3
2. Clock gear (Only EMCCR1 is possible)
   SYSCR0, SYSCR1, SYSCR2, EMCCR0
3. DFM
   DFMCR0

(6) DFM (Clock doubler)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DFMCR0	DFM control register 0	E8H	ACT1	ACT0	DLUPFG	DLUPTM				
			R/W	R/W	R	R/W				
			0	0	0	0				

	DFM	LUP	f_{FPH}	Lockup flag	Lockup time
00	STOP	STOP	f_{OSCH}	0: End LUP	0: $2^{12}/f_{OSCH}$
01	RUN	RUN	f_{OSCH}	1: Not end LUP	1: $2^{10}/f_{OSCH}$
10	RUN	STOP	f_{DFM}		
11	RUN	STOP	f_{OSCH}		

(7) 8-bit timer (1/2)

(7-1) TMRA01

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01RUN	Timer RUN	100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TA0REG	8-bit timer register 0	102H (Prohibit RMW)	–							
			W							
			Undefined							
TA1REG	8-bit timer register 1	103H (Prohibit RMW)	–							
			W							
			Undefined							
TA01MOD	8-bit timer source CLK & MODE	104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00 : Reserved 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TA0IN pin 01: ϕT1 10: ϕT4 11: ϕT16	
TA1FFCR	8-bit timer flip-flop control	105H (Prohibit RMW)					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA1FF 01: SET TA1FF 10: Clear TA1FF 11: Don't care		1: TA1FF Invert Enable	0: TMRA0 1: TMRA1 inversion

(7-2) TMRA23

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA23RUN	Timer RUN	108H	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double Buffer 0 : Disable 1 : Enable				IDLE2 0 : Stop 1 : Operate	8-bit timer run/stop control 0 : Stop and clear 1 : Run (Count up)		
TA2REG	8-bit timer register 0	10AH (Prohibit RMW)	–							
			W							
			Undefined							
TA3REG	8-bit timer register 1	10BH (Prohibit RMW)	–							
			W							
			Undefined							
TA23MOD	8-bit timer source CLK & MODE	10CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: Reserved 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TA3FFCR	8-bit timer flip-flop control	10DH (Prohibit RMW)					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA3FF 01: SET TA3FF 10: Clear TA3FF 11: Don't care		1: TA3FF Invert Enable	0: TMRA2 1: TMRA3 inversion

8-bit timer (2/2)

(7-3) TMRA45

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA45RUN	Timer RUN	110H	TA4RDE				I2TA45	TA45PRUN	TA5RUN	TA4RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0 : Stop and clear 1 : Run (Count up)		
TA4REG	8-bit timer register 0	112H (Prohibit RMW)	–							
			W							
			Undefined							
TA5REG	8-bit timer register 1	113H (Prohibit RMW)	–							
			W							
			Undefined							
TA45MOD	8-bit timer source CLK & MODE	114H	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: Reserved 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA4TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TA4IN pin 01: ϕT1 10: ϕT4 11: ϕT16	
TA5FFCR	8-bit timer flip-flop control	115H (Prohibit RMW)					TA5FFC1	TA5FFC0	TA5FFIE	TA5FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA5FF 01: SET TA5FF 10: Clear TA5FF 11: Don't care		1: TA5FF Invert enable	0: Timer 4 1: Timer 5 inversion

(7-4) TMRA67

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA67RUN	Timer RUN	118H	TA6RDE				I2TA67	TA67PRUN	TA7RUN	TA6RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TA6REG	8-bit timer register 0	11AH (Prohibit RMW)	–							
			W							
			Undefined							
TA7REG	8-bit timer register 1	11BH (Prohibit RMW)	–							
			W							
			Undefined							
TA67MOD	8-bit timer source CLK & MODE	11CH	TA67M1	TA67M0	PWM61	PWM60	TA7CLK1	TA7CLK0	TA6CLK1	TA6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: – 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA6TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TA7FFCR	8-bit timer flip-flop control	11DH (Prohibit RMW)					TA7FFC1	TA7FFC0	TA7FFIE	TA7FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA7FF 01: SET TA7FF 10: Clear TA7FF 11: Don't care		1: TA7FF Invert enable	0: Timer 6 1: Timer 7 inversion

