Low Voltage/Low Power

CMOS 16-bit Microcontrollers

TMP93CM40F/TMP93CM41F

1. Outline and Device Characteristics

TMP93CM40/M41 are high-speed advanced 16-bit microcontrollers developed for controlling medium to large-scale equipment. The TMP93CM41 does not have a ROM, the TMP96CM40 has a built-in ROM. Otherwise, the devices function in the same way.

TMP93CM40F/TMP93CM41F are housed in 100-pin mini flat package. Device characteristics are as follows:

(1) Original 16-bit CPU (900L CPU)
- TLCS-90 instruction mnemonic upward compatible.
- 16M-byte linear address space
- General-purpose registers and register bank system
- 16-bit multiplication/division and bit transfer/arithmetic instructions
- High-speed micro DMA
  - 4 channels (1.6 μs/2 bytes at 20MHz)

(2) Minimum instruction execution time
- 200ns at 20MHz

(3) Internal RAM: 2K byte
Internal ROM:

TMP93CM40	32K-byte ROM
TMP93CM41	None

(4) External memory expansion
- Can be expanded up to 16M bytes (for both programs and data).
- AM8/$\overline{16}$ pin (select the external data bus width).
- Can mix 8- and 16-bit external data buses.
  ···Dynamic data bus sizing

(5) 8-bit timer: 2 channels
(6) 8-bit PWM timer: 2 channels
(7) 16-bit timer: 2 channels
(8) Pattern generator: 4 bits, 2 channels
(9) Serial interface: 2 channels
(10) 10-bit A/D converter: 4 channels
(11) Watchdog timer
(12) Chip select/wait controller: 3 blocks
(13) Interrupt functions
- 2 CPU interrupts··· ···SWI instruction, and Illegal instruction
- 14 internal interrupts
- 6 external interrupts

7-level priority can be set.

(14) I/O ports:
79 pins for TMP93CM40 and 61 pins for TMP93CM41
(15) Standby function : 4 halt modes (RUN, IDLE2, IDLE1, STOP)
(16) Clock Gear Function
- High-frequency clock can be changed fc to fc/16
- Dual clock operation

(17) Wide Operating Voltage
- V_{CC} = 2.7 to 5.5V

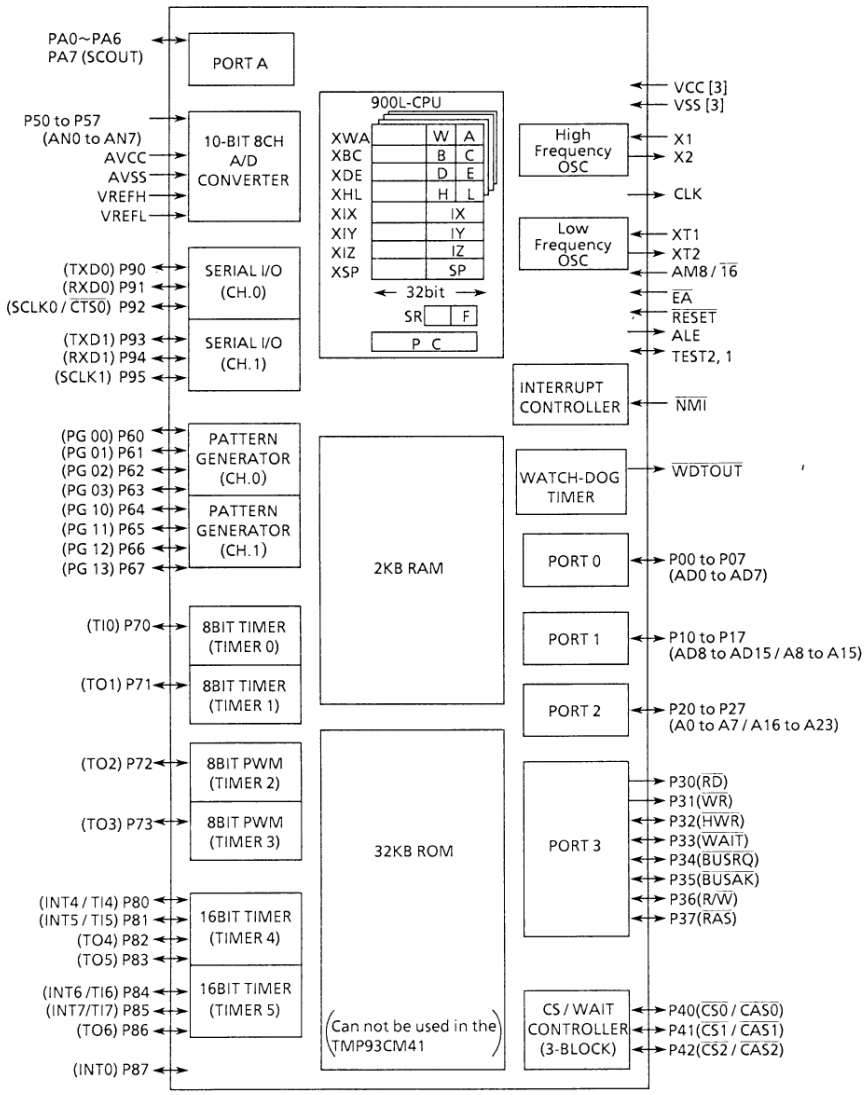


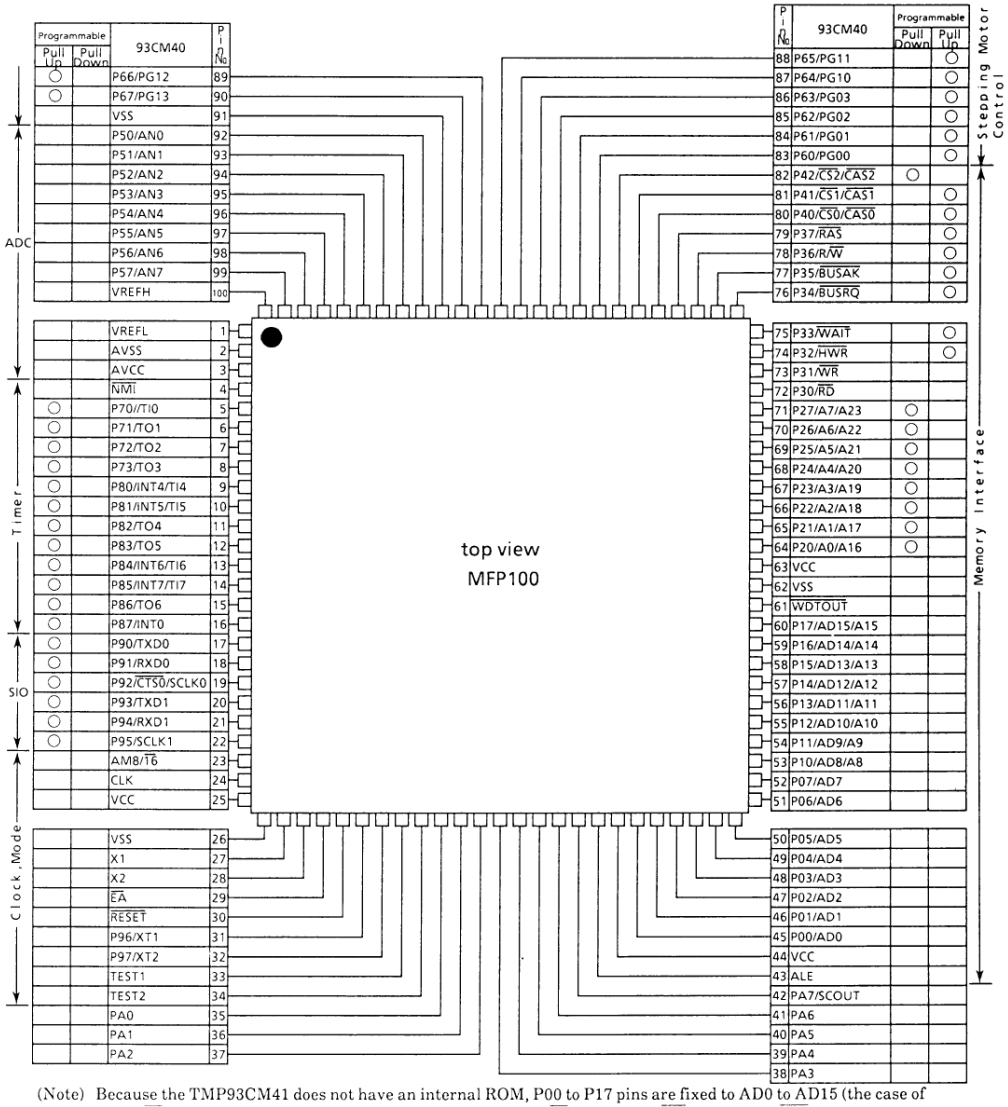
Figure 1. TMP93CM40/TMP93CM41 Block Diagram

2. Pin Assignment and Functions

The assignment of input/output pins for TMP93CM40/ TMP93CM41, their name and outline functions are described below.

2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP93CM40F/ TMP93CM41F.



(Note) Because the TMP93CM41 does not have an internal ROM, P00 to P17 pins are fixed to AD0 to AD15 (the case of AM8 / $\overline{16}$ = 0), or to AD0 to AD7, A8 to A15 (the case of AM8 / $\overline{16}$ = 1) ; P30 to $\overline{RD}$; and P31 to $\overline{WR}$.

Figure 2.1. Pin Assignment (100-pin MFP)

2.2 Pin Names and Functions

The names of input/output pins and their functions are described below.

Table 2.2. Pin Names and Functions

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 on a bit basis Address / data (lower): 0 to 7 for address / 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 on a bit basis Address data (upper): 8 to 15 for address / data bus Address: 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 selection of I/O on a bit basis (with pull-down resistor) Address: 0 to 7 for address bus Address: 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 to7
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
P34 $\overline{BUSRQ}$	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 to 15, A0 to 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins. (For external DMAC)
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 to 15, A0 to 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins are at high impedance after receiving BUSRQ. (For external DMAC)
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, write cycle.
P37 $\overline{RAS}$	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs RAS strobe for DRAM.
P40 $\overline{CS0}$ $\overline{CAS0}$	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs CAS strobe for DRAM when address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P41 $\overline{CS1}$ $\overline{CAS1}$	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P42 $\overline{CS2}$ $\overline{CAS2}$	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P50 to P53 AN0 to AN3	4	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1	Input	Pin for reference voltage input to A/D converter
AGND	1	Input	Ground pin for A/D converter
P60 to P63 PG00 to PG03	4	I/O Output	Ports 60 to 63: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 00 to 03
P64 to P67 PG10 to PG13	4	I/O Output	Ports 64 to 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 10 to 13
P70 TI0	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 T01	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 T02	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 T03	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 TI4 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
P81 TI5 INT5	1	I/O Input Input	Port 81: I/O port (with pull-up resistor) Timer input 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with rising edge
P82 TO4	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 TO5	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port (with pull-up resistor) Serial receive data 0
P92 $\overline{\text{CTS0}}$ SCKL0	1	I/O Input I/O	Port 92: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send) Serial Clock I/O
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) 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	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O 1
PA7 SCOUT	1	I/O Output	Port A7: I/O port System clock output: Outputs system clock or 1/2 oscillation clock for synchronizing to external circuit.
$\overline{\text{WDTOUT}}$	1	Output	Watchdog timer output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs ⌈System Clock ÷ 2 ⌋Clock. Pulled-up during reset (can be reset to Output Disable for reducing noise).
$\overline{\text{EA}}$	1	Input	External access: “0” should be input with TMP93CM41 “1” should be input with TMP96CM40.
AM8/$\overline{16}$	1	Input	Address mode: Selects external data bus width For TMP93CM40: “1” should be input. The data bus width for external access is set by Chip Select/WAIT Control register, Port 1 Control register. For TMP93CM41: “0” should be input with fixed 16bit bus width or 16bit bus interlarded with 8bit bus. “1” should be input with fixed 8bit bus width.
ALE	1	Output	Address latch enable. Can be set Output disable for reducing noise.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
XT1 P96	1	Input I/O	Low Frequency Oscillator connecting pin Port 96: I/O port (Open Drain Output)
XT2 P97	1	Output I/O	Low Frequency Oscillator connecting pin Port 97: I/O port (Open Drain Output)
TEST1/TEST2	2	Output Input	TEST1 Should be connected with TEST2 pin
VCC	3		Power supply pin
VSS	3		GND pin (0V)
AVCC	1		Power supply pin for A/D converter
AVSS	1		GND pin for A/D converter (0V)

Note: Pull-up/pull-down resistor can be released from the pin by software.

3. Operation

This section describes in blocks the functions and basic operations of TMP93CM40A/M41A devices.

Check the [7. Care Points and Restriction] because the Care Points, etc., are described.

3.1 CPU

TMP93CM40A/M41A devices have a built-in high-performance 16-bit CPU (900L CPU). (For CPU operation, see TLCS-900 CPU in the previous section).

This section describes CPU functions unique to TMP 93CM40/M41 that are not described in the previous section.

3.1.1 Reset

To reset the TMP93CM40, the $\overline{\text{RESET}}$ input must be kept at 0 for at least 160 system clocks (160 states: 16μs at 20MHz) within an operating voltage range and with a stable oscillation.

When reset is accepted, the CPU sets as follows:

- Program counter (PC) to 8000H.
  PC (7 : 0) → stored data to 8000H
  PC (15 : 8) → stored data to 8001H
  PC (23 : 16) → stored data to 8002H

Note: Reset Vector address is different with each product. Set PC (23 : 16) to "00H" and locate Reset Vector within 64K-byte area for TMP93CM40/M41.

- Stack pointer (XSP) for system mode to 100H.
- IFF2 to 0 bits of status register to 111. (Sets mask register to interrupt level 7.)
- MAX bit of status register to 0. (Sets to minimum mode.)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

When reset is released, instruction execution starts from PC (reset vector). CPU internal registers other than the above are not changed.

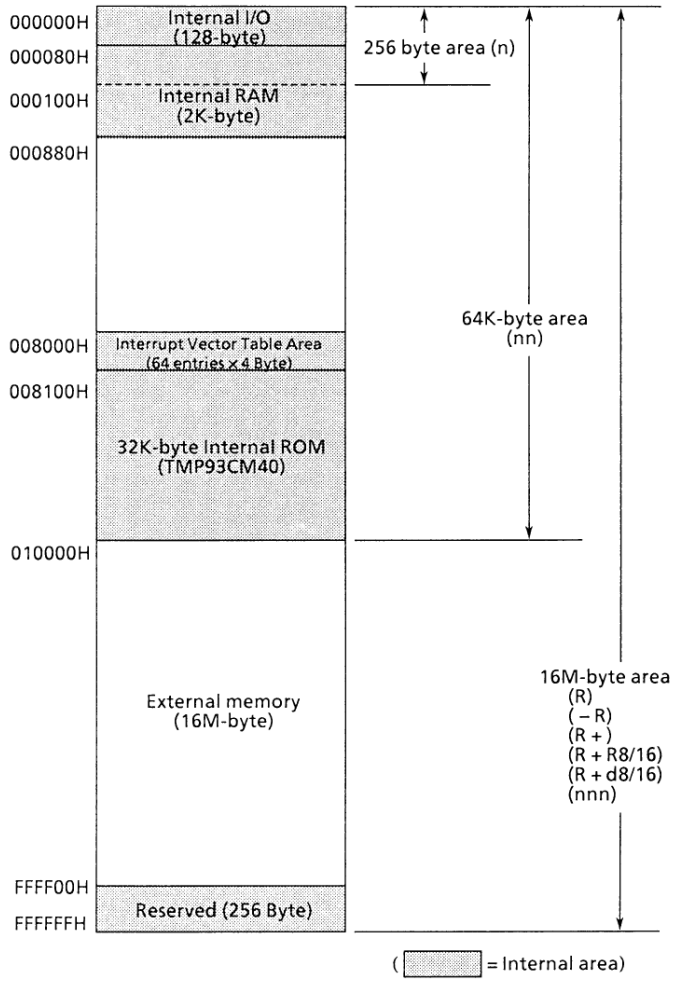
When reset is accepted, processing for built-in I/Os, ports, and other pins is as follows:

- Initializes built-in I/O registers as per specifications.
- Sets port pins (including pins also used as built-in I/Os) to general-purpose input/output port mode.
- Sets the $\overline{\text{WDTOUT}}$ pin to 0. (Watchdog timer is set to enable after reset.)
- Pulls up the CLK pin to 1.
- Sets the ALE pin to 0 (TMP93CM41), to High Impedance (Hz) (TMP93CM40).

Note: By resetting, register in the CPU except program counter (PC), status register (SR) and stack pointer (XSP) and the data in internal RAM are not changed.

3.2 Memory Map

Figure 3.2 is a memory map of the TMP93CM40/M41.



Note : Resetting sets the stack pointer (XSP) to 100H.
The 256 Byte Area from FFFF00H to FFFFFFH can not be used.

Figure 3.2. Memory Map

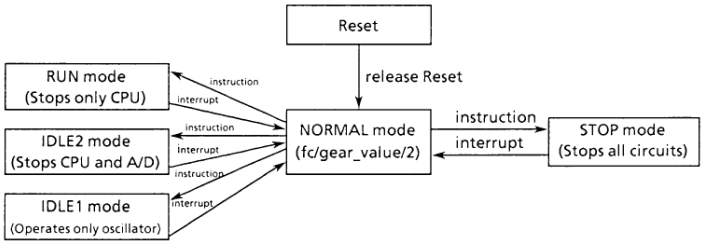
3.3 Dual Clock Standby Function

Dual Clock, Standby Control Circuits consist of (1) System Clock Controller, (2) Prescaler Clock Controller, and (3) Standby Controller.

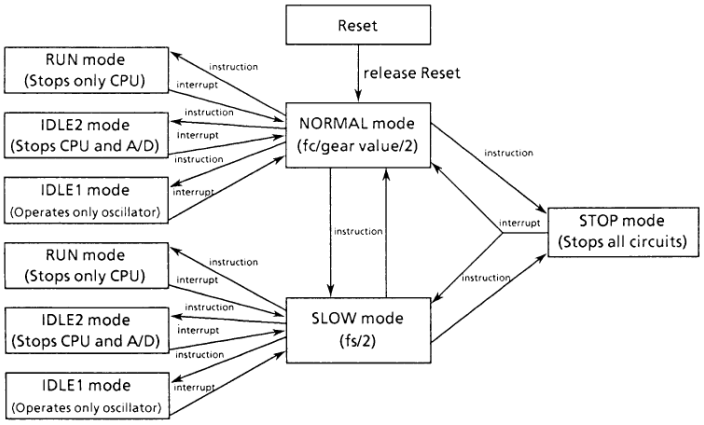
The Oscillator operation mode is classified to (a) Single Clock mode (only X1, X2 pin), and (b) Dual Clock mode (X1, X2, XT1, XT2 pin).

Figure 3.3.1 shows a transition figure. Figure 3.3.2 shows the block diagram.

Figure 3.3.3 shows I/O registers.



(a) Signal Clock mode transition figure



(b) Dual Clock mode transition figure

Figure 3.3.1 (1/2). Transition Figure

The Clock Frequency input from X1, X2 pin is called fc, and the Clock Frequency input from XT1, XT2 pin is called fs. The clock frequency selected by SYSCR1 <SYSCK> is called system clock f_{FPH}. The divided clock of f_{FPH} is called system clock f_{SYS}, and the 1 cycle of f_{SYS} is called 1 state.