(8) 16-bit timer (1/2)

(8-1) TMRB0

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB0RUN	Timer control	180H	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write 0			IDLE2 0: Stop 1: Operate	16-bit timer run/stop control 0 : Stop and clear 1 : Run (Count up)		
TB0MOD	16-bit timer source CLK & MODE	182H (Prohibit RMW)	TB0CT1	TB0ET1	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			TB0FF1 0: TRG disable 1: TRG enable		0: Soft-capture 1: Un-defined	Capture timming (TB0IN0, TB0IN1) 00: Disable 01: ↑, ↑ 10: ↑, ↓ 11: ↑, ↓ (TA1OUT)		1: UC0 clear enable	Source clock 00: TB0IN0 pin 01: ϕT1 10: ϕT4 11: ϕT16	
			When CAP1	When match with TREG1						
TB0FFCR	16-bit timer flip-flop control	183H (Prohibit RMW)	TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Control TB0FF1 00: Invert 01: Set 10: Clear 11: Don't care ※ Always read as 11.		TB0FF0 Inversion trigger 0: Disable trigger 1: Enable trigger				Control TB0FF0 00: Invert 01: Set 10: Clear 11: Don't care ※ Always read as 11.	
					Invert when the UC value is loaded to TB0CP1.	Invert when the UC value is loaded to TB0CP0.	Invert when the UC value is matches to TB0RG1.	Invert when the UC value is matches to TB0RG0.		
TB0RG0L	16-bit timer register 0L	188H (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG0H	16-bit timer register 0H	189H (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG1L	16-bit timer register 1L	18AH (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG1H	16-bit timer register 1H	18BH (Prohibit RMW)	–							
			W							
			Undefined							
TB0CP0L	Capture register 0L	18CH	–							
			R							
			Undefined							
TB0CP0H	Capture register 0H	18DH	–							
			R							
			Undefined							
TB0CP1L	Capture register 1L	18EH	–							
			R							
			Undefined							
TB0CP1H	Capture register 1H	18FH	–							
			R							
			Undefined							

16-bit timer (2)

(8-2) TMRB1

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB1RUN	Timer control	190H	TB1RDE	–			I2TB1	TB1PRUN		TB1RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double buffer 0: Disable 1: Enable	Always write 0			IDLE2 0: Stop 1: Operate	16-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TB1MOD	16-bit timer source CLK & MODE	192H (Prohibit RMW)	TB1CT1	TB1ET1	TB1CP0I	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			TB1FF1 INV TRG 0: TRG disable 1: TRG enable		0: Soft-capture 1: Un-defined	Capture timming (TB1IN0, TB1IN1) 00 : Disable 01 : ↑, ↑ 10 : ↑, ↓ 11 : ↑, ↓ (TA1OUT)		1: UC0 clear enable	Source clock 00 : TB1IN0 pin 01 : ϕT1 10 : ϕT4 11 : ϕT16	
			When CAP1	When match with TREG1						
TB1FFCR	16-bit timer flip-flop control	193H (Prohibit RMW)	TB1FF1C1	TB1FF1C0	TB1C1T1	TB1C0T1	TB0E1T1	TB1E0T1	TB1FF0C1	TB1FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Control TB1FF1 00: Invert 01: Set 10: Clear 11: Don't care ※ Always read as 11.		TB1FF0 Invert Trigger 0: Disable trigger 1: Enable trigger				Control TB1FF0 00: Invert 01: Set 10: Clear 11: Don't care ※ Always read as 11.	
					Invert when the UC value is loaded to TB1CP1.	Invert when the UC value is loaded to TB1CP0.	Invert when the UC value is matches to TB1RG1.	Invert when the UC value is matches to TB1RG0.		
TB1RG0L	16-bit timer register 0L	198H (Prohibit RMW)	–							
			W							
			Undefined							
TB1RG0H	16-bit timer register 0H	199H (Prohibit RMW)	–							
			W							
			Undefined							
TB1RG1L	16-bit timer register 1L	19AH (Prohibit RMW)	–							
			W							
			Undefined							
TB1RG1H	16-bit timer register 1H	19BH (Prohibit RMW)	–							
			W							
			Undefined							
TB1CP0L	Capture register 0L	19CH	–							
			R							
			Undefined							
TB1CP0H	Capture register 0H	19DH	–							
			R							
			Undefined							
TB1CP1L	Capture register 1L	19EH	–							
			R							
			Undefined							
TB1CP1H	Capture register 1H	19FH	–							
			R							
			Undefined							

(9) UART/serial channel (1/2)

(9-1) UART/SIO channel0

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial channel 0 buffer	200H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC0CR	Serial channel 0 control	201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Over run	Parity	Framing	0: SCLK0 ↑ 1: SCLK0 ↓	1: Input SCLK0 pin
SC0MOD0	Serial channel 0 mode0	202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit 8	1: CTS enable	1: Receive enable	1: Wakeup enable	00: I/O interface 01: UART 7-bit 10: UART 8-bit 11: UART 9-bit		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK0	
BR0CR	Baud rate control	203H	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W	R/W						
			0	0	0	0	0	0	0	0
			Always write 0	1: (16 – K)/16 divided enabl	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set the frequency divisor N 0 to F			
BR0ADD	Serial channel 0 K setting register	204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Set the frequency divisor K (1 to F)			
SC0MOD1	Serial channel 0 mode 1	205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 1: Full duplex 0: Half duplex						

(9-2) IrDA

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIRCR	IrDA control register	207H	PLSEL	–	TXEN	RXEN	SIRWD3	SIRWD1	SIRWD1	SIRWD0
			R/W	R/W	R/W	R/W	R/W			
			0	0	0	0	0	0	0	0
			Transmit pulse width 0: 3/16 1: 1/16	Always write 0	Transmission 0: Disable 1: Enable	Receiving 0: Disable 1: Enable	Receiving pulse width Possible: 1 to 14 Not possible: 0, 15			

UART/serial channel (2)

(9-3) UART/SIO channel1

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial channel 1 buffer	208H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control	209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Over run	Parity	Framing	0: SCLK1 ↑ 1: SCLK1 ↓	1: Input SCLK1 pin
SC1MOD0	Serial channel 1 mode	20AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit 8	1: CTS enable	1: Receive enable	1: Wakeup enable	00: I/O interface 01: UART 7-bit 10: UART 8-bit 11: UART 9-bit		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK1	
BR1CR	Baud rate control	20BH	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write 0	1: (16 – K) /16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set the frequency divisor N 0 to F			
BR1ADD	Serial channel 1 K setting register	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set the frequency divisor K (1 to F)			
SC1MOD1	Serial channel 1 mode 1	20DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 1: Full duplex 0: Half duplex						

(10) I²C bus/serial interface

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	Serial bus interface control register 1	240H (I²C bus mode) (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/SWRMON
			W			R/W		W	W	R/W
			0	0	0	0		0	0	0/1
			Number of transfer bits 000: 8, 001: 1, 010: 2 011: 3, 100: 4, 101: 5 110: 6, 111: 7			Acknowledge mode 0: Disable 1: Enable		Setting for the divisor value n 000: 5, 001: 6, 010: 7 011: 8, 100: 9, 101: 10 110: 11, 111: Reserved		
		240H (SIO mode) (Prohibit RMW)	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			Transfer 0: Stop 1: Start	Transfer 0: Continue 1: Abort	Transfer mode 00: 8-bit transmit mode 10: 8-bit transmit/ receive mode 11: 8-bit received mode			Setting for the divisor value n 000: 4, 001: 5, 010: 6 011: 7, 100: 8, 101: 9 110: 10, 111: SCK pin		
SBI0DBR	SBI data buffer register	241H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (receiving) /W (transmission)							
			Undefined							
I2C0AR	I²C bus address register	242H (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Setting slave address							Address recognition 0: Enable 1: Disable
When read SBI0SR	Serial bus interface status register	243H (I²C bus mode) (Prohibit RMW)	MST	TRX	BB	PIN	AL/SBIM1	AAS/SBIM0	AD0/SWRST1	LRB/SWRST0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: Receive 1: Transmit	Bus status monitor 0: Free 1: Busy	INTSBI request monitor 0: Request 1: Cancel	Arbitration lost detection monitor 1: Detect	Slave address match detection monitor 1: Detect	GENERAL CALL detection monitor 1: Detect	Lost receive bit monitor 0: 0 1: 1
When write SBI0CR2	Serial bus interface control register 2				Start/stop condition generation 0: Start condition 1: Stop condition		Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset generate write "10" and "01", then an internal reset signal is generated.	
When read SBI0SR	Serial bus interface status register	243H (SIO mode) (Prohibit RMW)					SIOF/SBIM1	SEF/SBIM2	–	–
							R/W	R/W	R/W	R/W
							0	0	0	0
							Transfer status monitor 0: Stopped 1: Terminated in process	Shift operation status monitor 0: Stopped 1: Terminated in process		
When write SBI0CR2	Serial bus interface control register 2						Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Always write 0.	Always write 0.
SBI0BR0	Serial bus interface baud rate register 0	244H (Prohibit RMW)	–	I2SBI0						
			W	R/W						
			0	0						
			Always write 0.	IDLE2 0: Abort 1: Operate						
SBI0BR1	Serial bus interface baud rate register 1	245H (Prohibit RMW)	P4EN	–						
			W	W						
			0	0						
			Clock control 0: Abort 1: Operate	Always write 0						