Operating Mode		Oscillator		CPU	internal I/O	System clock f_{SYS}
		High Frequency (fc)	Low Frequency (fs)			
Single Clock	RESET	oscillation	stop	reset	reset	$fc/32$
	NORMAL			operate	operate	programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$)
	RUN			stop		
	IDLE2				stop only A/D	
	IDLE1				stop	
	STOP	stop				—
Dual Clock	RESET	oscillation	stop	reset	reset	$fc/32$
	NORMAL		programmable	operate	operate	programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$)
	SLOW	programmable	oscillation			$fs/2$
	RUN	Oscillator using as system clock : oscillation Other oscillator : programmable		stop		programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$, $fs/2$)
	IDLE2				stop only A/D	
	IDLE1				stop	
	STOP	stop				—

Figure 3.3.1 (2/2). Internal Operation and System Clock

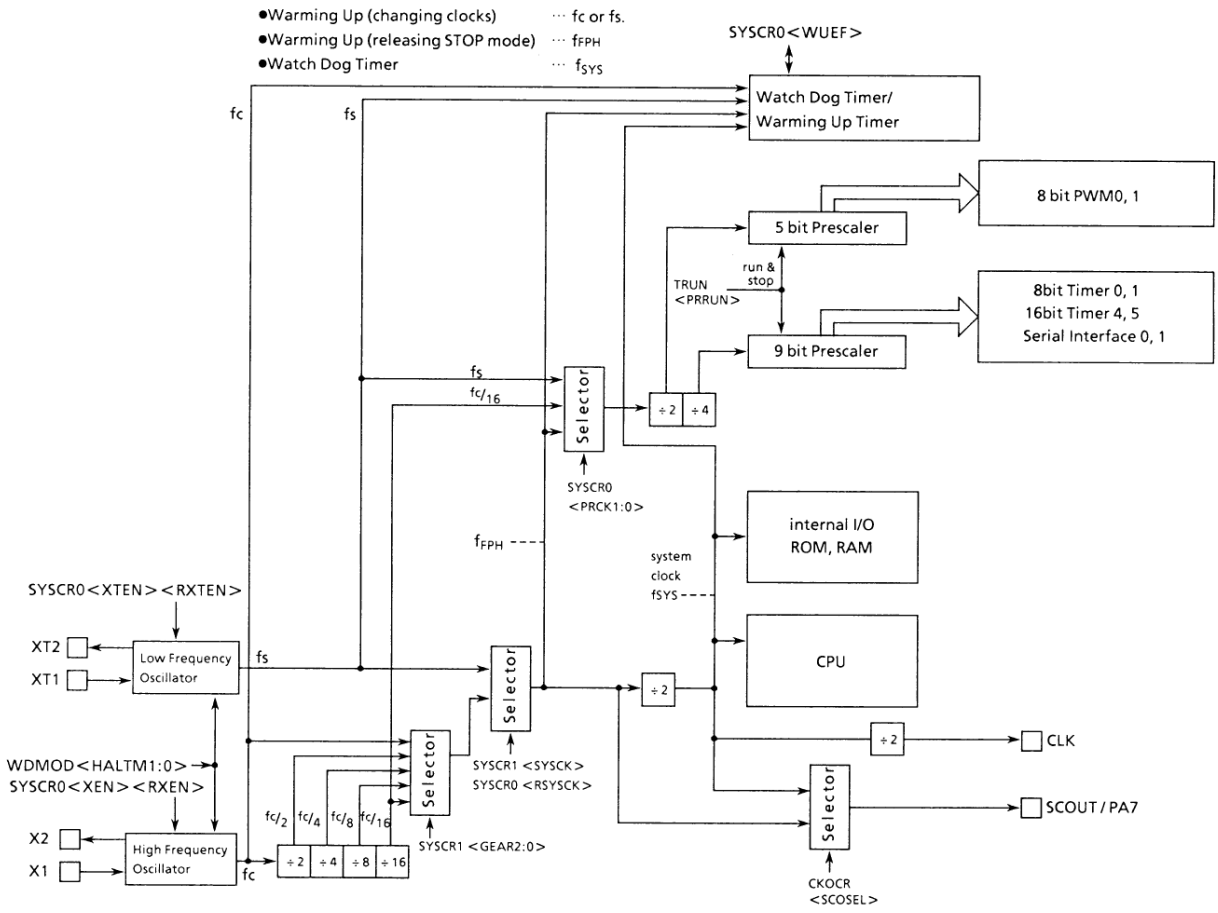


Figure 3.3.2. Block Diagram of Dual Clock, Standby Circuits

SYSCR0 (006EH)

	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	Warming Up Timer 0 write : don't care 1 write : start timer 0 read : end warming up 1 read : not end warming up	select prescaler clock 00 : f_{FPH} 01 : fs 10 : fc/16 11 : (reserved)	

SYSCR1 (006FH)

	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)		

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM0	HALTM0	RESCR	DRVE
Read / Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 1 : enable	WDT Detection Time 00 : $2^{15}/f_{SYS}$ 01 : $2^{17}/f_{SYS}$ 10 : $2^{19}/f_{SYS}$ 11 : $2^{21}/f_{SYS}$		Warming Up Timer 0 : 2^{14}/ inputted frequency 1 : 2^{16}/ inputted frequency	Standby mode 00 : RUN mode 01 : STOP mode 10 : IDLE1 mode 11 : IDLE2 mode		1 : Connects WDT output to $\overline{RESET}$ pin internally.	1 : Drives pin even in STOP mode

Figure 3.3.3. I/O Register About Dual Clock, Standby

(1) 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 <GEAR 2 : 0> changes high frequency clock gear either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16), these functions can reduce the power consumption.

The system clock (f_{SYS}) is set to fc/32 (fc/16 x 1/2) because of <XEN> = "1", <XEN> = "0", <SYSCK> = "0", <GEAR 2 : 0> = "100" by resetting.

For example, f_{SYS} is set to 0.5MHz by resetting 16MHz oscillator is connected to X1, X2 pins.

The high frequency (fc) and low frequency (fs) clocks can be easily obtained by connecting a resonator to the X1/X2, XT1/XT2 pins, respectively. Clock input from an external oscillator is also possible.

The XT1, XT2 pins have also Port 96, 97 function. Therefore, single clock mode, the XT1, XT2 pins can be used as I/O port pins.

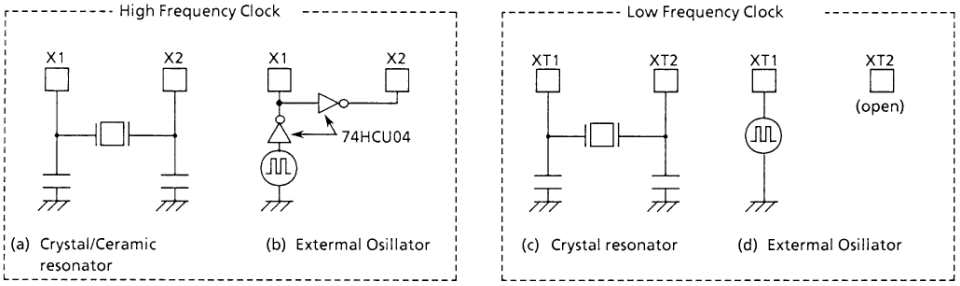


Figure 3.3.4. Examples of Resonator Connection

* Accurate Adjustment of the Oscillation Frequency

The CLK pin outputs 1/2 clock frequency (f_{SYS}/2) to monitor the oscillation clock.

With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

* Clock modes and Warming-up time

When the resonator is connected to X1, X2, or XT1, XT2 pin, the warming-up timer is used to change the operation frequency after getting stabilized oscillation.

The warming up time can be selected by WDMOD <WARM>.

This starting and ending of warming up timer are performed like the following example 1, 2 by program.

Note 1: The warming up timer is also used as a watchdog timer. So, when it is used as a warming up timer, the watchdog timer must be disabled.

Note 2: When using the oscillator (not resonator) with stabilized oscillation, a warming up timer is not needed.

Note 3: The warming up timer is operated by an oscillation clock. Therefore, warming up time has an error.

Table 3.3.1 Warming Up Time

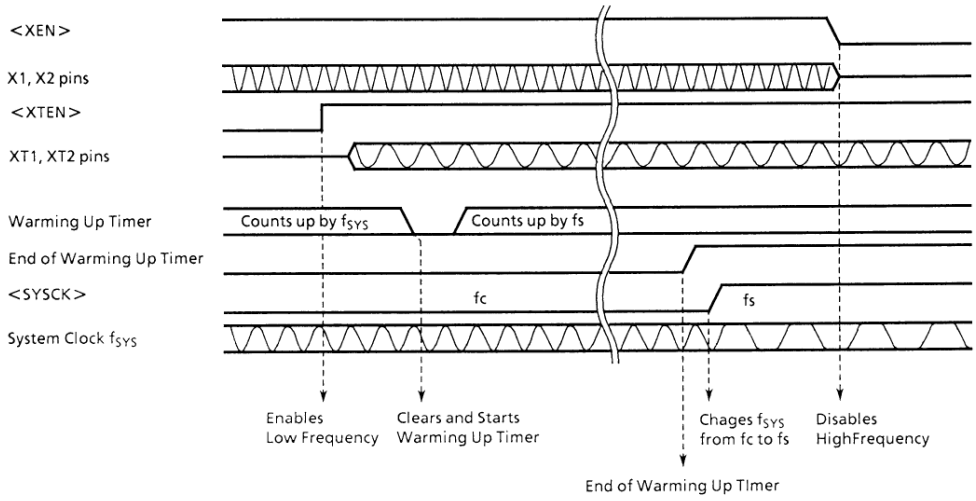
Warming Up Time WDMOD<WARM>	Change to NORMAL	Change toSLOW
0 (2^{14}/frequency)	1.024 (ms)	500 (ms)
1 (2^{16}/frequency)	4.096 (ms)	2000 (ms)

at fc = 16 MHz, fs = 32.768 kHz

Setting Example 1

The case of changing from High Frequency (fc) to Low Frequency (fs).

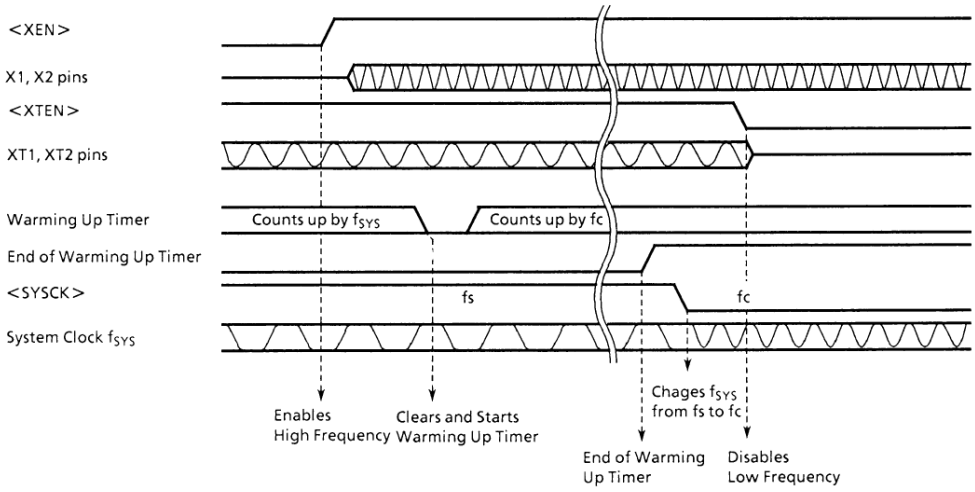
```
SYSCR0    EQU   006EH
SYSCR1    EQU   006FH
WDCR      EQU   005DH
WDMOD     EQU   005CH
          LD    (WDCR), B1H      ;
          RES   7, (WDMOD)       ;  } Disables Watchdog Timer.
          SET   4, (WDMOD)       ; Sets Warming Up Time to 2^16/fs.
          SET   6, (SYSCR0)      ; Enables Low Frequency Oscillation
          SET   2, (SYSCR0)      ; Clears and starts Warming Up Timer.
WUP :     BIT   2, (SYSCR0)      ;
          JR    NZ, WUP          ;  } Detects End of Warming Up Timer.
          SET   3, (SYSCR1)      ; Changes fSYS from fc to fs.
          RES   7, (SYSCR0)      ; Disables High Frequency Oscillation.
          SET   7, (WDMOD)       ; Enables Watchdog Timer.
```



Setting Example 2

The case of changing from Low Frequency (fs) to High Frequency (fc).

```
SYSCR0    EQU   006EH
SYSCR1    EQU   006FH
WDCR      EQU   005DH
WDMOD     EQU   005CH
          LD    (WDCR), B1H      ;  } Disables Watchdog Timer.
          RES   7, (WDMOD)       ;  }
          RES   4, (WDMOD)       ; Sets Warming Up Time to 2^14/fc.
          SET   7, (SYSCR0)      ; Enables High Frequency (fc).
          SET   2, (SYSCR0)      ; Clears and Starts Warming Up Timer.
WUP :     BIT   2, (SYSCR0)      ;  } Detects End of Warming Up Timer.
          JR    NZ, WUP          ;  }
          RES   3, (SYSCR1)      ; Changes fSYS from fs to fc.
          RES   6, (SYSCR0)      ; Disables Low Frequency Oscillation.
          SET   7, (WDMOD)       ; Enable Watchdog timer
```



Setting Example 3

The case of changing gear value of high frequency

```
SYSCR1    EQU   006FH

          LD    (SYSCR1), XXXX0000B    ; Changes fSYS to fc/2
          LD    (SYSCR1), XXXX0100B    ; Changes fSYS to fc/32
```

X : don't care

(2) Prescaler Clock Controller

The 9 bit prescaler provides a clock to 8bit Timer 0, 1, 16bit Timer 4, 5, and Serial Interface 0, 1, and the 5 bit prescaler provides a clock to 8 bit PWM Timer 0,1.

The clock input to the 5 bit prescaler is a clock divided by 2 which is selected either f_{FPH}, fc/16, or fs by SYSCR0 <PRCH1 : 0> register.

The clock input to the 9 bit prescaler is a clock divided by 4 which is selected either f_{FPH}, fc/16, or fs by SYSCR0 <PRCH1 : 0> register.

<PRCK1 : 0> register is initialized to "00" resetting.

When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop 9, 5 bit prescaler before "HALT" instruction is executed.

(3) Standby Controller

When the "HALT" instruction is executed at NORMAL or SLOW mode, the operating mode changes RUN, IDLE2, IDLE2, or STOP mode depending on the contents of the HALT mode setting register WDMOD <HALTM 1 : 0>.

① RUN: Only the CPU halts; power consumption remains unchanged.

② IDLE2: The built-in oscillator and the specified I/O operates.
The power consumption is reduced to 1/3 than that during NORMAL operation.

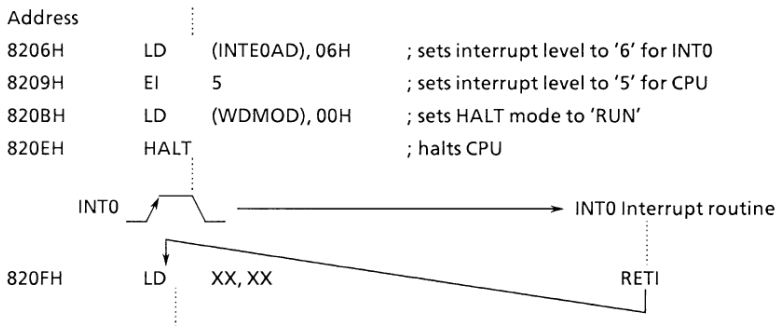
③ IDLE1: Only the built-in oscillator operates, while all other built-in circuits stop. The power consumption is reduced to 1/10 or less than that during NORMAL operation.

④ STOP: All internal circuits including the built-in oscillator stop. This greatly reduces power consumption.

These HALT states can be released by resetting or requesting an interrupt. The methods for releasing the HALT status are shown in Table 3.3 (2).

Either a non-maskable or maskable interrupt with EI (enable interrupt) condition is acknowledged and interrupt processing is processed. A maskable interrupt with DI (disable interrupt) condition is also acknowledged and CPU starts executing an instruction that follows the HALT instruction, but the interrupt request flag is held at "1".

(Example releasing "RUN" mode)
INT0 interrupt releases HALT state when the RUN mode is on.



When the halt state is released by a reset, that status in effect before entering the halt status (including built-in RAM) is held.

① RUN mode

Figure 3.3.5 shows the timing for releasing the HALT state by interrupts in the RUN/IDLE2 mode.

In the RUN mode, the system clock in the MCU continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction. Until the HALT state is released, the CPU repeats dummy cycles. In the HALT state, an interrupt request is sampled with the rising edge of the “CLK” signal.

The external interrupts (INT4, 5, 6, 7) releases only RUN and IDLE2 mode.

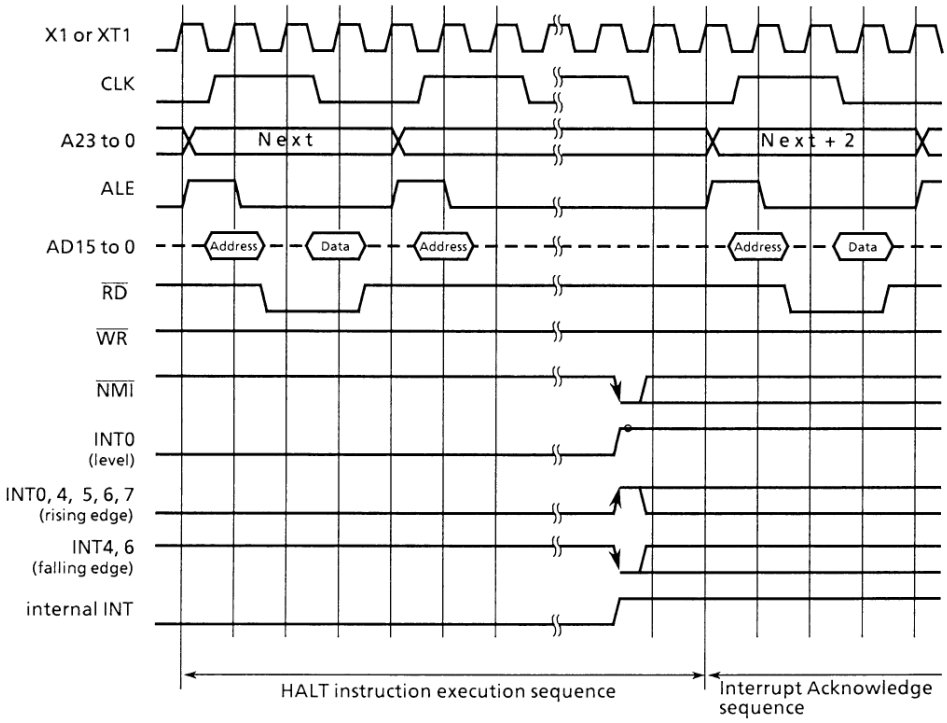


Figure 3.3.5. Timing Chart for Releasing the HALT State by Interrupt in RUN/IDLE2 Modes

② IDLE2 mode

Figure 3.3.5 shows the timing for releasing the HALT state by interrupts in the RUN/IDLE2 mode.

In the IDLE2 mode, the HALT state is released by an interrupt with the same timing as in the RUN mode, except the internal operation of the MCU. In the RUN mode, only the CPU stops executing the current instruction, and the system clock is supplied to all internal devices. In the IDLE2 mode, however, the system clock is supplied to only specific internal I/O devices.

③ IDLE1 mode

Figure 3.3.6 illustrates the timing for releasing the HALT state by interrupts in the IDLE1 mode.

In the IDLE1 mode, only the internal oscillator operates. The system clock in the MCU stops, and the CLK pin is fixed at the “1” level.

In the HALT state, an interrupt request is sampled asynchronously with the system clock, however the HALT release (restart of operation) is performed synchronously with it.

The interrupts except $\overline{\text{NMI}}$ and INT0 are disabled during this mode.

When the IDLE1 mode is used, set TRUN <PRRUN> to “0” to stop 9, 5 bit prescaler before “HALT” instruction is executed.

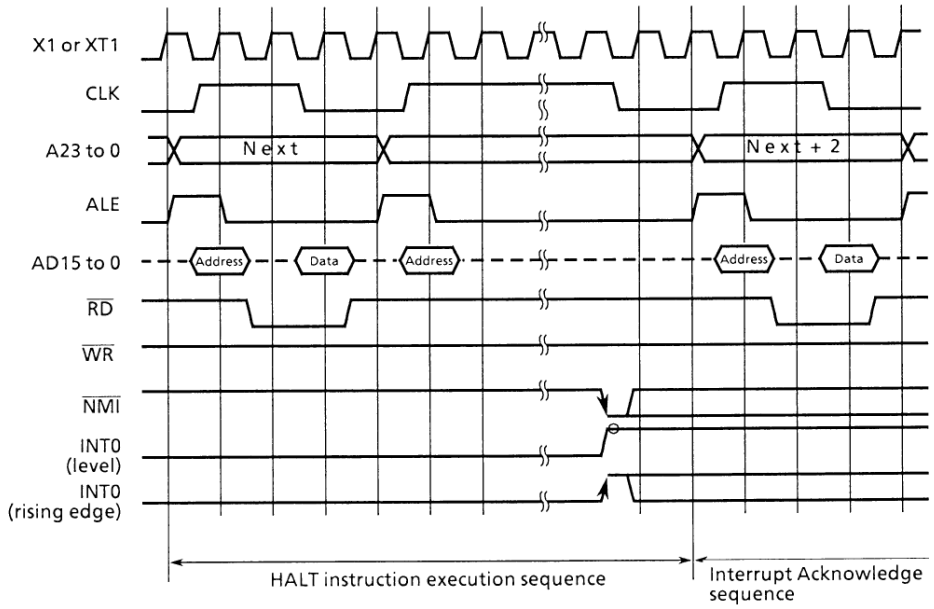


Figure 3.3.6. Timing Chart of HALT Released by Interrupts in IDLE1 Mode

④ STOP mode

Figure 3.3.7 is a timing chart for releasing the HALT state by interrupts in the STOP mode.

The STOP mode is selected to stop all internal circuits including the internal oscillator. In this mode, all pins except the special ones are put in the high-impedance state, independent of the internal operation of the MCU. Table 3.3 (1) summarizes the state of these pins in the STOP mode. Note, however, that the pre-halt state (The status prior to execution of HALT instruction) of all output pins can be retained by setting the internal I/O register WDMOD <DRVE> to "1". The content of this register in initialized to "0" by resetting.

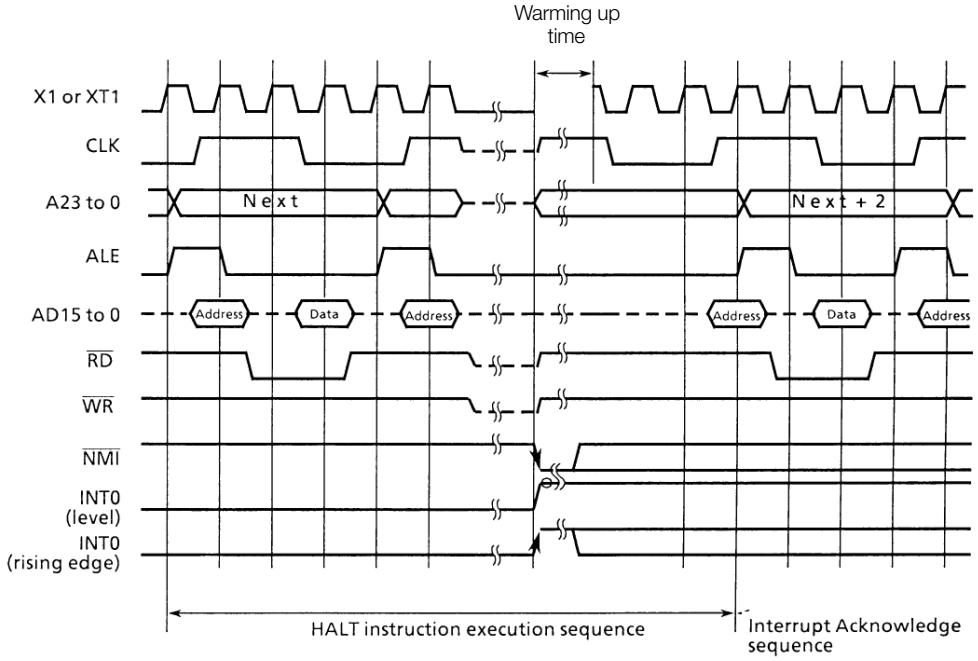


Figure 3.3.7. Timing Chart of HALT Released by Interrupt in STOP Mode

Only either the $\overline{NMI}$, INT0, or $\overline{RESET}$ can release the STOP mode.