(11) AD convertor

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD mode register 0	2B0H	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
			R		R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			1: End	1: Busy	Always write "0"	Always write "0"	Interrupt on repeat mode	1: Repeat	1: Scan	1: Start
ADMOD1	AD mode register 1	2B1H	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
			R/W				R/W			
			0	0			0	0	0	0
			1: VREF on	IDLE2 0: Abort 1: Operate			1: Enable for external start	Input channel 000: AN0 AN0 001: AN1 AN0→AN1 010: AN2 AN0→AN1→AN2 011: AN3 AN0→AN1→AN2→AN3 100: AN4 AN4 101: AN5 AN4→AN5 110: AN6 AN4→AN5→AN6 111: AN7 AN4→AN5→AN6→AN7		
ADREG04L	AD result register 0/4 low	2A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
ADREG04H	AD result register 0/4 high	2A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG15L	AD result register 1/5 low	2A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
ADREG15H	AD result register 1/5 high	2A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG26L	AD result register 2/6 low	2A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
ADREG26H	AD result Register 2/6 high	2A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG37L	AD result register 3/7 low	2A6H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
ADREG37H	AD result register 3/7 high	2A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

(12) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT mode register	300H	WDTE	WDTP1	WDTP0			I2WDT	RESCR	–
			R/W	R/W	R/W			R/W	R/W	R/W
			1	0	0			0	0	0
			1: WDT enable	00: 2^{15}/fsys 01: 2^{17}/fsys 10: 2^{19}/fsys 11: 2^{21}/fsys				IDLE2 0: Abort 1: Operate	1: Reset connect internally WDT out to reset pin	Always write 0
WDCR	WD control	301H (Prohibit RMW)	–							
			W							
			–							
			B1H: WDT disable 4EH: WDT clear							

(13) RTC (Real Time Clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
RTCCR	RTC control register	310H	–					RTCSEL1	RTCSEL0	RTCRUN
			R/W					R/W		R/W
			0					0	0	0
			Always write 0					00: 2^{14}/fs 01: 2^{13}/fs 10: 2^{12}/fs 11: 2^{11}/fs		0: Stop and clear 1: Run

6. Port Section Equivalent Circuit Diagram

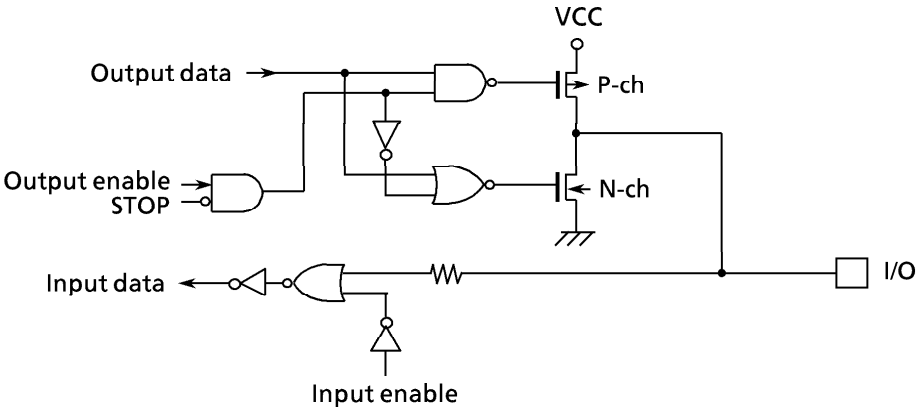
- Reading the circuit diagram

  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.
  The dedicated signal is described below.