When the STOP mode is released except by the $\overline{RESET}$, the system clock starts outputting after warming up time to get the stabilized oscillation.

A warming up time can be set using WDMOD <WARM> bit.

See the example of warming up time in Table 3.3.2.

When the STOP mode is released by $\overline{RESET}$, it is necessary to keep the $\overline{RESET}$ signal at "0" long enough to release to get the stabilized oscillation because the warming up counter is ignored.

The warming up counter operates when the STOP mode is released even when the system which is used as an external oscillator. As a result, it takes warming up time from inputting the releasing request to outputting the system clock.

The NORMAL/SLOW mode selection is possible after released STOP mode.

This is selected by SYSCR0 <RSYSCK> register. Therefore, setting to <RSYSCK>, <RXEN>, <RXTEN> is necessary before "HALT" instruction is executed.

Additionally, setting value to <SYSCK>, <XEN>, <XTEN> are ignored.

(Setting Example)

The STOP mode is entered when the low frequency (fs) operates, and after that high frequency operates after releasing by $\overline{NMI}$.

```
Address
         SYSCR0    EQU   006EH
         SYSCR1    EQU   006FH
         WDMOD     EQU   005CH

9000H    RES   4, (WDMOD)          ; Sets Warming Up Time to 2^14/fc
9002H    LD    (SYSCR0), -11000 - - B ; Operates High Frequency after released.
9005H    HALT
```

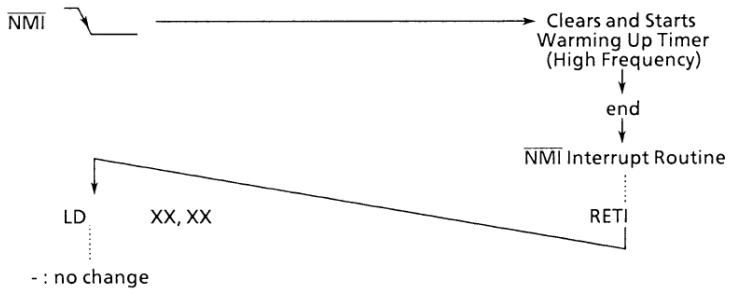


Table 3.3 (1) Pin States in STOP Mode

Pin Name	I/O	TMP93CM40		TMP93CM41	
		DRVE = 0	DRVE = 1	DRVE = 0	DRVE = 1
P0	Input mode / AD0 to 7 Output mode	– –	– Output	– ×	– ×
P1	Input mode / AD8 to 15 Output mode / A8 to 15	– –	– Output	– ×	– ×
P2	Input mode Output mode / A0 to 7, A16 to 23	PD* PD*	PD* Output	PD* PD*	PD* Output
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	Output	–	Output	–	"1" Output
P32 to P37	Input mode Output mode	PU PU	PU Output	←	
P40, P41	Input mode Output mode	PU* PU*	PU Output		
P42 ($\overline{CS2}$ / $\overline{CAS2}$)	Input mode Output mode	PD* PD*	PD Output		
P5	Input	–	–		
P6	Input mode Output mode	PU* PU*	PU Output		
P7	Input mode Output mode	PU* PU*	PU Output		
P80 to P86	Input mode Output mode	PU* PU*	PU Output		
P87 (INT0)	Input mode Output mode	PU PU	PU Output		
P90 to P95	Input mode Output mode	PU* PU*	PU Output		
PA7 to PA0 PA7	Input mode Output mode SCOUT	– – –	– Output "0"		
$\overline{NMI}$	Input	Input	Input		
$\overline{WDTOUT}$	Output	Output	Output		
ALE	Output	"0"	"0"		
CLK	Output	–	"1"		
$\overline{RESET}$	Input	Input	Input		
$\overline{EA}$, AM8/$\overline{16}$	Input	Input	Input		
X1	Input	–	–		
X2	Output	"1"	"1"		
P97 to 96 P96 P97	Input mode Output mode XT1 XT2	– – – "1"	– Output – "1"		

– : Input for input mode / input pin is invalid; output mode / output pin is at high impedance.
Input : Input enable state.
Input : Input gate in operation. Fix input voltage to 0 or 1 so that input pin stays constant.
Output : Output state
PU : Programmable pull-up pin. Fix the pin to avoid through current since the input gate operates when a pull-up pin resistor is not set.
PD : Programmable pull-down pin. Fix the pin like a pull-up pin when a pull-down resistor is not set.
* : Input gate disable state. No through current even if the pin is set to high impedance.
× : Cannot set.

Note: Port registers are used for controlling programmable pull-up / pull-down. If a pin is also used for an output function (eg, TO1) and the output function is specified, whether pull-up or pull-down is selected depends on the output function data. If a pin is also used for an input function, whether pull-up or pull-down is selected depends on the port register setting value only.

Table 3.3 (2) Operation During Halt and How to Release the Halt Mode

Halt mode			RUN	IDLE2	IDLE1	STOP
WDMOD<HALTM1, 0>			00	11	10	01
Block	CPU		Halt			
	I/O port		Keep the state when the "HALT" instruction was executed.			See Table 3.3 (1)
	8 bit Timer		Operate		Stop	
	8 bit PWM Timer					
	16 bit Timer					
	Pattern Generator					
	Serial Interface					
	A/D Converter					
	Watch Dog Timer					
	Interrupt Controller					
Halt Releasing Source	Interrupt	$\overline{NMI}$	○	○	○	○
		INTWD	○	○	—	—
		INT0	○	○	○	○
		INT4, 5, 6, 7	○	○	—	—
		INTT0, 1, 2, 3	○	○	—	—
		INTTR4, 5, 6, 7	○	○	—	—
		INTRX0, TX0	○	○	—	—
		INTRX1, TX1	○	○	—	—
		INTAD	○	—	—	—
	$\overline{RESET}$		○	○	○	○

Note : On condition that your system allows the interruption to insert during HALT (STOP) operation and chooses the different clock source before and after HALT operation, when the system receives the interrupt during HALT (STOP) operation, the oscillation may be chosen the frequency the system operates before HALT operation. If your system avoids this, match the value <SYSCK> and <RSYSCK> before HALT operation.

Note: On the condition that the system allows the interruption to insert during HALT (STOP) operation and chooses the different clock source before and after HALT operation, when the system receives the interrupt during HALT (STOP) operation, the oscillation may be chosen in the frequency the system operates before HALT operation. If your system avoids this, match the value <SYSCK> and <RSYSCK> before HALT operation.

Table 3.3.2 Warming up Time After Releasing the STOP Mode Example

Operation clock after the stop mode	Warming-up time [ms]		Clock
	WDMOD <WARM> = 0	WDMOD <WARM> = 1	
fc fc/2 fc/4 fc/8 fc/16	1.024 2.048 4.096 8.192 16.384	4.096 8.192 16.384 32.768 65.536	fc = 16MHz
fs	500	200	fs = 32.768kHz

3.4 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask flip-flop (IFF2 to 0) and the built-in interrupt controller.

TMP93CM40/M41 have altogether the following 22 interrupt sources:

- Interrupts from the CPU···2
  (Software interrupts, and Illegal (undefined) instruction execution)
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0, and INT4 to 7)···6
- Interrupts from built-in I/Os···14

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority (variable) can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority of 7.

When an interrupt is generated, the interrupt controller sends the value of the priority of the interrupt source to the CPU. When more than one interrupt is generated simultaneously, the interrupt controller sends the value of the highest priority (7 for non-maskable interrupts is the highest) to the CPU.

The CPU compares the value of the priority sent with the value in the CPU interrupt mask register (IFF2 to 0). If the value is greater than that of the CPU interrupt mask register, the interrupt is accepted. The value in the CPU interrupt mask register (IFF2 to 0) can be changed using the EI instruction (contents of the EI num/IFF <2:0> = num). For example, programming EI 3 enables acceptance of maskable interrupts with a priority of 3 or greater, and non-maskable interrupts which are set in the interrupt controller. The DI instruction (IFF <2 : 0> = 7) operates in the same way as the EI 7 instruction. Since the priority values for maskable interrupts are 0 to 6, the DI instruction is used to disable maskable interrupts to be accepted. The EI instruction becomes effective immediately after execution. (With the TLCS-90, the EI instruction becomes effective after execution of the subsequent instruction.)

In addition to the general-purpose interrupt processing mode described above, there is also a high-speed μDMA processing mode. High-speed μDMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

Figure 3.4 (1) is a flowchart showing overall interrupt processing.

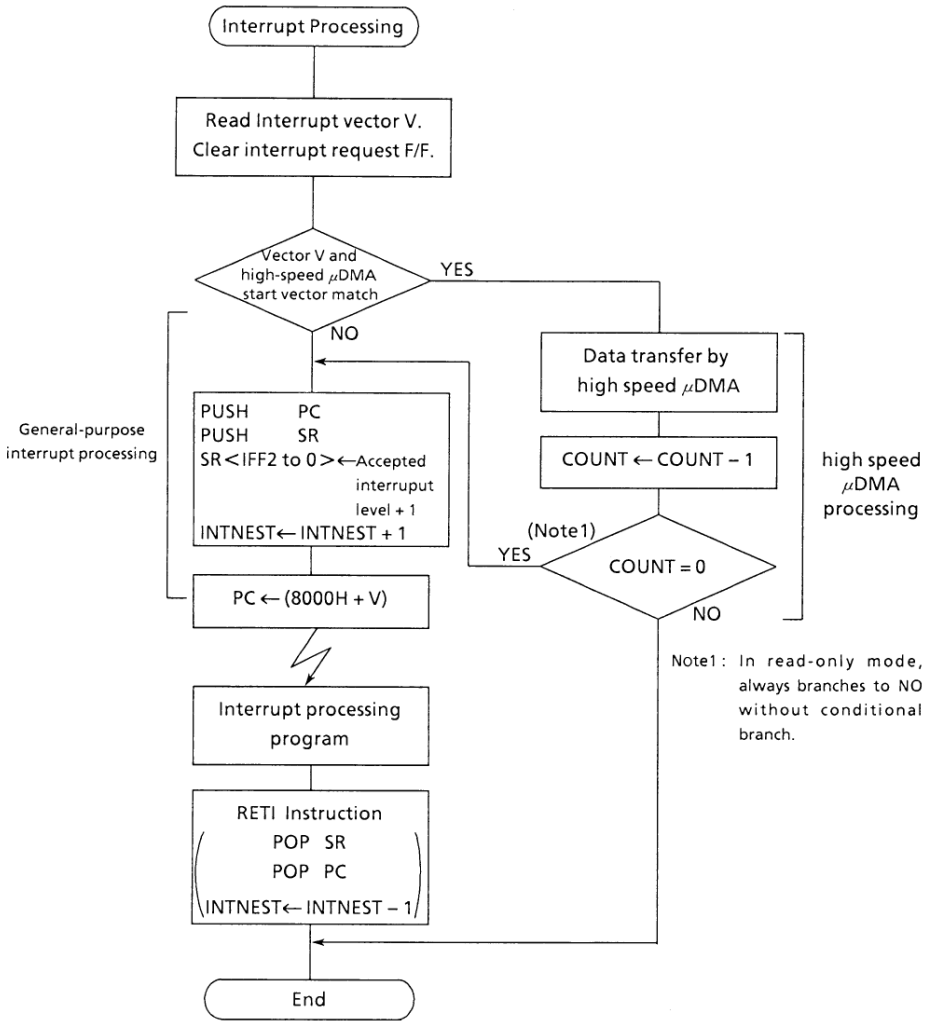


Figure 3.4 (1). Interrupt Processing Flowchart

3.4.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows:

(1) The CPU reads the interrupt vector from the interrupt controller. When more than one interrupt with the same level is generated simultaneously, the interrupt controller generates interrupt vectors in accordance with the default priority (which is fixed as follows: the smaller the vector value, the higher the priority), then clears the interrupt request.

(2) The CPU pushes the program counter and the status register to the system stack area (area indicated by the system mode stack pointer (XSP)).

(3) The CPU sets a value in the CPU interrupt mask register <IFF2 to 0> that is higher by 1 than the value of the accepted interrupt level. However, if the value is 7, 7 is set without an increment.

(4) The CPU increments the INTNEST (Interrupt Nesting Counter).

(5) The CPU jumps to address 8000H + interrupt vector, then starts the interrupt processing routine.

The following diagram shows all the above processing state number.

Bus Width of Stack Area	Bus Width of Interrupt Vector Area	Interrupt Processing State Number	
		MAX mode	MIN mode
8 bit	8 bit	35	31
	16 bit	31	27
16 bit	16 bit	29	27
	8 bit	25	23

To return to the main routine after completion of the interrupt processing, the RETI instruction is usually used. Executing this instruction restores the contents of the program counter and the status registers.

Though acceptance of non-maskable interrupts cannot be disabled by program, acceptance of maskable interrupts can. A priority can be set for each source of maskable interrupts. The CPU accepts an interrupt request with a priority higher than the value in the CPU mask register <IFF2 to 0>. The CPU mask register <IFF2 to 0> is set to a value higher by 1 than the priority of the accepted interrupt. Thus, if an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

The interrupt request with a priority higher than the accepted now interrupt during the CPU is processing above (1) to (5) is accepted before the 1st instruction in the interrupt processing routine, causing interrupt processing to nest. This is the overlapped Non-Maskable interrupt (level “7”).) The CPU does not accept an interrupt of the same level as that of the interrupt being processed.

Resetting initializes the CPU mask registers <IFF2 to 0> to 7; therefore, maskable interrupts are disabled.

The addresses 008000H to 0080FFH (256 bytes) of the TMP93CM40/M41 are assigned for interrupt processing entry area.

Table 3.4 (1) TMP93CM40/M41 Interrupt Table

Default priority	Type	Interrupt source	Vector value "V"	Address refer to vector	High-speed μDMA start vector
1		Reset , or SWI0 instruction	0 0 0 0 H	8 0 0 0 H	–
2		SWI 1 instruction	0 0 0 4 H	8 0 0 4 H	–
3		INTUNDEF : Illegal instruction, or SWI2	0 0 0 8 H	8 0 0 8 H	–
4	Non-	SWI 3 instruction	0 0 0 C H	8 0 0 C H	–
5	maskable	SWI 4 instruction	0 0 1 0 H	8 0 1 0 H	–
6		SWI 5 instruction	0 0 1 4 H	8 0 1 4 H	–
7		SWI 6 instruction	0 0 1 8 H	8 0 1 8 H	–
8		SWI 7 instruction	0 0 1 C H	8 0 1 C H	–
9		$\overline{\text{NMI}}$ Pin	0 0 2 0 H	8 0 2 0 H	08H
10		INTWD : Watchdog timer	0 0 2 4 H	8 0 2 4 H	09H
11		INT0 pin	0 0 2 8 H	8 0 2 8 H	0AH
12		INT4 pin	0 0 2 C H	8 0 2 C H	0BH
13		INT5 pin	0 0 3 0 H	8 0 3 0 H	0CH
14		INT6 pin	0 0 3 4 H	8 0 3 4 H	0DH
15		INT7 pin	0 0 3 8 H	8 0 3 8 H	0EH
–		(Reserved)	0 0 3 C H	8 0 3 C H	0FH
16		INTT0 : 8-bit timer0	0 0 4 0 H	8 0 4 0 H	10H
17		INTT1 : 8-bit timer1	0 0 4 4 H	8 0 4 4 H	11H
18		INTT2 : 8-bit timer2 / PWM0	0 0 4 8 H	8 0 4 8 H	12H
19		INTT3 : 8-bit timer3 / PWM1	0 0 4 C H	8 0 4 C H	13H
20		INTTR4 : 16-bit timer4 (TREG4)	0 0 5 0 H	8 0 5 0 H	14H
21	Maskable	INTTR5 : 16-bit timer4 (TREG5)	0 0 5 4 H	8 0 5 4 H	15H
22		INTTR6 : 16-bit timer5 (TREG6)	0 0 5 8 H	8 0 5 8 H	16H
23		INTTR7 : 16-bit timer5 (TREG7)	0 0 5 C H	8 0 5 C H	17H
24		INTRX0 : Serial receive (Channel.0)	0 0 6 0 H	8 0 6 0 H	18H
25		INTTX0 : Serial send (Channel.0)	0 0 6 4 H	8 0 6 4 H	19H
26		INTRX1 : Serial receive (Channel.1)	0 0 6 8 H	8 0 6 8 H	1AH
27		INTTX1 : Serial send (Channel.1)	0 0 6 C H	8 0 6 C H	1BH
28		INTAD : A/D conversion completion	0 0 7 0 H	8 0 7 0 H	1CH
–		(Reserved)	0 0 7 4 H	8 0 7 4 H	1DH
to		to	to	to	to
–		(Reserved)	0 0 F C H	8 0 F C H	3FH

3.4.2 High-Speed μDMA

In addition to the conventional interrupt processing, the TLCS-900 also has a high-speed μDMA function. When an interrupt is accepted, in addition to an interrupt vector, the CPU receives data indicating whether processing is high-speed μDMA mode or general-purpose interrupt. If high-speed μDMA mode is requested, the CPU performs high-speed μDMA processing.

The TLCS-900 can process at very high speed compared with the TLCS-90 μDMA because it has transfer parameters in dedicated registers in the CPU. Since those dedicated registers are assigned as CPU control registers, they can only be accessed by the LDC instruction.

(1) High-Speed μDMA Operation

High-speed μDMA operation starts when the accepted interrupt vector value matches the μDMA start vector value set in the interrupt controller. The high-speed μDMA has four channels so that it can be set for up to four types of interrupt source.

When a high-speed μDMA interrupt is accepted, data is automatically transferred from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented. If the value in the counter after decrementing is other than 0, high-speed μDMA processing is completed. If the value in the counter after decrementing is 0, general-purpose interrupt processing is performed. In read-only mode, which is provided for DRAM refresh, the value in the counter is ignored and dummy read is repeated.

32-bit control registers are used for setting transfer source/destination addresses. However, the TLCS-900 has only 24 address pins for output. A 16M-byte space is available for the high-speed μDMA.

There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore, data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes, see the description of transfer mode registers.

The transfer counter has 16 bits, so up to 65536 transfers (the maximum when the initial value of the transfer counter is 0000H) can be performed for one interrupt source by high-speed μDMA processing.

Interrupt sources processed by high-speed μDMA processing are those with the high-speed μDMA start vectors listed in Table 3.4 (1).

The following timing chart is a high-speed μDMA cycle of the Transfer Address Increment mode (the other mode except the Read-only mode is same as this).

(Condition: MIN mode, 16bit Bus width for 16M Byte, 0 wait).

Setting to Reset / Interrupt Vector

① Reset Vector

8000H	PC (7:0)
8001H	PC (15:8)
8002H	PC (23:16)
8003H	XX

Though the register mode is the maximum (MAX) mode after reset, The Reset Vector must be defined with in 64K byte area form 0000H to FFFFH.

② Interrupt Vector (except Reset Vector)

Address refer to vector		
	+0	PC (7:0)
	+1	PC (15:8)
	+2	PC (23:16)
	+3	XX

XX : don't care

(Setting Example)

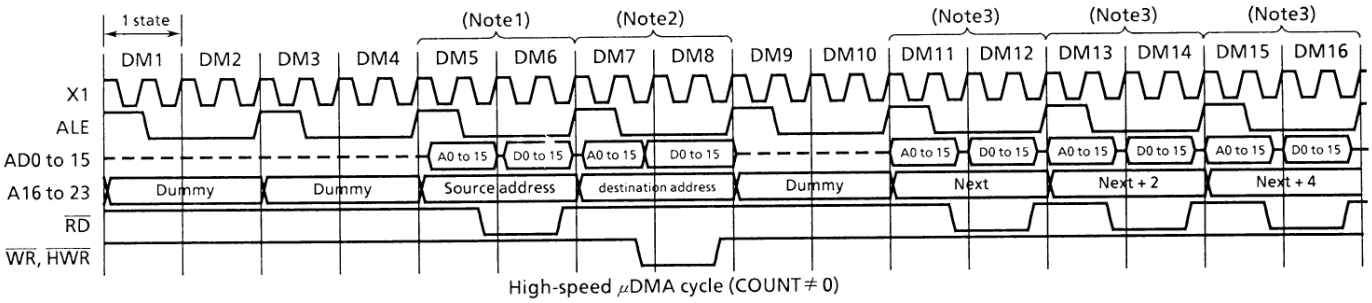
Reset Vectir : 8100H, $\overline{NMI}$ Vector : 9ABCH, INTAD Vector : 123456h.

```
                                 (The value "99H" as a vector is for the explaining
                                  and it does not have a special meaning.)
ORG     8000H
DL      99008100H       ; Reset = 8100H

ORG     8020H
DL      99009ABCH       ; NMI = 9ABCH

ORG     8070H
DL      99123456H       ; INTAD = 123456H

ORG     8100H
LD      A, B                    (cf)
  :                               ORG, DL are the Assembler Directive.
ORG     9ABCH                     ORG : control location counter
LD      B, C                      DL  : define the long word (32 bits) data
  :
ORG     123456H
LD      C, A
  :
```

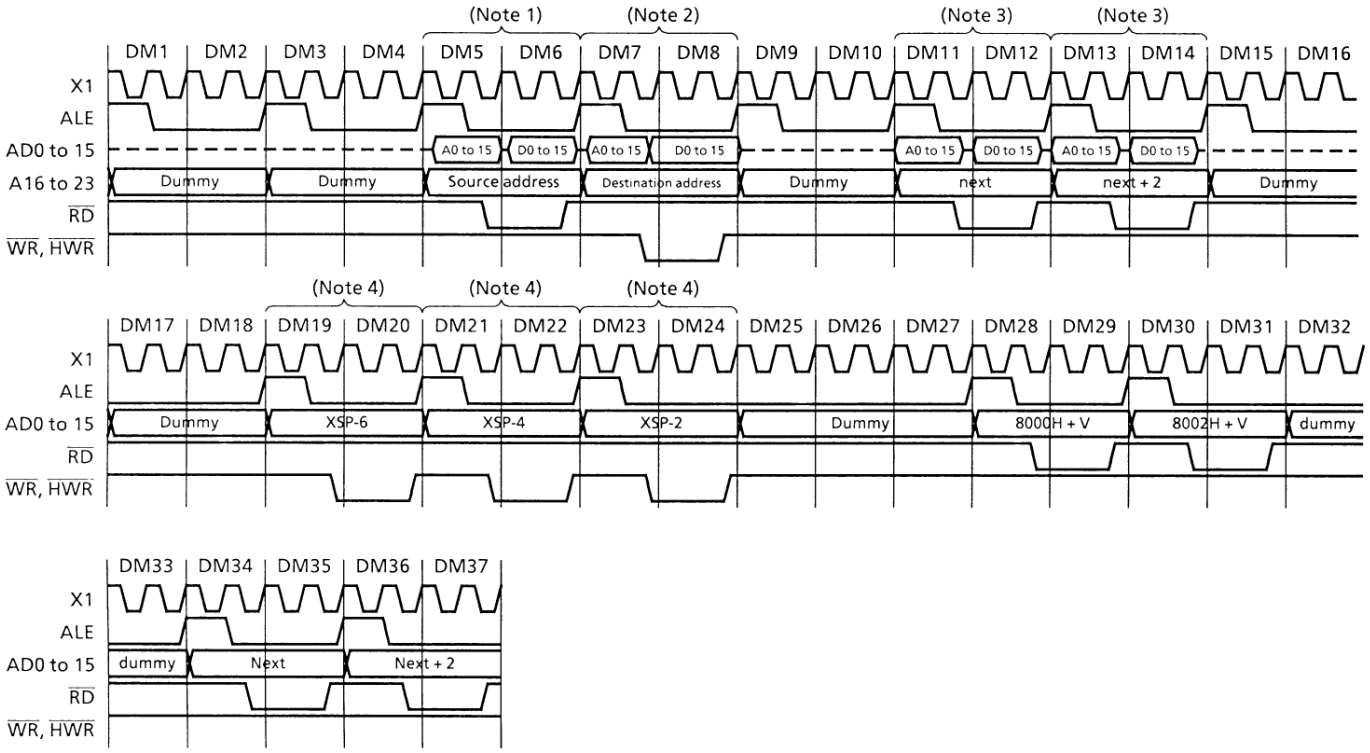


High-speed μDMA cycle (COUNT ≠ 0)

(Note 1) This is added 2 states the case of the bus width of source address area is 8 bit.

(Note 2) This added 2 states the case of the bus width of destination address area is 8 bit.

(Note 3) This may be a dummy cycle with instruction queue buffer.



High-speed μDMA cycle (COUNT = 0)

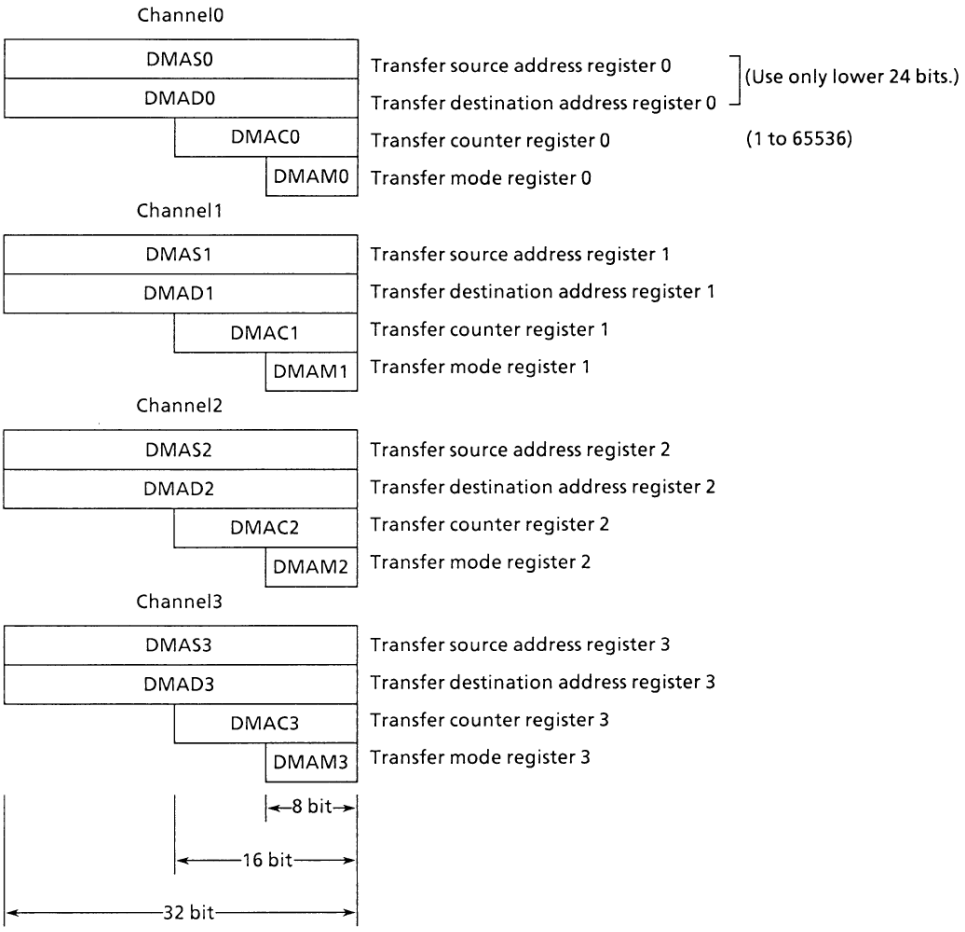
(Note 1) This is added 2 states the case of the bus width of source address area is 8 bit.

(Note 2) This added 2 states the case of the bus width of destination address area is 8 bit.

(Note 3) This be a dummy cycle with instruction queue buffer.

(Note 4) This is added 2 states the case of the bus width of stack address area is 8 bit.

(2) Register Configuration (CPU Control Register)



These Control Registers cannot be set only “LCD cr, r” instruction.

(3) Transfer Mode Register Details

(DMAM0 to 3)

0	0	0	0	Mode

Note : When setting values for this register, set the upper 4 bits to 0.

Z: 0 = byte transfer, 1 = word transfer

					execution time (Min) at 20 MHz
0	0	0	Z	Transfer destination address INC mode ..... for I/O to memory (DMADn +) ← (DMASn) DMACn←DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode ..... for I/O to memory (DMADn −) ← (DMASn) DMACn←DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode ..... for I/O to memory (DMADn) ← (DMASn +) DMACn←DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode ..... for I/O to memory (DMADn) ← (DMASn −) DMACn←DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	0	Z	Fixed address mode ..... I/O to I/O (DMADn) ← (DMASn) DMACn←DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	1	0	Read-only mode ..... for DRAM refresh Dummy← (DMASn) ; Reads 4 bytes. DMASn←DMASn + 4 ; Increments lower word only. DMACn←DMACn − 1	14 states (1.4 μs)
1	0	1	1	Counter mode ..... for interrupt counter DMASn←DMASn + 1 DMACn←DMACn − 1 if DMACn = 0 then INT.	11 states (1.1 μs)

(1 states = 100 ns at 20 MHz, High frequency mode)

Note : n : corresponds to high-speed μDMA channels 0 to 3.
DMADn + / DMASn + : Post-increment (Increments register value after transfer.)
DMADn − / DMASn − : Post-decrement (Decrement register value after transfer.)

Do not use undefined codes for transfer mode control.

<Example for Usage of read only mode (DRAM refresh)>

* Clock Condition

System Clock: fc
Clock Gear : 1 (fc)

When the hardware configuration is as follows:

DRAM mapping size: = 1MB
DRAM data bus size: = 8 bits
DRAM mapping address range: = 100000H to 1FFFFFH

Set the following registers first; refresh is performed automatically.

① Register initial value setting

```
LD      XIX, 100000H
LDC     DMAS0,  XIX     ···mapping start address
LD      A, 00001010B
LDC     DMAM0,  A       ···read only mode (for
                           DRAM refresh)
```

② Timer Setting

Set the timers so that interrupts are generated at intervals of 62.5μs or less.

③ Interrupt controller setting

Set the timer interrupt mask higher than the other interrupt mask. Write the above timer interrupt vector value in the High-Speed μDMA start vector register, DMA0V.

(Operation description)

The DRAM data bus is an 8-bit bus and the μDMA is in read-only mode (4 bytes), so refresh is performed four times per interrupt.

When a 512 refresh/8ms DRAM is connected, DRAM refresh is performed sufficiently if the μDMA is started every 15.625μs x 4 = 62.4μs or less, since the timing is 15.625μs/refresh.

(Overhead)

Each processing time by the High-Speed μDMA is 1.8μs (18 states) at 20MHz with an 8-bit data bus.

In the above example, the micro DMA is started every 62.5μs, 1.8μs/62.5μs = 0.029; thus, the overhead is 2.88%.

(Note)

When the Bus is released ($\overline{\text{BUSAK}}$ = "0") which must wait to accept the interrupt, DRAM refresh is not performed because of the high-speed μDMA is generated by an interrupt.

3.4.3 Interrupt Controller

Figure 3.4.3 (1) is a block diagram of the interrupt circuits. The left half of the diagram shows the interrupt controller; the right half includes the CPU interrupt request signal circuit and the HALT release signal circuit.

Each interrupt channel (total of 20 channels) in the interrupt controller has an interrupt request flip-flop, interrupt priority setting register, and a register for storing the high-speed μDMA start vector. The interrupt request flip-flop is used to latch interrupt requests from peripheral devices. The flip-flop is cleared to 0 at reset, when the CPU reads the interrupt channel vector after the acceptance of interrupt, or when the CPU executes an instruction that clears the interrupt of that channel (writes 0 in the clear bit of the interrupt priority setting register).

For example, to clear the INT0 interrupt request, set the register after the DI instruction as follows.

INTE0AD←----- 0 --- Zero-clears the INT0 Flip-Flop.

The status of the interrupt request flip-flop is detected by reading the clear bit. Detects whether there is an interrupt request for an interrupt channel.

The interrupt priority can be set by writing the priority in the interrupt priority setting register (e.g., INTE0AD, INTE45, etc.) provided for each interrupt source. Interrupt levels to be set are from 1 to 6. Writing 0 or 7 as the interrupt priority disables the corresponding interrupt request. The priority of the non-maskable interrupt ($\overline{\text{NMI}}$ pin, watchdog timer, etc.) is fixed to 7. If interrupt requests with the same interrupt level are generated simultaneously, interrupts are accepted in accordance with the default priority (the smaller the vector value, the higher the priority).

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 to 0> set in the Status Register by the interrupt request signal with the priority value sent; 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 to 0>. Interrupt requests 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 to 0>.

The interrupt controller also has four registers used to store the high-speed μDMA start vector. These are I/O registers; unlike other DMA registers (DMAS, DMAD, DMAM, and DMAC), they can be accessed in either normal or system mode. Writing the start vector of the interrupt source for the μDMA processing (see Table 3.4 (1)), enables the corresponding interrupt to be processed by μDMA processing. The values must be set in the μDMA parameter registers (e.g., DMAS and DMAD) prior to the μDMA processing.

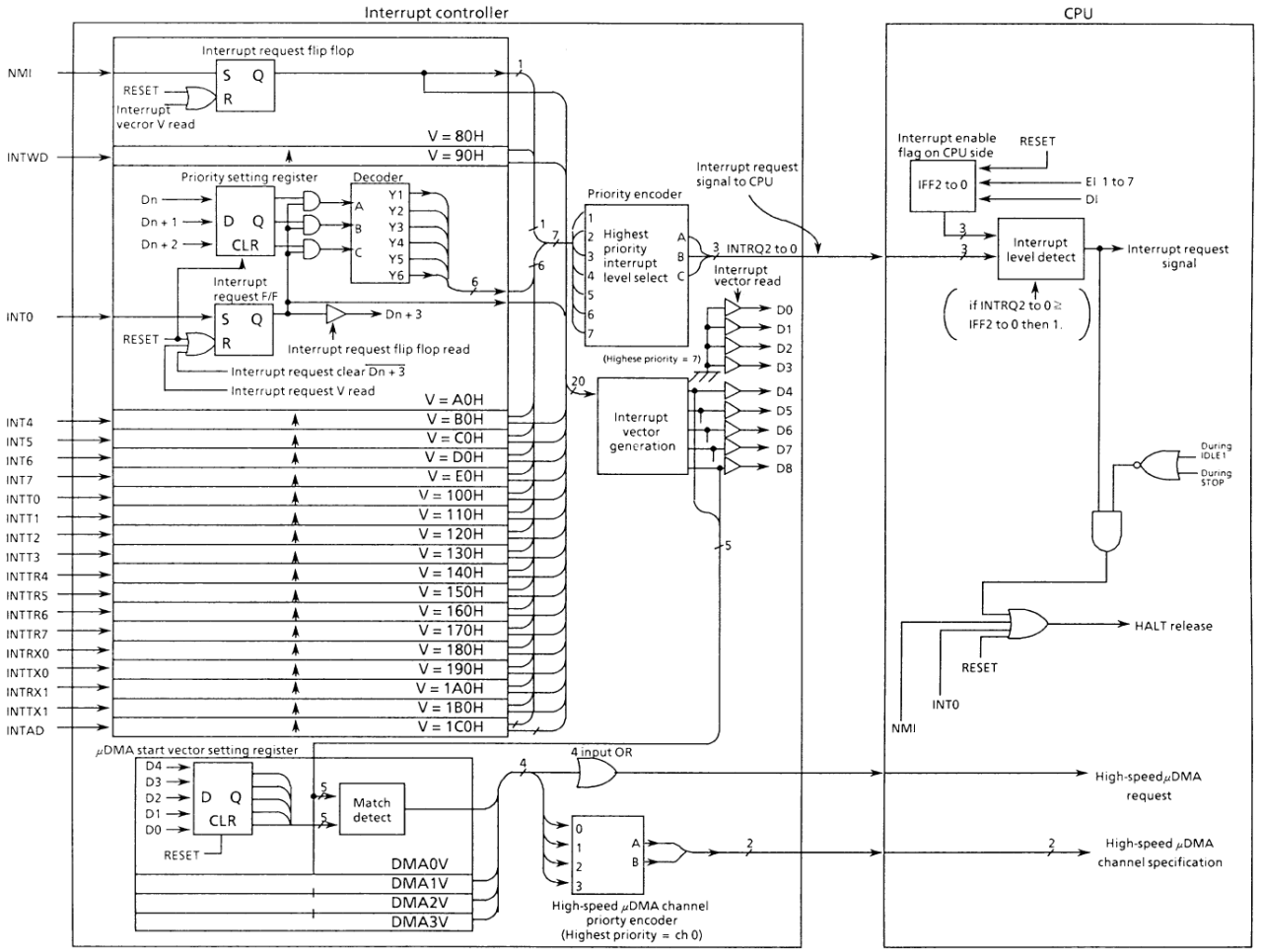


Figure 3.4.3 (1). Block Diagram of Interrupt Controller

(1) Interrupt Priority Setting Register

(Read-modify-write is prohibited.)

Symbol	Address	7	6	5	4	3	2	1	0	
INTE0AD	0070H	INTAD				INT0				←Interrupt source
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0	←bit Symbol
		R/W	W			R/W	W			←Read / Write
		0	0	0	0	0	0	0	0	←After reset
INTE45	0071H	INT5				INT4				
		I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE67	0072H	INT7				INT6				
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET10	0073H	INTT1 (Timer1)				INTT0 (Timer0)				
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTEPW10	0074H	INTT3 (Timer3 / PWM1)				INTT2 (Timer2 / PWM0)				
		IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET54	0075H	INTTR5 (TREG5)				INTTR4 (TREG4)				
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET76	0076H	INTTR7 (TREG7)				INTTR6 (TREG6)				
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES0	0077H	INTTX0				INTRX0				
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES1	0078H	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	

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

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	----- Don't care -----

* Note about clearing interrupt request flag

The interrupt request flag of INTAD, INTRX0, INTRX1 are not cleared by writing “00” to IXXC because they are level interrupts.

They can be cleared only by resetting or reading ADREGn/SCBUFn.

(2) External Interrupt Control

Interrupt Input Mode Control Register

IIMC (007BH)

	7	6	5	4	3	2	1	0
bit Symbol						WUEF	PRCK1	PRCK0
Read / Write								
After reset						0	0	0
Function						1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Can be accepted in $\overline{NMI}$ rising edge.

Read-modify-write is prohibited.

INT0 input enable (Note)

0	INT0 disable (P87 function only)
1	Input enable

$\overline{NMI}$ rising edge enable

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

INT0 level enable

0	Rising edge detect interrupt
1	High level interrupt

Note1 : The INT0 pin can also be used for standby release as described later. Even if the pin is not used for standby release, setting this register to “0” maintains the port function during standby mode.

Note2 : Case of changing from level to edge for INT0 pin mode (<I0LE>“1”→“0”)

Execution example:

```
LD  (INTE0AD), xxxx0000B    ; INT0 disenable,clean the request flag
LD  (IIMC)   , xxxxx10xB    ;Change from level to edge
LD  (INTE0AD), xxxx0nnnB    ; Set interrupt levle “n” for INT0, clear
                              the request flag
```

Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
$\overline{NMI}$	—	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	P87	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 1
INT4	P80	Rising edge	T4MOC<CAP12M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T4MOD<CAP12M1, 0> = 1, 0
INT5	P81	Rising edge	———
INT6	P84	Rising edge	T5MOC<CAP34M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T5MOD<CAP34M1, 0> = 1, 0
INT7	P85	Rising edge	———

(3) High-Speed μDMA Start Vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector with each channel's μDMA start vector (bits 4 to 8 of the interrupt vector). When both match, the interrupt is processed in μDMA mode for the channel whose value matched.

If the interrupt vector matches more than one channel, the channel with the lower channel number has a higher priority.

High Speed μDMA0 Start Vector (read-modify-write is not possible.)

DMA0V (007CH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
Read / Write				W				
After reset				0	0	0	0	0

High Speed μDMA1 Start Vector (read-modify-write is not possible.)

DMA1V (007DH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
Read / Write				W				
After reset				0	0	0	0	0

High Speed μDMA2 Start Vector (read-modify-write is not possible.)

DMA2V (007EH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
Read / Write				W				
After reset				0	0	0	0	0

High Speed μDMA3 Start Vector (read-modify-write is not possible.)

DMA3V (007FH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
Read / Write				W				
After reset				0	0	0	0	0

(4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently of each other. Therefore, if the instruction used to clear an interrupt request flag of an interrupt is fetched before the interrupt is generated, it is possible that the CPU might execute the fetched instruction to clear the interrupt request flag while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector “0028H” and start the interrupt processing from the address “8028H”.

To avoid this, make sure that the instruction used to clear the interrupt request flag comes after the DI instruction.

3.5 Functions of Ports

The TMP93CM40/TMP96PM40 has 79 bits for I/O ports. The TMP93CM41 has 61 bits for I/O ports because Port0, Port1, P30, and P31 are dedicated pins for AD0 to 7, AD8 to 15, $\overline{RD}$, and $\overline{WR}$.

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 lists the function of each port pin.

(R: ↑ = With programmable pull-up resistor
↓ = WIth programmable pull-down)

Table 3.5 Functions of Ports

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port0	P00 to P07	8	I/O	–	Bit	AD0 to AD7
Port1	P10 to P17	8	I/O	–	Bit	AD8 to AD15/ A8 to A15
Port2	P20 to P27	8	I/O	↓	Bit	A0 to A7/ A16 to A23
Port 3	P30	1	Output	–	(Fixed)	$\overline{RD}$
	P31	1	Output	–	(Fixed)	$\overline{WR}$
	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	$\overline{RAS}$
Port4	P40	1	I/O	↑	Bit	$\overline{CS0}$ / $\overline{CAS0}$
	P41	1	I/O	↑	Bit	$\overline{CS1}$ / $\overline{CAS1}$
	P42	1	I/O	↓	Bit	$\overline{CS2}$ / $\overline{CAS2}$
Port5	P50 to P57	8	Input	–	(Fixed)	AN0 to AN7
Port6	P60 to P67	8	I/O	↑	Bit	PG00 to PG03, PG10 to PG13
Port7	P70	1	I/O	↑	Bit	TI0
	P71	1	I/O	↑	Bit	TO1
	P72	1	I/O	↑	Bit	TO2
	P73	1	I/O	↑	Bit	TO3
Port8	P80	1	I/O	↑	Bit	T14/INT4
	P81	1	I/O	↑	Bit	T15/INT5
	P82	1	I/O	↑	Bit	TO4
	P83	1	I/O	↑	Bit	TO5
	P84	1	I/O	↑	Bit	TI6 / INT6
	P85	1	I/O	↑	Bit	TI7 / INT7
	P86	1	I/O	↑	Bit	TO6
	P87	1	I/O	↑	Bit	INT0
Port9	P90	1	I/O	↑	Bit	TXD0
	P91	1	I/O	↑	Bit	RXD0
	P92	1	I/O	↑	Bit	$\overline{CTS0}$
	P93	1	I/O	↑	Bit	TXD1
	P94	1	I/O	↑	Bit	RXD1
	P95	1	I/O	↑	Bit	SCLK1
	P96	1	I/O	–	Bit	XT1
	P97	1	I/O	–	Bit	XT2
PortA	PA7 to PA0	8	I/O	–	Bit	SCOUT (PA7)

Resetting makes the port pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output function as input ports except P96/XT1, P97/XT2.

To set port pins for built-in functions, a program is required.

Since the TMP96C141 has an external ROM, some ports are permanently assigned to the CPU.

- P00 to P07 → AD0 to AD7
- P10 to P17 → AD8 to AD15
- P30 → $\overline{RD}$
- P31 → $\overline{WR}$

* Notes about the bus release and programmable pull-up/down I/O ports:

When buses are released ($\overline{BUSAK}$ = 0), the TMP96C141/TMP96CM40/TMP96PM40 sets the output buffer for AD0 to AD15, A0 to A23, and bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$/$\overline{CAS0}$ - $\overline{CS2}$/$\overline{CAS2}$) to off to set them to high impedance. The internal programmable pull-up/pull-down resistors continue to operate. Resistors are programmable only for operations in input mode; not in output mode.