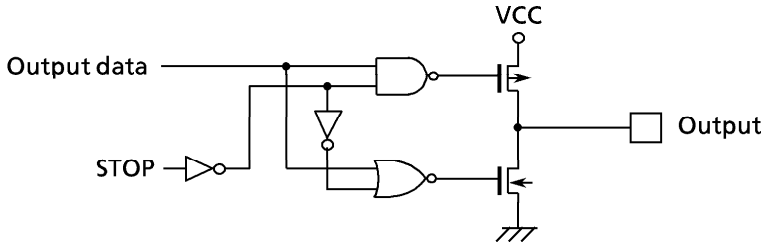
  STOP: This signal becomes active 1 when the HALT mode setting register is set to the STOP mode (SYSCR2<HALTM1:0>=01) tand the CPU executes the HALT instruction. When the drive enable bit SYSCR2<DRVE> is set to 1, however, STOP remains at 0.

- The input protection resistans ranges from several tens of ohms to several hundreds of ohms.

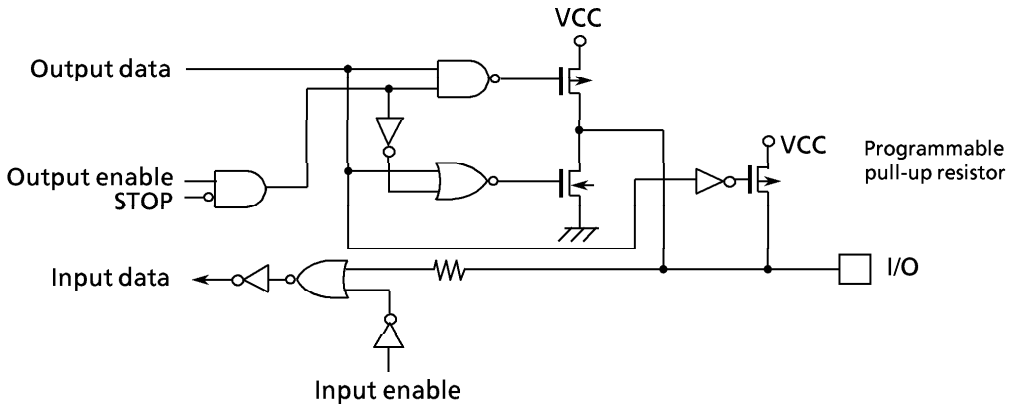
■ P0 (AD0 to AD7), P1 (AD8 to AD15, A8 to A15), P2 (A16 to A23, A0 to A7), P60, P64 to P66, P70 to P75, P80 to P87, P91, P92, P94, P95, PA0 to A7



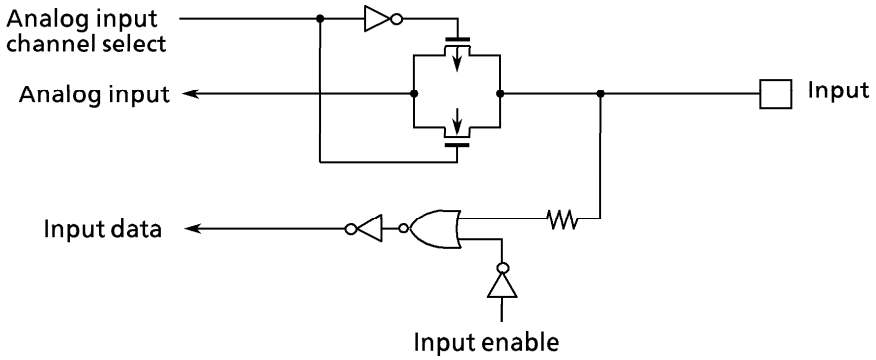
■ P30 ($\overline{RD}$), P31 ($\overline{WR}$)