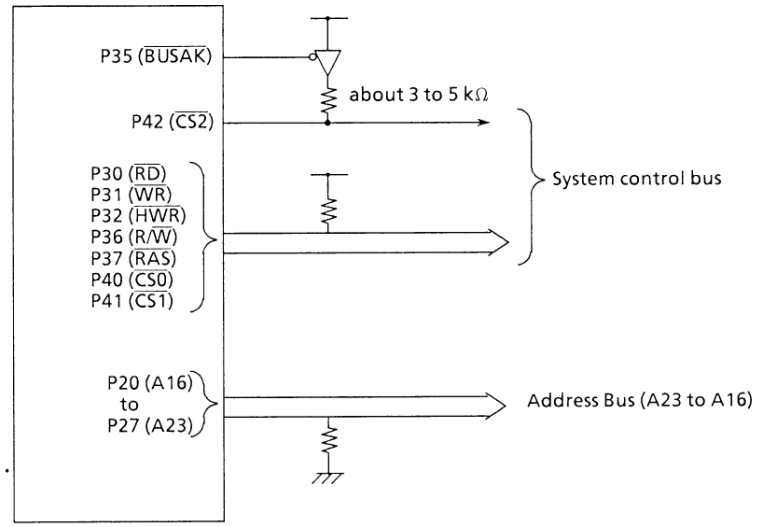
Pin states at bus release are shown below.

Pin Name	Pin state at bus release	
	Port mode	Function mode
P00 to P07 (AD0 to AD7) P10 to P17 (AD8 to AD15)	The state is not changed (does not become high-impedance (HZ).)	becomes high-impedance (HZ)
P30 ($\overline{RD}$) P31 ($\overline{WR}$)	becomes high-impedance (HZ)	←
P32 ($\overline{HWR}$) P37 ($\overline{RAS}$)	The output buffer to OFF. The programmable pull-up resistor is ON only when the output latch is equal to “1”.	The output buffer to OFF. The programmable pull-up resistor is ON regardless of the output latch.
P36 (R/$\overline{W}$) P40 ($\overline{CS0}$/$\overline{CAS0}$) P41 ($\overline{CS1}$/$\overline{CAS1}$)	↑	The output buffer to OFF. The programmable pull-down resistor is undefined.
P42 ($\overline{CS2}$/$\overline{CAS2}$)	The output buffer to OFF. The programmable pull-up resistor is ON only when the output latch is equal to “0”.	The output buffer to OFF. The programmable pull-down resistor is undefined.
P20 to P27 (A16 to A23)	The state is not changed. (does not become high-impedance (HZ).)	The output buffer to OFF. The programmable pull-up resistor is ON only when the output latch is equal to “0”.

The following are the example of the interface circuit of the above pins when the bus releasing function is used.

When the bus is released, both internal memory and internal I/O cannot 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 set the detection time of WDT.



Example of external bus interface using bus release function.

The above circuit is necessary to fix the signal level when the bus is released.

Reset sets P30 ($\overline{RD}$), P31 ($\overline{WR}$) to output, P40 ($\overline{CS0}$), P41 ($\overline{CS1}$), P32 ($\overline{HWR}$), P36 (R/$\overline{W}$), P37 ($\overline{RAS}$), and P35 ($\overline{BUSAK}$) are set to input mode using a pull-up resistor, P42 ($\overline{CS2}$) and P20 to P27 (A16 to 23) to input with pull-down resistor.

The above circuit is necessary to fix the signal level after reset because of the external pull-up resistor collisions with the internal pull-down resistor.

The value of this external pull-up resistor value must be 3 to 5 kΩ. (The value of the internal pull-down resistor is about 50 to 150kΩ)

P20 to 27 (A16 to 23) also needs a circuit like circuit P42 ($\overline{CS2}$) to fix the signal level.

But for the P20 to P27 (A16 to 23) which does not have the means (“L” is active), add pull down directly like the above circuit.

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 control register 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 7). To access external memory, Port 0 functions as an address data bus (AD 0 to 7) and all bits of the control register P0CR are cleared to 0.

With the TMP9eCM041, which needs external ROMs, Port 0 always functions as an address data bus (AD0 to 7) regardless of the value set in control register P0CR.

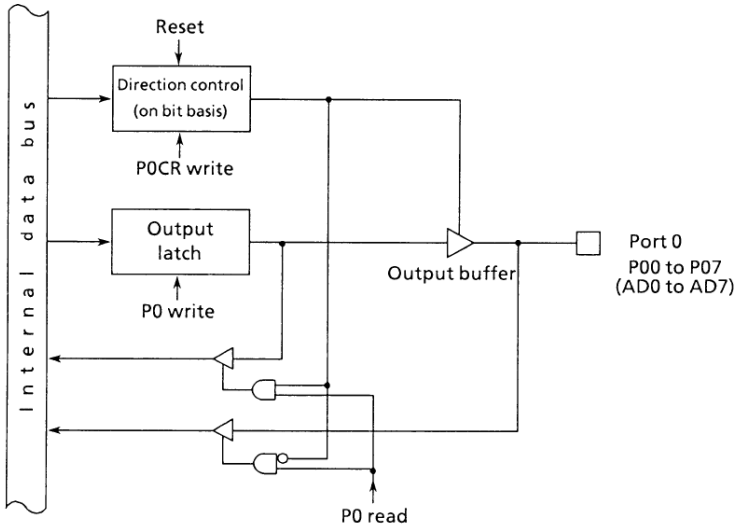


Figure 3.5 (1). Port 0

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 15) or an address bus (A8 to 15).

With the TMP93CM41, which needs external ROMs, Port 1 always functions as an address data bus (AD8 to 15) (AM8/$\overline{16}$ = "0"), as an address bus (A8 to 15) (AM8/$\overline{16}$ = "1") regardless of the value set in control register P1CR.

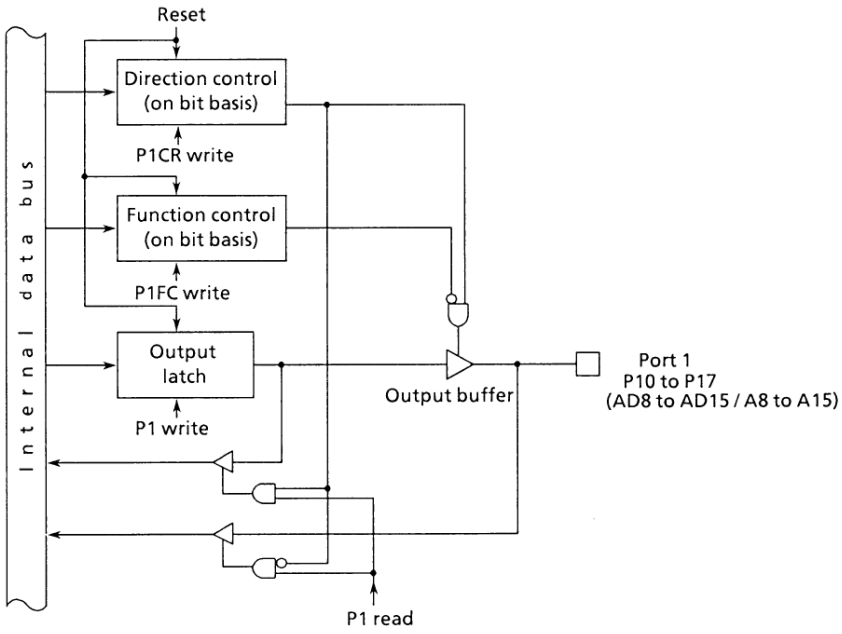


Figure 3.5 (2). Port 1

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	Input mode (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	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:IN 1:OUT (At external access, Port 0 becomes AD7 to 0 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	Input mode (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	R/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	R/W							
After reset	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00 : IN, 01 : OUT, 10 : AD15 to 8, 11 : A15 to 8							

Port 1 function setting

P1CR<P1XC> \ P1FC<P1XF>	0	1
0	Input port	Address data bus (AD15 to 8)
1	Output port	Address bus (A15 to 8)

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

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

Figure 3.5 (3). 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 resets all bits of output latch P2, control register P2CR and function register P2FC to 0. It also sets Port 2 to input mode and connects a pull-down resistor. To disconnect the pull-down resistor, write 1 in the output latch.

In addition to functioning as a general-purpose I/O port, Port 2 also functions as an address data bus (A0 to 7) and an address bus (A16 to 23).

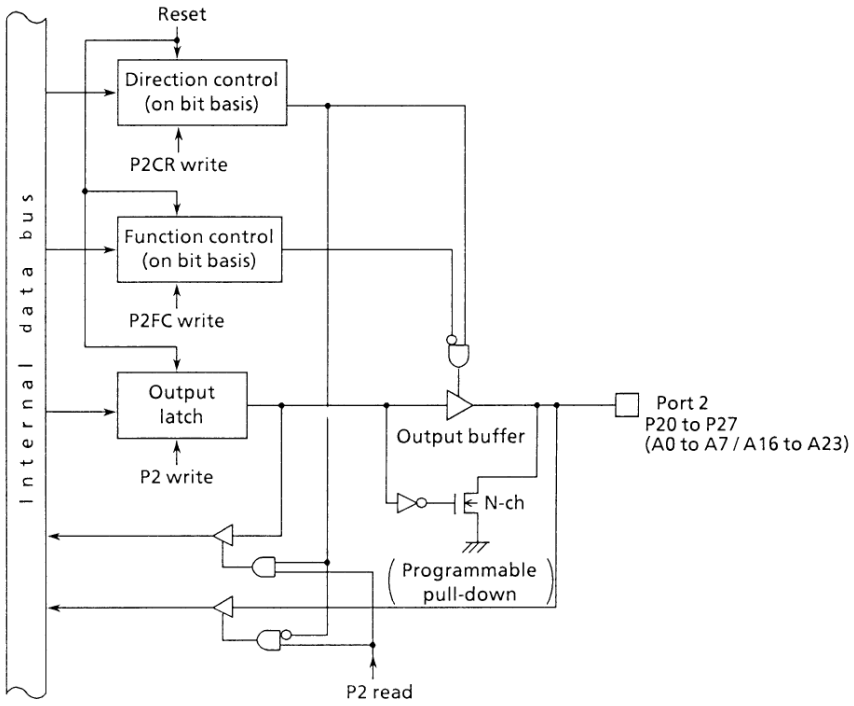


Figure 3.5 (4). Port 2

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	Input mode (Output latch register is cleared to "0".)							

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 : IN, 01 : OUT, 10 : A7 to 0, 11 : A23 to 16							

→ Port 2 function setting

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

P2CR<P2XC> \ P2FC<P2XF>	0	1
0	Input port	Address data bus (A7 to 0)
1	Output port	Address bus (A23 to 16)

Note : <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR.
To set as an address bus A23 to 16, set P2FC after setting P2CR.

Figure 3.5 (5). 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 resets all bits of output latch P3, 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.

With the TMP96C140, 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.

With the TMP93CM41, which comes with an external ROM, Port 30 outputs the $\overline{RD}$ signal; P31, the $\overline{WR}$ signal, regardless of the values set in function registers P30F and P31F.

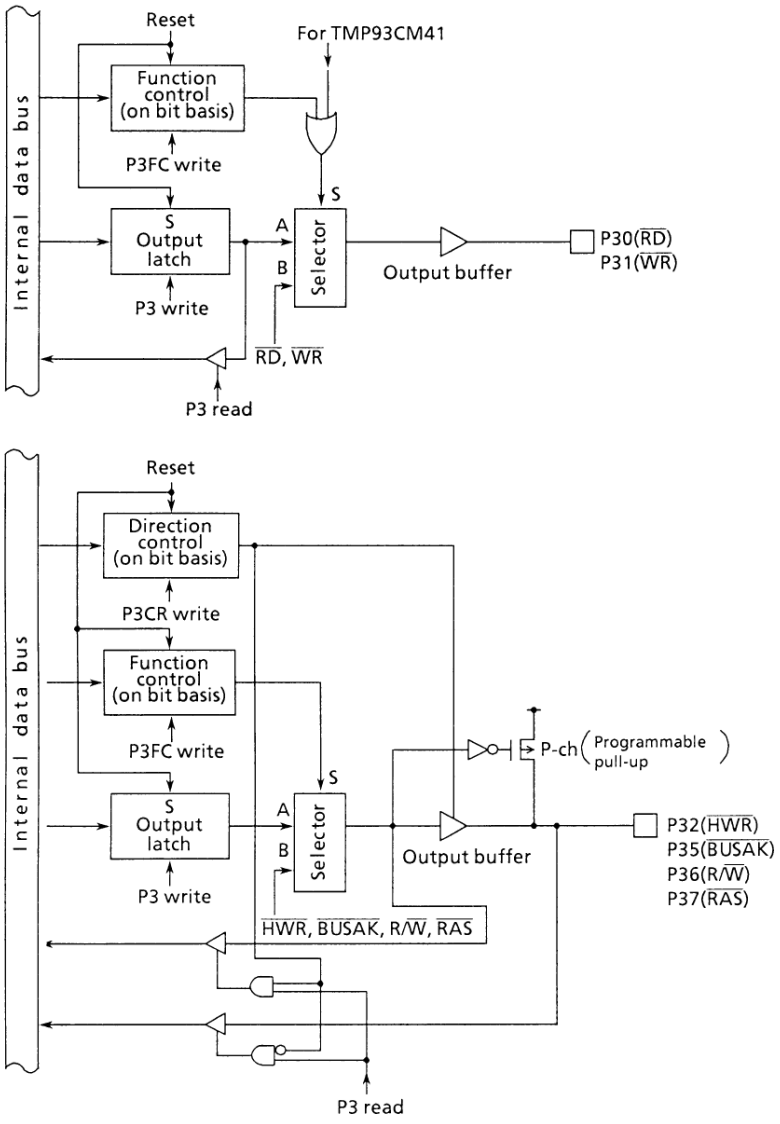


Figure 3.5 (6). Port 3 (P30, P31, P32, P35, P36, P37)

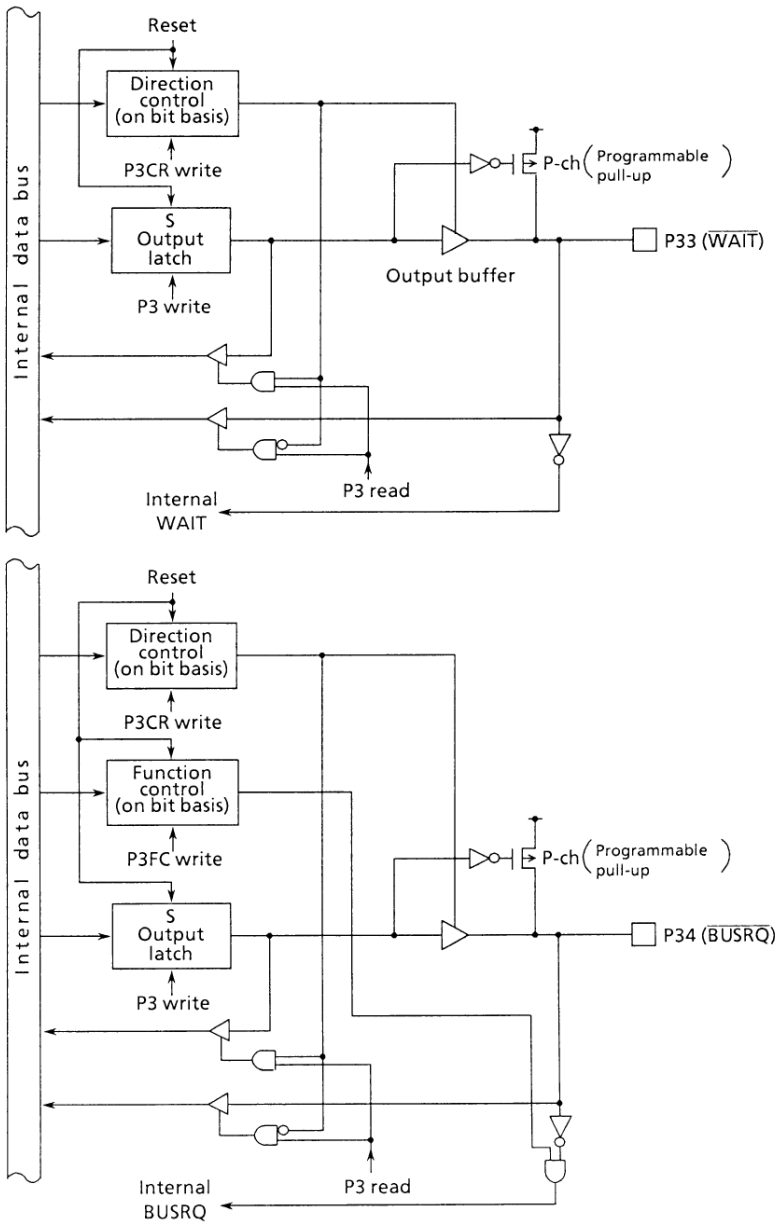
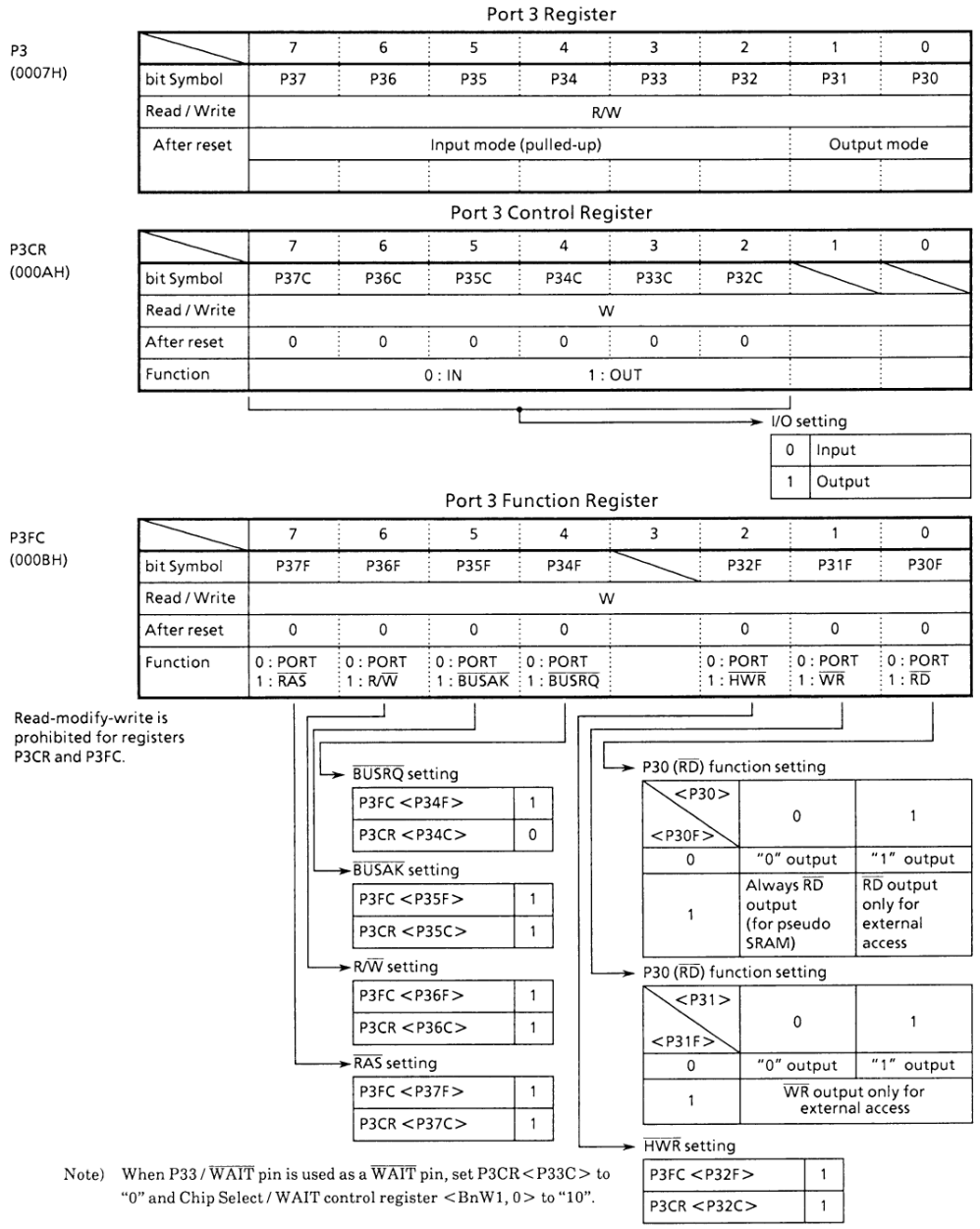


Figure 3.5 (7). Port 3 (P33, P34)



Note) When P33 / $\overline{WAIT}$ pin is used as a $\overline{WAIT}$ pin, set P3CR<P33C> to "0" and Chip Select / WAIT control register <BnW1, 0> to "10".

Figure 3.5 (8). Registers for Port 3

3.5.5 Port 4 (P40 to P42)

Port 4 is a 3-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 the P40 and P42 output latch registers to 1.
- Resets all bits of the P42 output latch register, the control register P4CR, and the function register P4FC to 0.
- Sets P40 and P41 to input mode and connects a pull-up resistor.
- Sets P42 to input mode and connects a pull-down 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 $\overline{CS2}$ or $\overline{CAS0}$ to $\overline{CAS2}$).