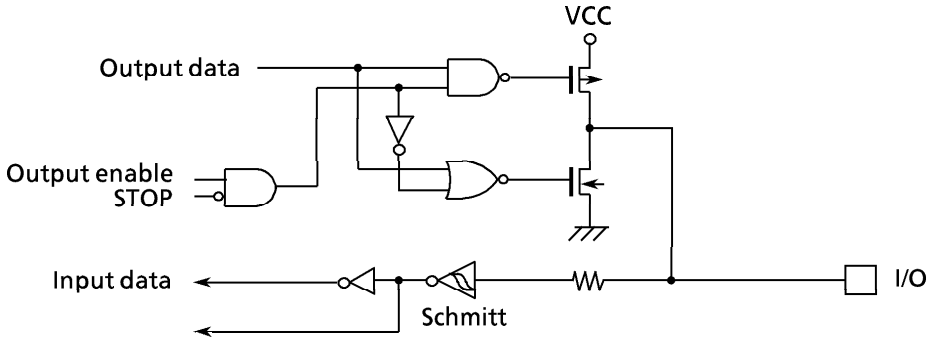
■ P32 to P37, P40 to P43



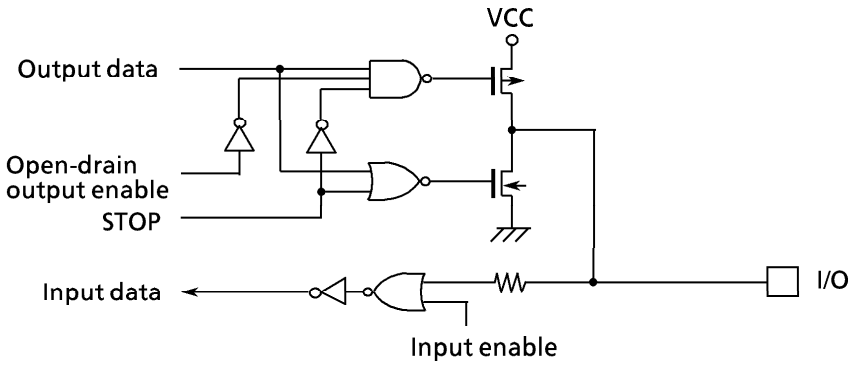
■ P5 (AN0 to AN7)



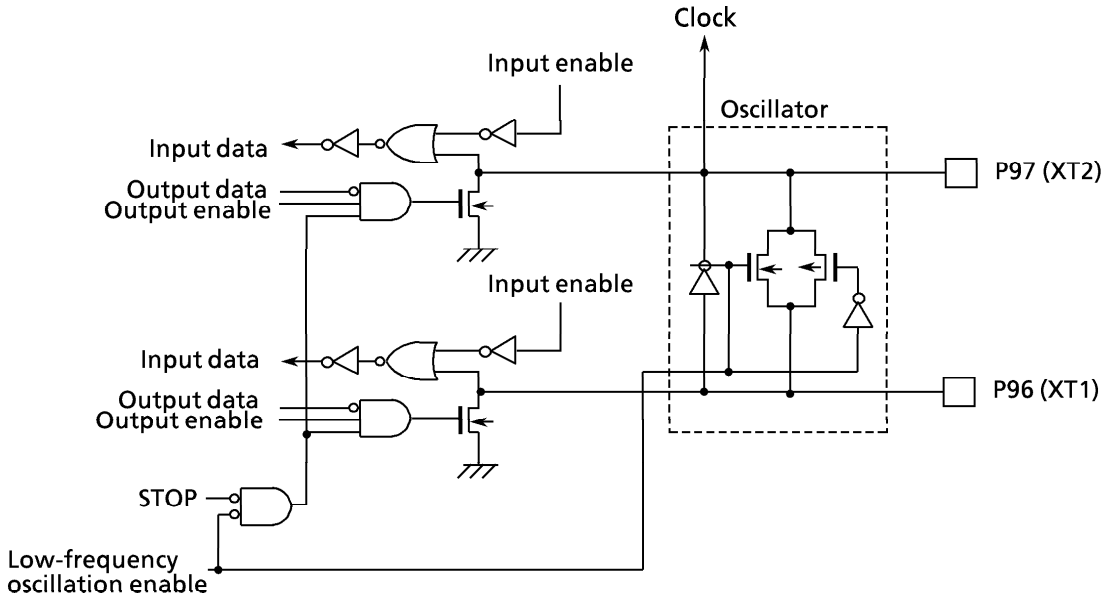
■ P63 (INT0)