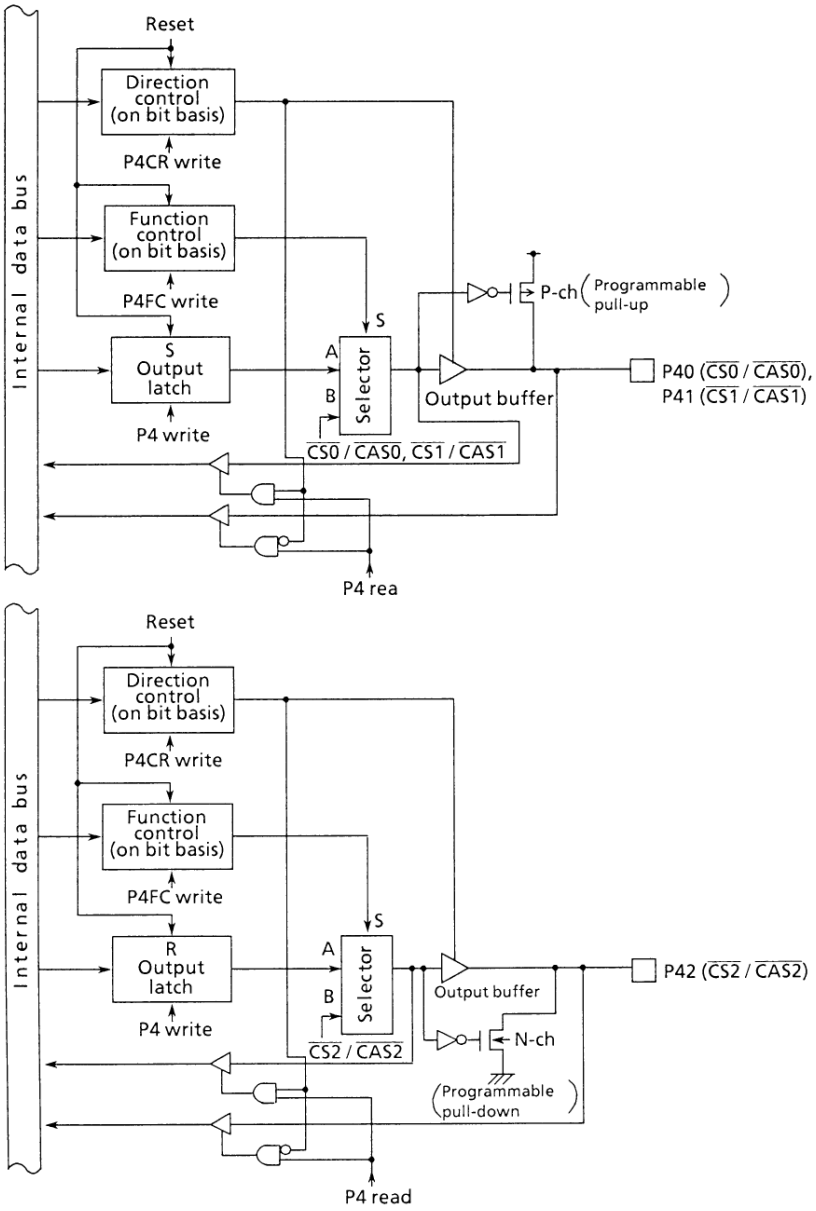
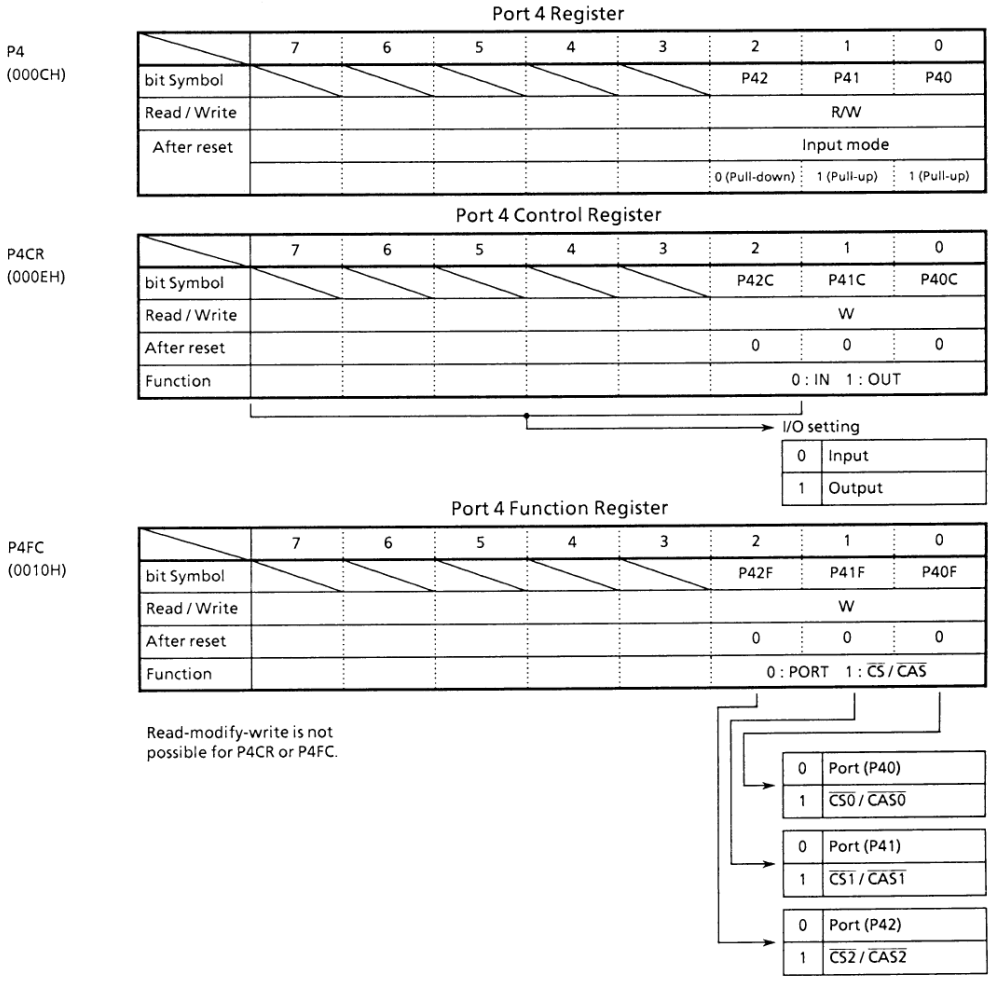


Figure 3.5 (9). Port 4



Note : To output chip select signal ($\overline{CS0}$ / $\overline{CAS0}$ to $\overline{CS2}$ / $\overline{CAS2}$), set the corresponding bits of the control register P4CR and the function register P4FC.
The B0CS, B1CS, and B2CS registers of the chip select / wait controller are used to select the $\overline{CS}$ / $\overline{CAS}$ function.

Figure 3.5 (10). 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 A/D Converter.

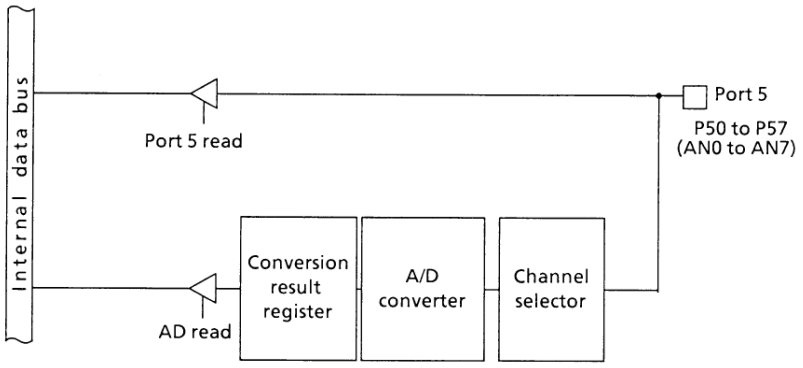


Figure 3.5 (11). Port 5

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	Input mode							

Note) The input channel selection of A/D Converter is set by A/D Converter mode register ADMOD2.

Figure 3.5 (12). Registers for Port 5

3.5.7 Port 6 (P60 to P67)

Port 6 is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 6 as an input port and connects a pull-up resistor. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, Port 6 also functions as a pattern generator PG0/PG1 output. PG0 is assigned to P60 to P63; PG1, to P64 to P67. Writing 1 in the corresponding bit of the port 6 function register (P6FC) enables PG output. Resetting resets the function register P6FC value to 0, and sets all bits to ports.

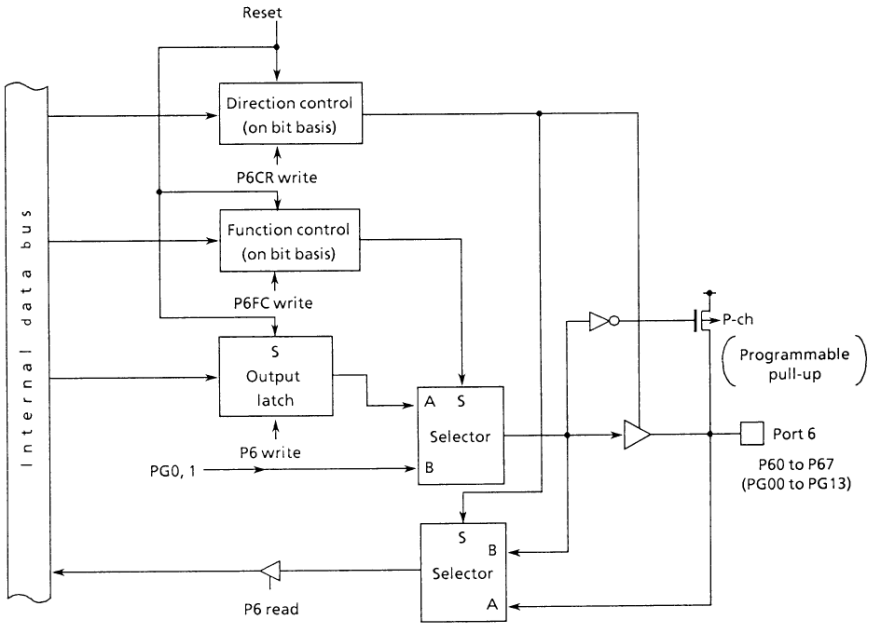


Figure 3.5 (13). Port 6

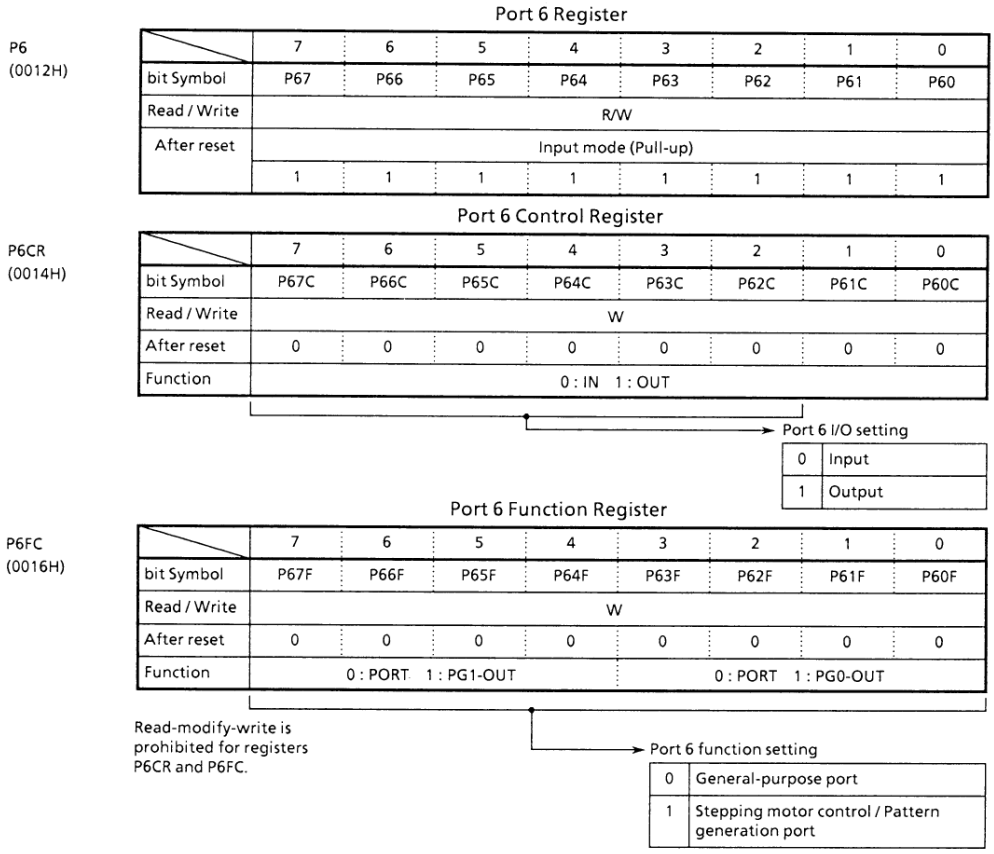


Figure 3.5 (14). Registers for Port 6

3.5.8 Port 7 (P70 to P73)

Port 7 is a 4-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 7 as an input port and connects a pull-up resistor. In addition to functioning as a general-purpose I/O port, Port 70 also functions as an input clock pin TI0; Port 71 as an 8-bit timer output (TO1), Port 72 as a PWM0 output (TO2), and Port 73 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the Port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7FC value to 0, and sets all bits to ports.

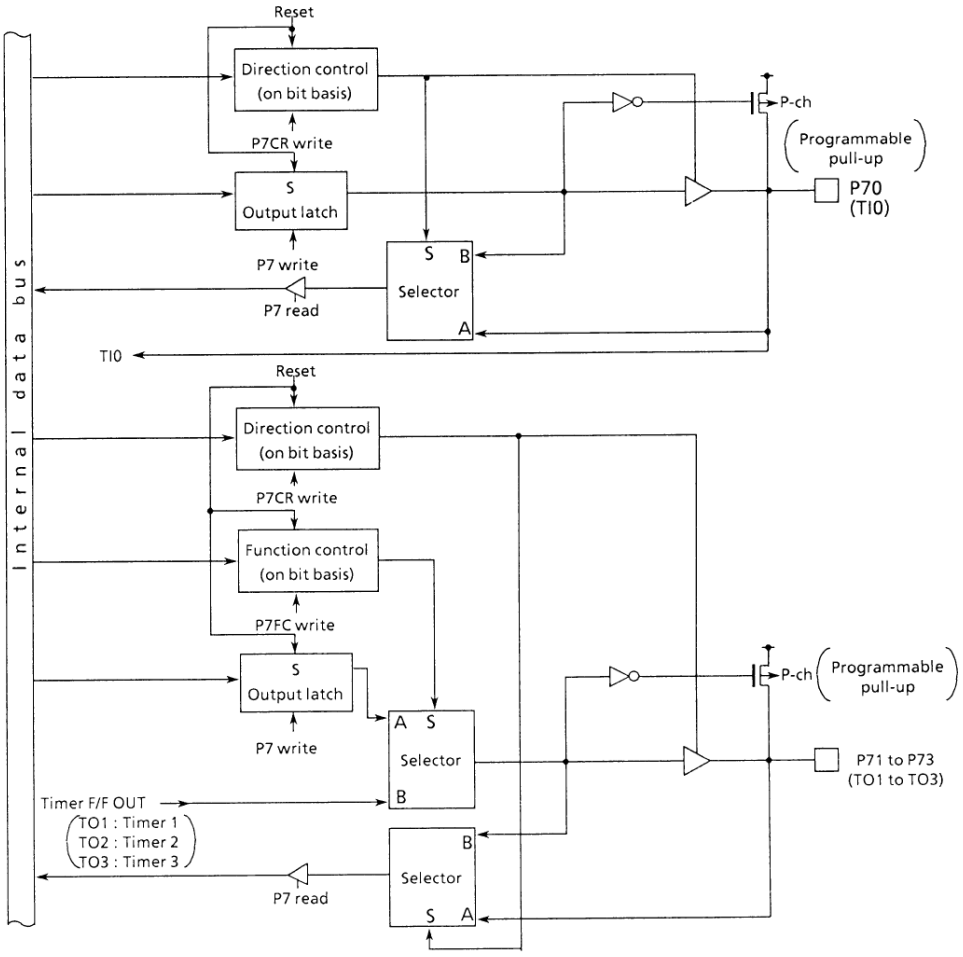
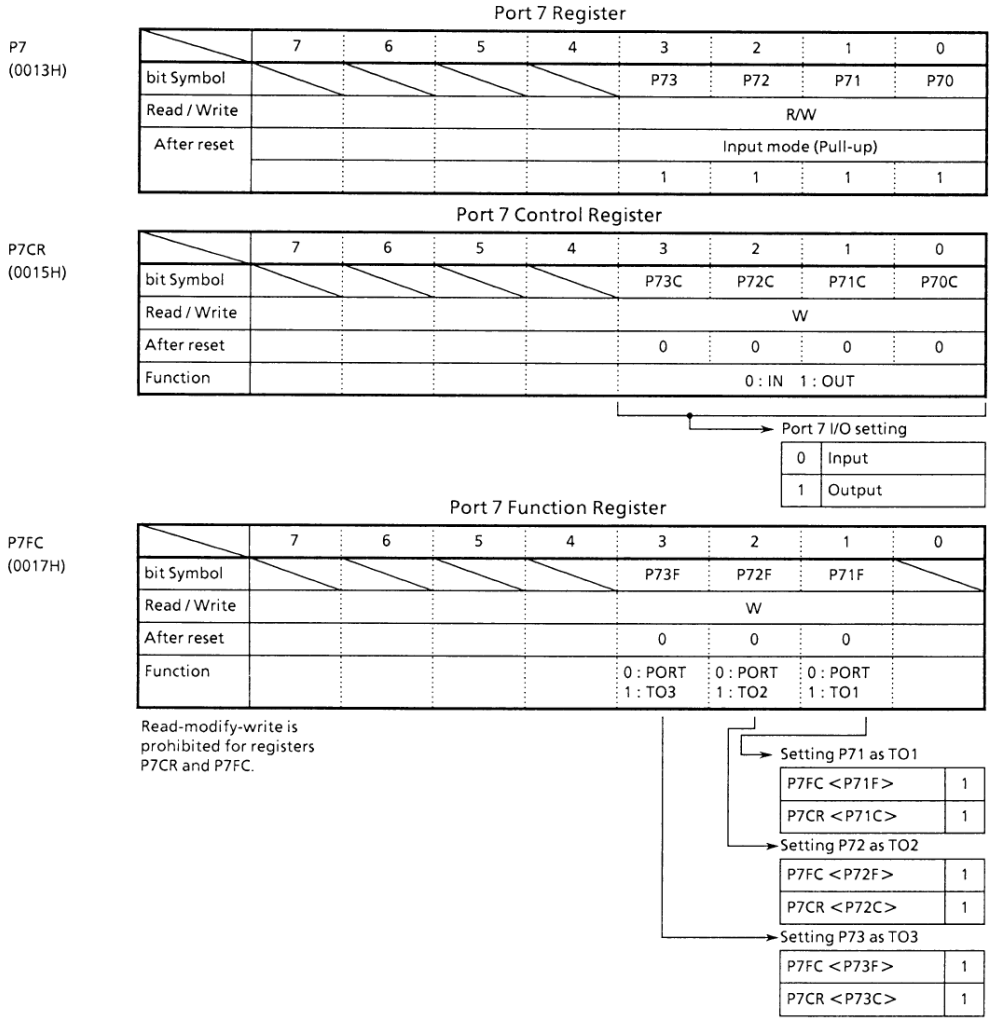


Figure 3.5 (15). Port 7



Note) P70 / TI0 pin does not have a register changing PORT / FUNCTION.
For example, when it is used as an input port (P70), the input signal for P70 is inputted to 8 bit Timer 0 as a timer input 0 (TI0).

Figure 3.5 (16). Registers for Port 7

3.5.9 Port 8 (P80 to P83)

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 and connects a pull-up resistor. 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 4 and 5 clocks, an output for 16-bit timer F/F 4, 5 and 6 output, and an input for INT0. Writing "1" in the corresponding bit of the Port 8 function register (P8FC) enables those functions. Resetting resets the function register P8FC value to "0", and sets all bits to ports.