■ P61 (SO/SDA), P62 (SI/SCL), P90 (TXD0), P93 (TXD1)

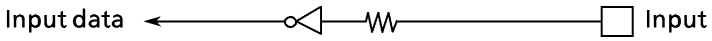


■ P96 (XT1), P97 (XT2)

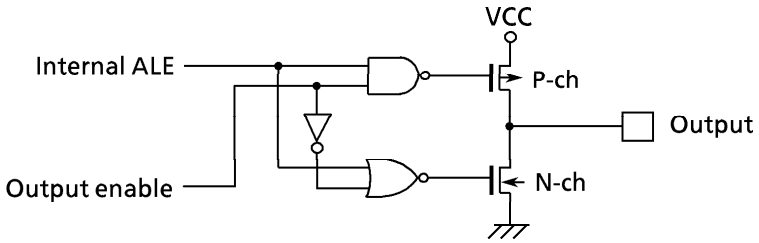


■ $\overline{\text{NMI}}$

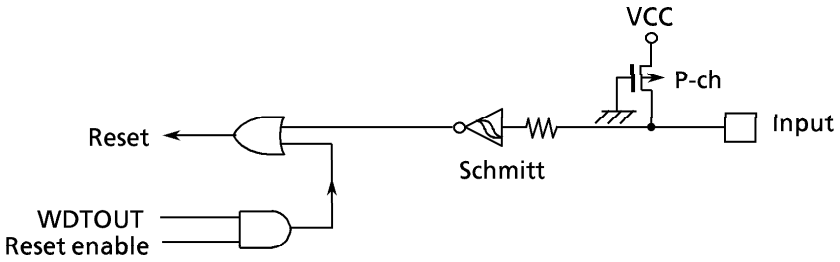
■ AM0 to AM1



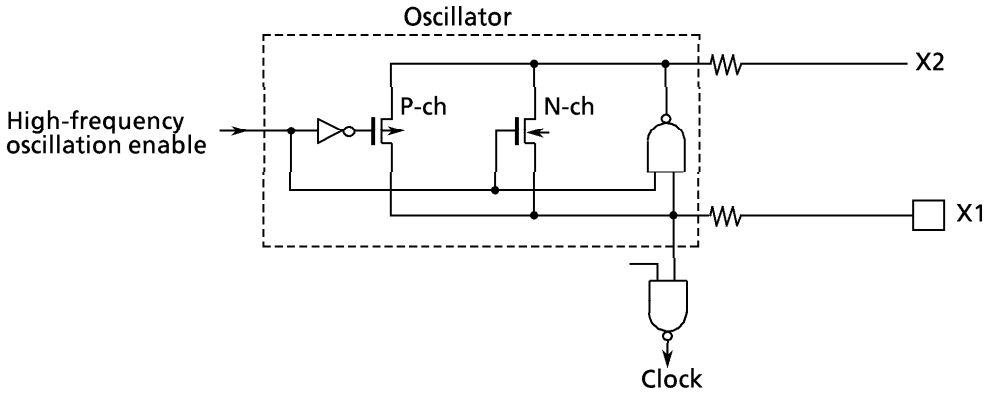
■ ALE



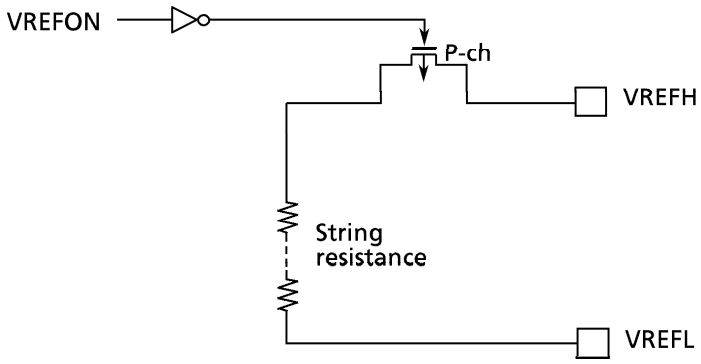
■ $\overline{\text{RESET}}$



■ X1, X2



■ VREFH, VREFL



7. Points to Note and Restrictions

(1) Notation

① How a built-in I/O register is denoted: Register symbol <Bit symbol>
e.g.) TA01RUN<TA0RUN> ··· Bit TA0RUN of register TA01RUN

② Read-modify-write instruction

An instruction in which the CPU reads data from memory and writes the data to the same memory location in one instruction.
Example 1: SET 3, (TA01RUN) ··· Set bit3 of TA01RUN
Example 2: INC 1, (100H) ··· Increment the data at 100H

- A sample read-modify-write instructions using the TLCS-900

Exchange instruction
EX (mem), R

Arithmetic operation

ADD (mem), R/#	ADC (mem), R/#
SUB (mem), R/#	SBC (mem), R/#
INC #3, (mem)	DEC #3, (mem)

Logic operation

AND (mem), R/#	OR (mem), R/#
XOR (mem), R/#	

Bit manipulation

STCF #3/A, (mem)	RES #3, (mem)
SET #3, (mem)	CHG #3, (mem)
TSET #3, (mem)	

Rotate, shift

RLC (mem)	RRC (mem)
RL (mem)	RR (mem)
SLA (mem)	SRA (mem)
SLL (mem)	SRL (mem)
RLD (mem)	RRD (mem)

③ f_{OSCH}, fc, fs, f_{FPH}, f_{SYS}, one state

The clock frequency input from pins X1 and X2 is called f_{OSCH} the clock selected by DFMCR0<ACT1:0> is called fc, the clock frequency input from pins XT1 and XT2 is called fs, the clock selected by SYSCR1<SYSCK> is called f_{FPH}, and the clock frequency given by f_{FPH} divided by 2 is called f_{SYS}. One cycle of f_{SYS} is called one state.

(2) Points to note

① AM0, AM1 pins

Fix these pins to V_{CC} unless changing voltage.

② EMU0, EMU1 pins

Open pins.

③ Reserved area for address

TMP91CW12 do not have the reserved area.

④ HALT mode (IDLE1)