(1) P80 to P86

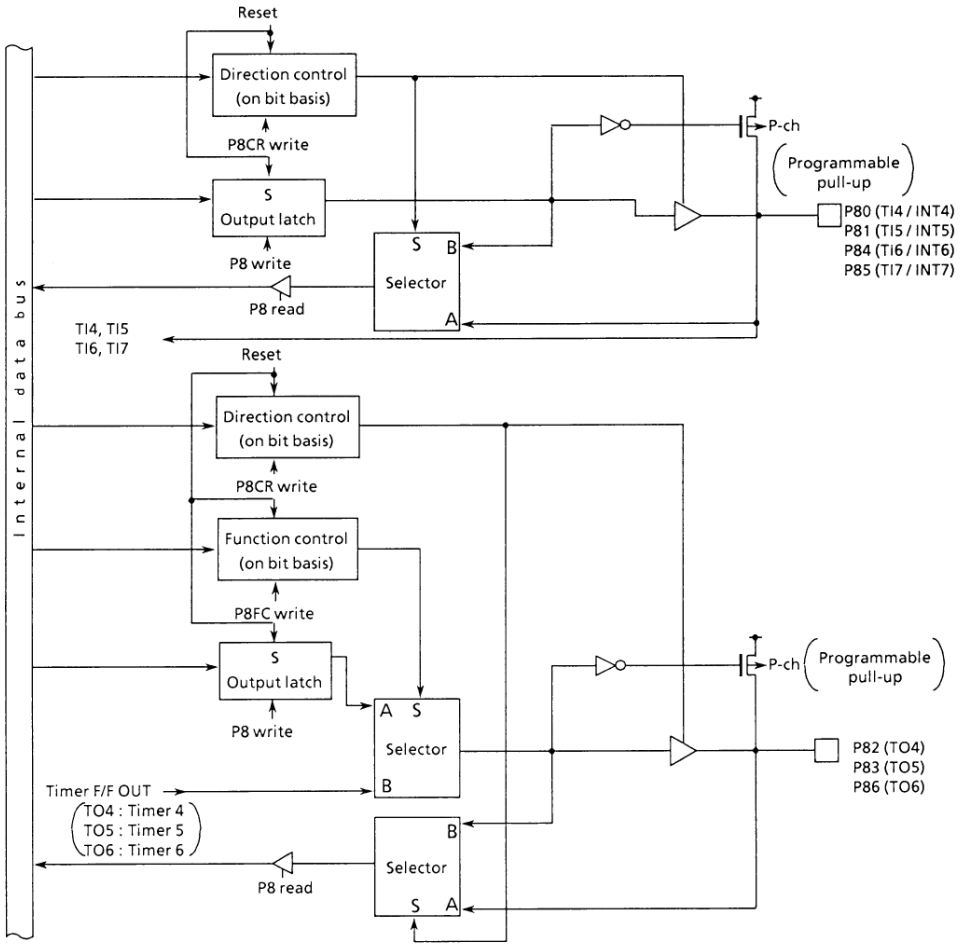


Figure 3.5 (17). Port 8 (P80 to P86)

(2) P87 (INT0)

Port 87 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

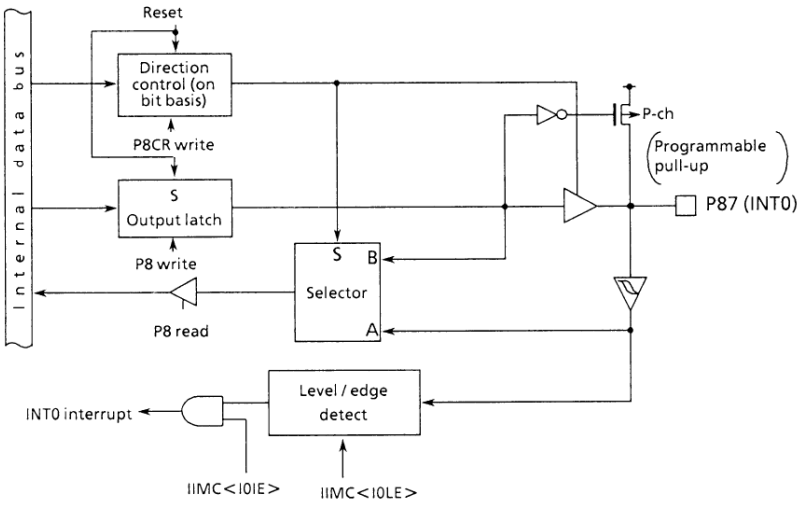
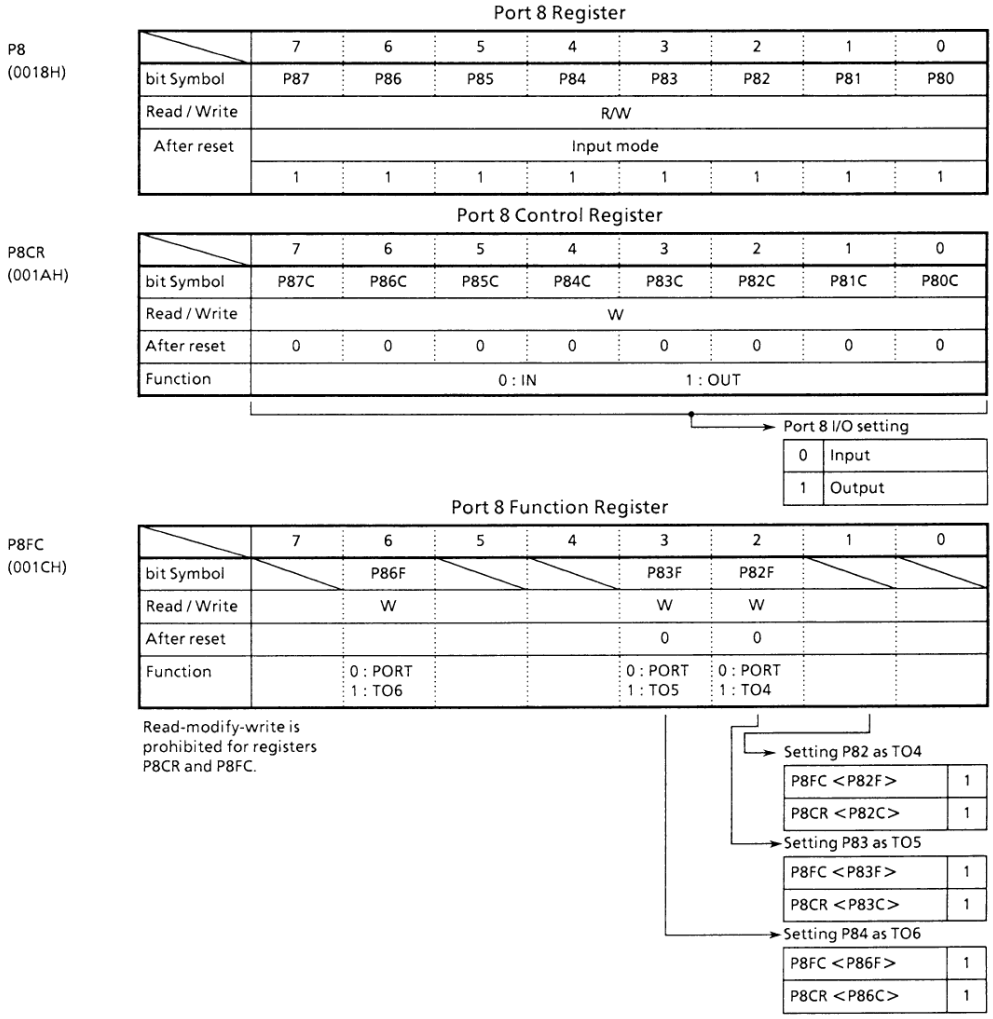


Figure 3.5 (18). Port 87



Note) P80 / TI4, P81 / TI5, P84 / TI6, P85 / TI7 pins do not have a register changing PORT / FUNCTION. Therefore this is the same as P70/TI0 pin.
When P87 / INT0 pin is used as an INT0 pin, set P8CR<P87C> to "0" and IIMC<I0IE> to "1".

Figure 3.5 (19). Registers for Port 8

3.5.10 Port 9 (P90 to P95)

• Port 90 to 95

Ports 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 95 to an input port and connects a pull-up resistor.

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 95 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 this function.

Resetting resets the function register value to "0" and sets all bits to ports.

• Port 96 to 97

Ports 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 97 to an open drain type buffer

Resetting sets P96 to 97 to an output port and outputs high-impedance (HZ) because output latch and control register are set to "1".

In addition to functioning as a general-purpose I/O port, P96 to 97 can also function as a low frequency oscillator pin fro dual clock mode. The dual clock function can be set by programming system clock control register SYSCR0, 1.

(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.

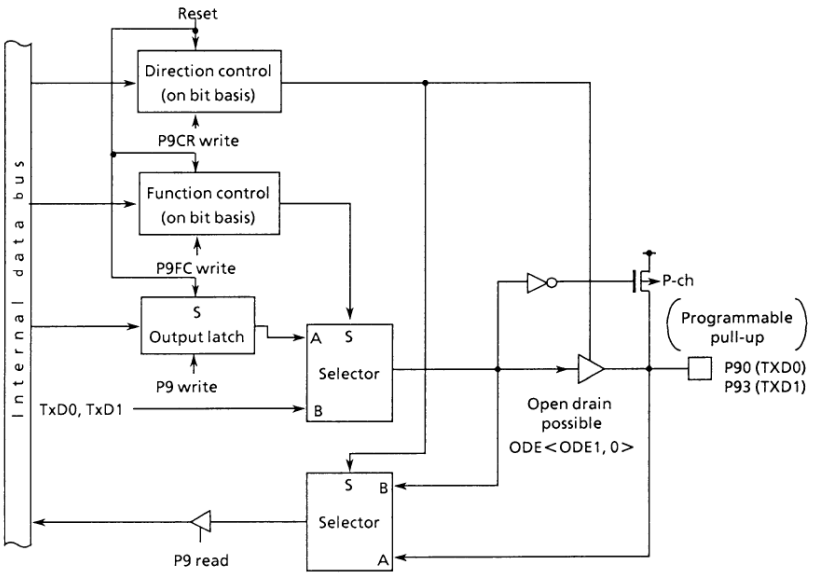


Figure 3.5 (20). Ports 90 and 93

(2) Ports 91, 94 (RXD0, 1) input pins for serial channels.

Ports 91 and 94 are I/O ports, and also used as RXD

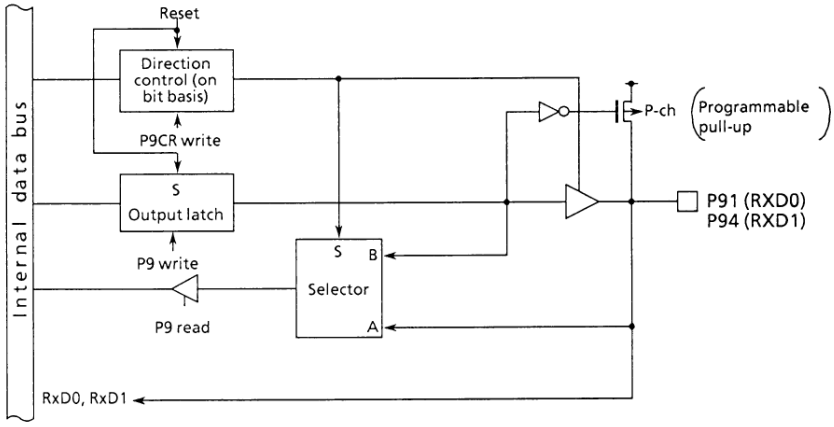


Figure 3.5 (21). Ports 91 and 94

(3) Port 92 ($\overline{CTS}$) pin and as a SCLK0 I/O pin for serial channels.

Port 92 is an I/O port, and also used as a $\overline{CTS}$ input

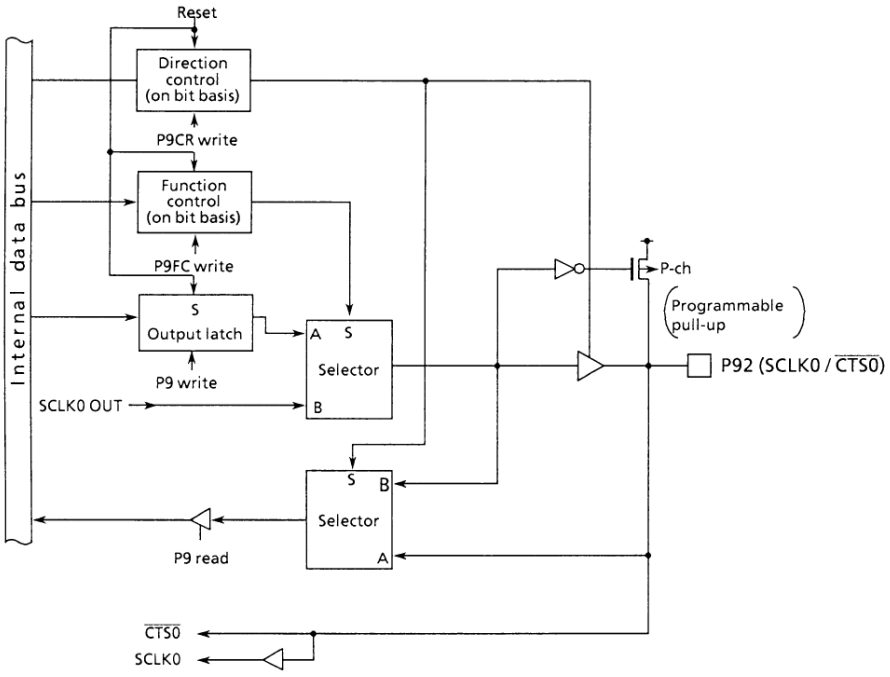


Figure 3.5 (22). Port 92

(4) Port 95 (SCLK) an SCLK1 I/O pin for serial channel 1.

Port 95 is a general-purpose I/O port. It is also used as

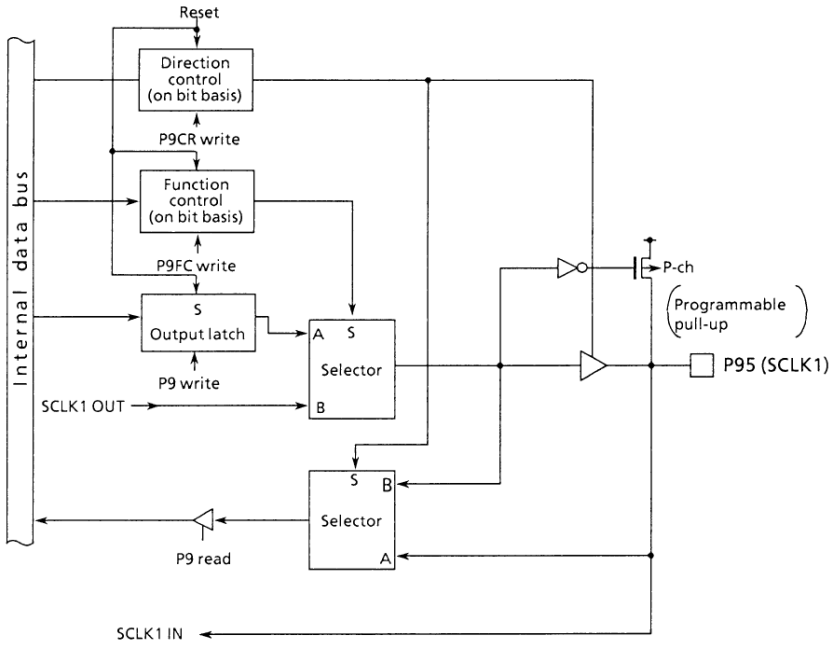


Figure 3.5 (23). Port 95

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

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

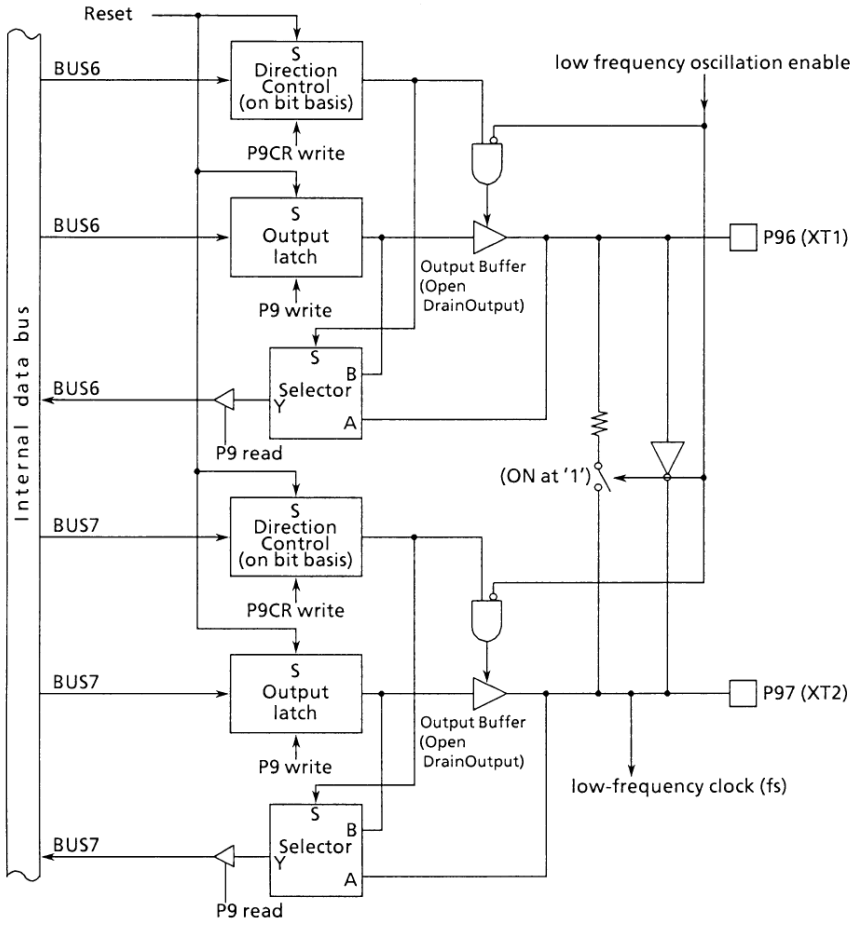


Figure 3.5 (24). Port 96 to 97

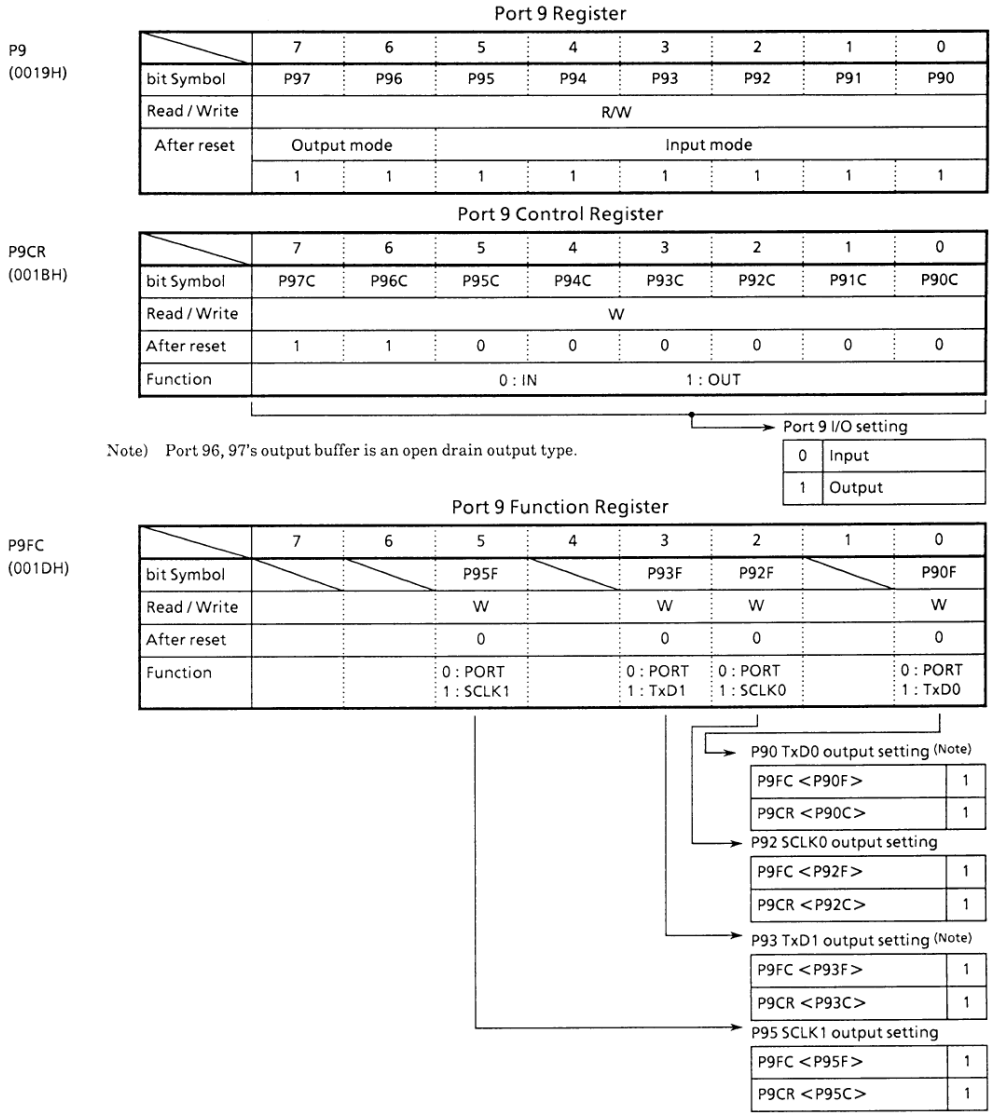


Figure 3.5 (25). Register for Port 9

3.5.11 Port A (PA0 to PA7)

Port A is an 8-bit general-purpose I/O port. I/O can be set on a bit basis by control register PACR.

Resetting sets Port A as an input port by resetting PACR. It also sets all bits of the output latch register to "1".

In addition to functioning as a general-purpose I/O port (only PA7), PA7 can also function as a clock output pin.

The clock output is f_{FPH} or f_{SYS} that is selected oscillator output clock. It is selected by CKOCR <SCOSEL>.

SCOUT function is enabled by setting PACR <PA7C> and CKOCR <SCOEN>.