When IDLE1 mode (Oscillator operation only) is used, set RTCCR<RTCRUN> to 0 to stop the timer for real time clock before the HALT instruction is executed.

⑤ Warm-up counter

The warm-up counter operates when STOP mode is released even if the system is using an external oscillator. As a result, a time equivalent warm-up time elapses from input of the release request to output of the system clock.

⑥ Programmable pull-up resistance

The programmable pull-up resistor can be turned ON/OFF by the program when the ports are used as input ports. When the ports are used as outputs, they cannot be turned ON/OFF by the program.
The data registers (e.g., P3) are used for the pull-up/down resistors ON/OFF. Consequently read-modify-write instructions are prohibited.

⑦ Bus releasing function

Refer to the "Note about the Bus Release" in 3.5 "Functions of Ports" because the pin state, when the bus is released, is written.

⑧ Watchdog timer

The watchdog timer starts operation immediately after the reset is released. When the watch dog timer will not be used, disable it. When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate. So, the watch dog timer continues to run. Therefore, be careful about the bus releasing time and set the detection timer of watch dog timer.

⑨ AD converter

The string resistor between the VREFH and VREFL pins can be cut by a program to reduce power consumption. When STOP mode is used, disable the resistor using the program before the HALT instruction is executed.

⑩ CPU (Micro DMA)

Only the "LDC cr, r" and "LDC r, cr" instructions can be used to access the control registers in the CPU (like the transfer source address register (DMASn)).

⑪ Undefined SFR

The undefined bit of SFR are read as undefined value.

⑫ POP SR instruction

Please execute POP SR instruction during DI condition.

⑬ Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{NMI}$, INT0 to INT4, INTRTC), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

8. Package Dimensions

P-LQFP100-1414-0.50C

Unit: mm