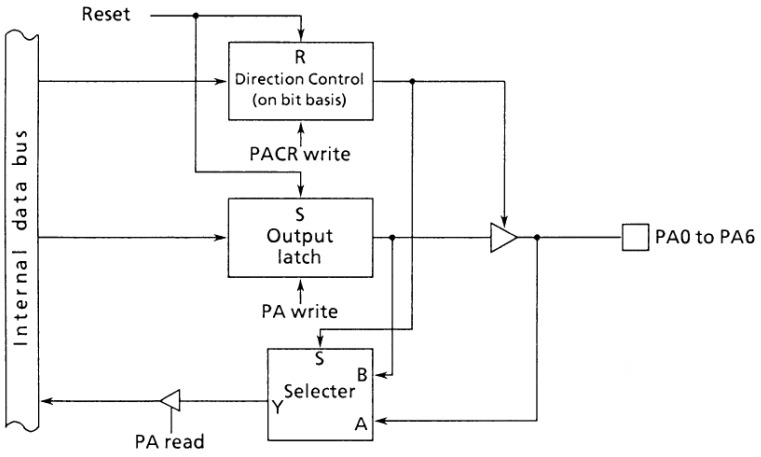


Figure 3.5 (26). Port A0 to A6

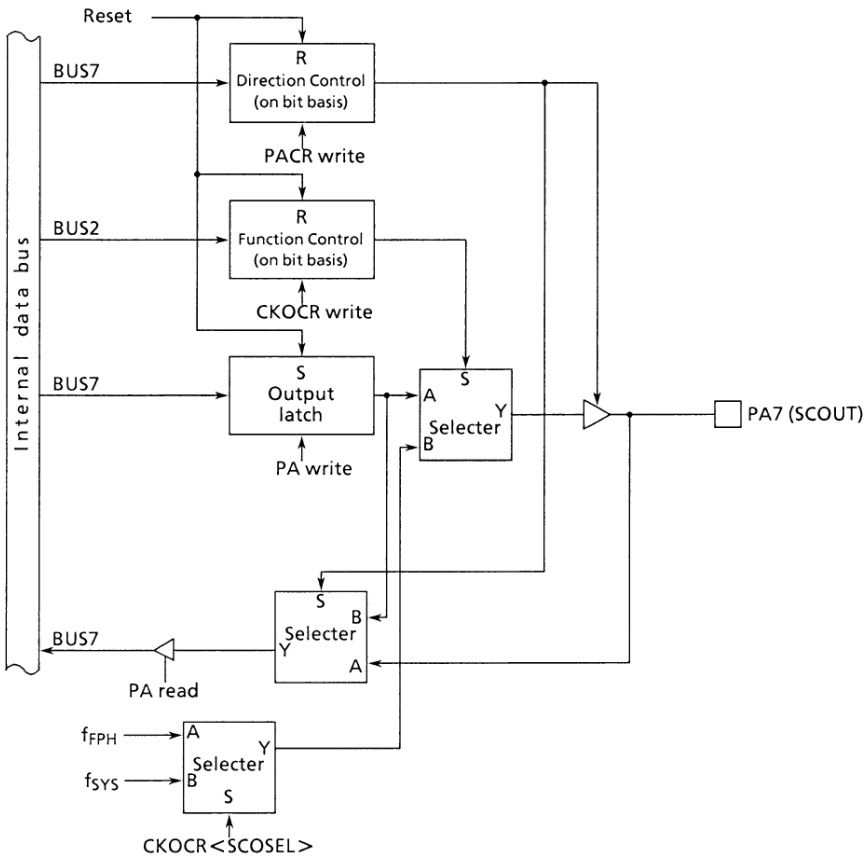
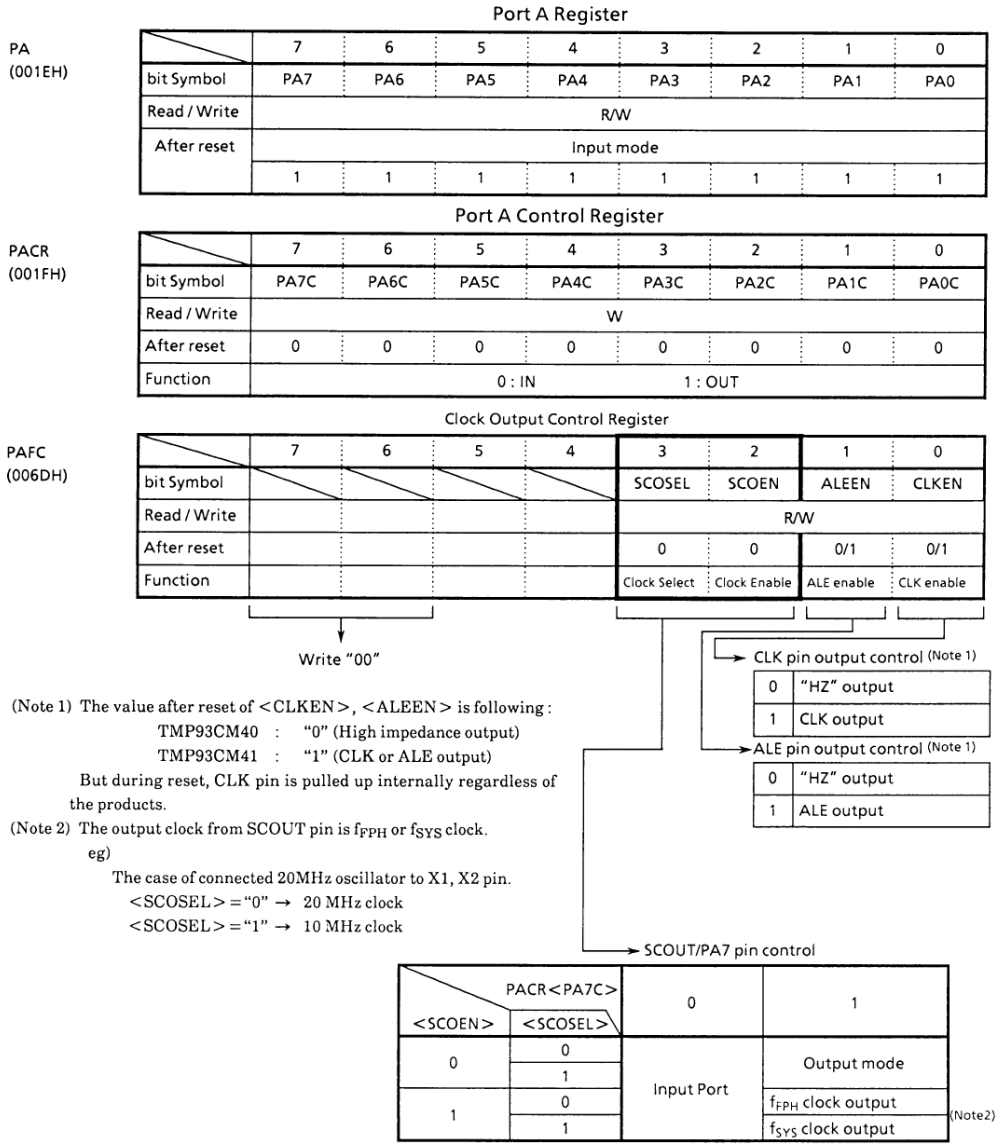


Figure 3.5 (27). Port A7



(Note 1) The value after reset of <CLKEN>, <ALEEN> is following :
TMP93CM40 : "0" (High impedance output)
TMP93CM41 : "1" (CLK or ALE output)
But during reset, CLK pin is pulled up internally regardless of the products.

(Note 2) The output clock from SCOUT pin is f_{FPH} or f_{SYS} clock.
eg)
The case of connected 20MHz oscillator to X1, X2 pin.
<SCOSEL> = "0" → 20 MHz clock
<SCOSEL> = "1" → 10 MHz clock

Figure 3.5 (28). Registers for Port A

3.6 Chip Select/Wait Control, AM8/$\overline{16}$ pin

TMP93CM40/M41 has a built-in chip select/wait controller used to control chip select ($\overline{CS0}$ - $\overline{CS2}$ pins), wait ($\overline{WAIT}$ pin), and data bus size (8 or 16 bits) for any of the three block address areas, and AM8/$\overline{16}$ pin selects external data bus width for TMP93CM41.

3.6.1 AM8/$\overline{16}$ pin

(1) TMP93CM40

Set this pin to "1". After reset, the CPU accesses the internal ROM with 16 bit bus width. When the CPU access an external bus area, the bus width is set by Chip Select/Wait Control Register (described at 3.6.2), P1CR and P1FC (The value "1" of this pin is ignored and the value set by register is active).

(2) TMP93CM41

(2-1) 16 bit bus width interlarded with 8 bit bus width or fixed 16 bit bus width

Set this pin to "0". Port 1/AD8 to 15/A8 to 15 pins are then fixed to A8 to 15 function compulsorily and the value of P1CR, P1FC are ignored.

When the CPU accesses an external bus area, the bus width is set by Chip Select/Wait Control Register as described in section 3.6.2.

However, the bus width program memory only after reset must be 16 bit bus width.

(2-2) fixed bit bus width

Set this pin to "1". Port 1/AD8 to 15/A8 to 15 pins are then fixed to A8 to 15 function compulsorily and the value of P1CR, P1FC are ignored. The value of bit 4: <B0BUS>, <B1BUS>, or <B2BUS> described at 3.6.2 are ignored and the bus width is fixed to 8 bit.

3.6.2 Control Registers

Table 3.6 (1) shows control registers

One block address areas are controlled by 1-byte CS/WAIT control registers (B0CS, B1CS, and B2CS). Registers can be written to only when the CPU is in system mode. (There are two CPU modes: system and normal.) The reason is that the settings of these registers have an important effect on the system.

(1) Enable

Control register bit 7 (B0E, B1E, and B2E) is a master bit used to specify enable (1)/disable (0) of the setting.

Resetting B0E and B1E to disable (0) and B2E to enable (1).

(2) CS/CAS Waveform select

Control register bit 5 (B0CAS, B1CAS, and B2CAS) is used to specify waveform mode output from the chip select pin ($\overline{CS0}$/$\overline{CAS0}$ - $\overline{CS2}$/$\overline{CAS2}$). Setting this bit to 0 specifies $\overline{CS0}$ to $\overline{CS2}$ waveforms; setting it to 1 specifies $\overline{CAS0}$ to $\overline{CAS2}$ waveforms.

Resetting clears bit 5 to 0.

(3) Data bus size select

Bit 4 (B0BUS, B1BUS, and B2BUS) of the control register is used to specify data bus size. Setting this bit to 0 accesses the memory in 16-bit data bus mode; setting it to 1 accesses the memory in 8-bit data bus mode.

Changing data bus size depending on the access address is called dynamic bus sizing. Table 3.6 (2) shows the details of the bus operation.

(4) Wait control

Control register bits 3 and 2 (B0W1, 0; B1W1, 0; B2W1, 0) are used to specify the number of waits. Setting these bits to 00 inserts a 2-state wait regardless of the $\overline{WAIT}$ pin status. Setting them to 01 inserts a 1-state wait regardless of the $\overline{WAIT}$ status. Setting them to 10 inserts a 1-state wait and samples the $\overline{WAIT}$ pin status. If the pin is low, inserting the wait maintains the bus cycle until the pin goes high. Setting them to 11 completes the bus cycle without a wait regardless of the $\overline{WAIT}$ pin status.

Resetting sets these bits to 00 (2-state wait mode).

(5) Address area specification

Control register bits 1 and 0 (B0C1, 0; B1C1, 0; B2C1, 0) are used to specify the target address area. Setting these bits to 00 enables settings ($\overline{CS}$ output, Wait state, Bus size, etc.) as follows:

* CS0 setting enabled when 7F00H to 7FFFH is accessed.
* CS1 setting enabled when 880H to 7FFFH is accessed.
* CS2 setting enabled when 8000H to 3FFFFFH is accessed, for the TMP93CM41, which does not have a built-in ROM.

CS2 setting enabled when 10000H to 3FFFFFH is accessed for the TMP93CM40, which has built-in ROM.

Setting bits to 01 enables setting for all CS's blocks and outputs a low strobe signal ($\overline{CS0}$/$\overline{CAS0}$ ~ $\overline{CS2}$/$\overline{CAS2}$) from chip select pins when 400000H to 7FFFFFH is accessed. Setting bits to 10 enables them 800000H to BFFFFFH is accessed. Setting bits to 11 enables them when C00000H to FFFFFFH is accessed.

Table 3.6 (1) Chip Select/Wait Control Register

Code	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block0 CS/WAIT control register	0068H	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1 : Master bit of bit 0 to 6		0 : $\overline{CS0}$ 1 : $\overline{CAS0}$	0 : 16 bit Bus 1 : 8 bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT		00 : 7F00H to 7FFFH 01 : 400000H to 10 : 800000H to 11 : C00000H to	
B1CS	Block1 CS/WAIT control register	0069H	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1 : Master bit of bit 0 to 6		0 : $\overline{CS1}$ 1 : $\overline{CAS1}$	0 : 16 bit Bus 1 : 8 bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT		00 : 880H to 7FFFH 01 : 400000H to 10 : 800000H to 11 : C00000H to	
B2CS	Block2 CS/WAIT control register	006AH	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1 : Master bit of bit 0 to 6		0 : $\overline{CS2}$ 1 : $\overline{CAS2}$	0 : 16 bit Bus 1 : 8 bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT +n 11 : 0WAIT		00 : 8000H to 01 : 400000H to 10 : 800000H to 11 : C00000H to	

Note: Only block 2 is enable (16-bit data bus, 2-wait mode) after reset.

Table 3.6 (2) Dynamic Bus Sizing

Operand Data Size	Operand Start Address	Memory Data Size	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 2n + 1	xxxxx xxxxx	b7 to b0 b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1 2n + 2	xxxxx xxxxx	b7 to b0 b15 to b8
		16 bits	2n + 1 2n + 2	b7 to b0 xxxxx	xxxxx b15 to b8
32 bits	2n + 0 (even number)	8 bits	2n + 0 2n + 1 2n + 2 2n + 3	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16 bits	2n + 0 2n + 2	b15 to b8 b31 to b24	b7 to b0 b23 to b16
	2n + 1 (odd number)	8 bits	2n + 1 2n + 2 2n + 3 2n + 4	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16 bits	2n + 1 2n + 2 2n + 4	b7 to b0 b23 to b16 xxxxx	xxxxx b15 to b8 b31 to b24

xxxxx: During a read, data input to the bus is ignored. At write, the bus is at high impedance and the write strobe signal remains non-active.

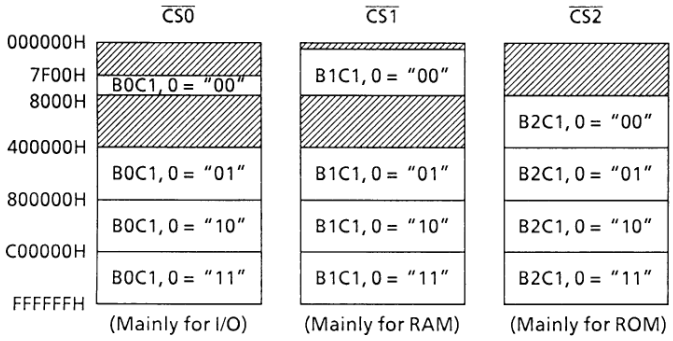
3.6.3 Chip Select Image

An image of the actual chip select is shown below. Out of the whole memory area, address areas that can be specified are divided into four parts. Addresses from 000000H to 3FFFFFH are divided differently: 7F00H to 7FFFH is specified for CS0; 880H to 7FFFH, for CS1; and 8000H to 3FFFFFH, for CS2. The reason is that a device other than ROM (i.e., RAM or I/O) might be connected externally.

7F00 to 7FFFH (256 bytes) for CS0 are mapped mainly for possible expansions to external I/O.

880H to 7FFFH (approximately 31K bytes) for CS1 are mapped there mainly for possible extensions to external RAM.

8000H to 3FFFFFH (approximately 4M bytes) for CS2 are mapped mainly for possible extensions to external ROM. After reset, CS2 is enabled in 16-bit bus and 2-wait. With the TMP93CM41, which does not have a built-in ROM, the program is externally read at address 8000H in this setting (16-bit bus, 2-wait). With the TMP93CM40, which has a built-in ROM, addresses from 8000H to FFFFFH are used as the internal ROM area; CS2 is disabled in this area. After reset, the CPU reads the program from the built-in ROM in 16-bit bus, 0-wait mode.



Note 1 : Access priority is highest for built-in I/O, then built-in memory, and lowest for the chip select / wait controller.

Note 2 : External areas other than $\overline{CS0}$ to $\overline{CS2}$ are accessed in 16-bit data bus (0 wait) mode.
When using the chip select/wait controller, do not specify the same address area more than once. (However, when addresses 7F00H to 7FFFH for CS0 and 880H to 7FFFH for CS1 are specified, in other words, specifications overlap, only the CS0 setting / pin is active.)

Note 3 : When the bus is released ($\overline{BUSAK}$ = "0"), $\overline{CS0}$ to $\overline{CS2}$ pins are also released (the output buffer is OFF). Refer to 「Note about the bus release」 in 3.5 Functions of Ports about the state of pins.

3.6.4 Example of Usage

(1) Example of Usage - 1

Figure 3.6 (1) is an example in which an external memory is connected to the TMP93CM41. In this example, a ROM is connected using 16-bit Bus; a RAM is connected using 8-bit Bus.

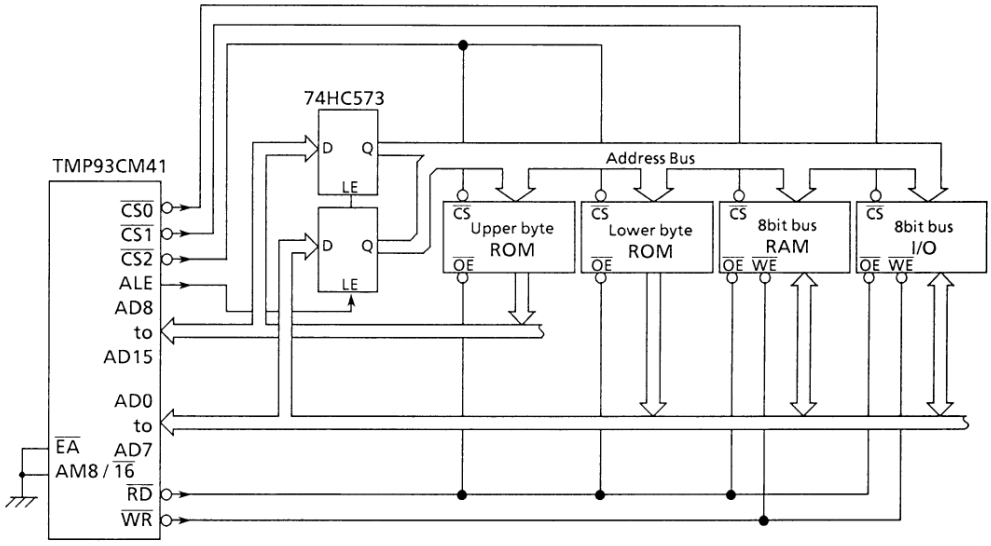


Figure 3.6 (1). Example of External Memory Connection (ROM = 16 bits, RAM and I/O = 8 bits)

Resetting sets pins $\overline{CS0}$ to $\overline{CS2}$ to input port mode. $\overline{CS0}$ and $\overline{CS1}$ are set high due to an internal pull-up resistor; $\overline{CS2}$, low due to an internal pull-down resistor. The program used to set these pins is as follows:

```
P4CR   EQU    0EH
P4FC   EQU    10H
B0CS   EQU    68H
B1CS   EQU    69H
B2CS   EQU    6AH
LD     (B0CS), 1X010000B    ; CS0 = 8 bit, 2WAIT, 7F00H to 7FFFH
LD     (B1CS), 1X011100B    ; CS1 = 8 bit, 0WAIT, 880H to 7EFFH
LD     (B2CS), 1X000100B    ; CS2 = 16 bit, 1WAIT, 8000H to 3FFFFFH
LD     (P4CR), XXXXX111B    } CS0, CS1, CS2 output mode setting
LD     (P4FC), XXXXX111B    }
```

(Note) X: don't care

(2) Example of Usage - 2

Figure 3.6 (2) is an example in which an external memory is connected to the TMP93CM41. In this example, a ROM, a RAM, and I/O are connected using 8-bit bus.

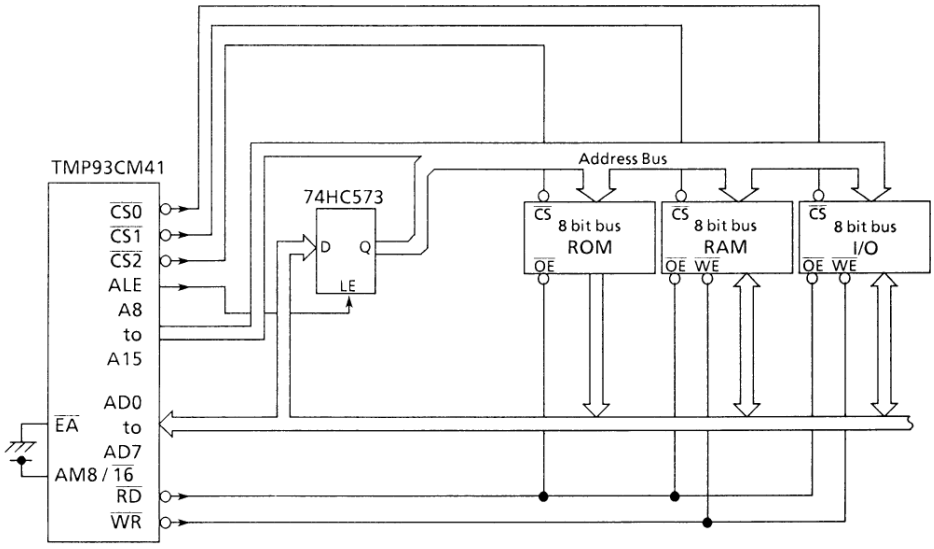


Figure 3.6 (2). Example of External Memory Connection (ROM and RAM and I/O = 8 bits)

Resetting sets pins $\overline{CS0}$ to $\overline{CS2}$ to input port mode. $\overline{CS0}$ and $\overline{CS1}$ are set high due to an internal pull-up resistor; $\overline{CS2}$, low due to an internal pull-down resistor. The program used to set these pins is as follows:

```
P4CR   EQU    0EH
P4FC   EQU    10H
B0CS   EQU    68H
B1CS   EQU    69H
B2CS   EQU    6AH
LD     (B0CS), 1X0X0000B    ; CS0 = 8 bit, 2WAIT, 7F00H to 7FFFH
LD     (B1CS), 1X0X1100B    ; CS1 = 8 bit, 0WAIT, 880H to 7EFFH
LD     (B2CS), 1X0X0100B    ; CS2 = 8 bit, 1WAIT, 8000H to 3FFFFFH
LD     (P4CR), XXXXX111B    } CS0, CS1, CS2 output mode setting
LD     (P4FC), XXXXX111B    }
```

(Note) X : don't care

3.7 8-bit Timers

TMP93CM40/M41 contains two 8-bit timers (timers 0 and 1), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

- 8-bit interval timer mode (2 timers)
- 16-bit interval timer mode (1 timer)
- 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) shows the block diagram of 8-bit timer (timer 0 and timer 1).

Each interval timer consists of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flip-flop (TFF1) is provided for pair of timer 0 and timer 1.

Among the input clock sources for the interval timers, the internal clocks of ϕT1, ϕT4, ϕT16, and ϕT256 are obtained from the 9-bit prescaler shown in Figure 3.7 (2).

The operation modes and timer flip-flops of the 8-bit timer are controlled by three control registers TMOD, TFFCR, and TRUN.

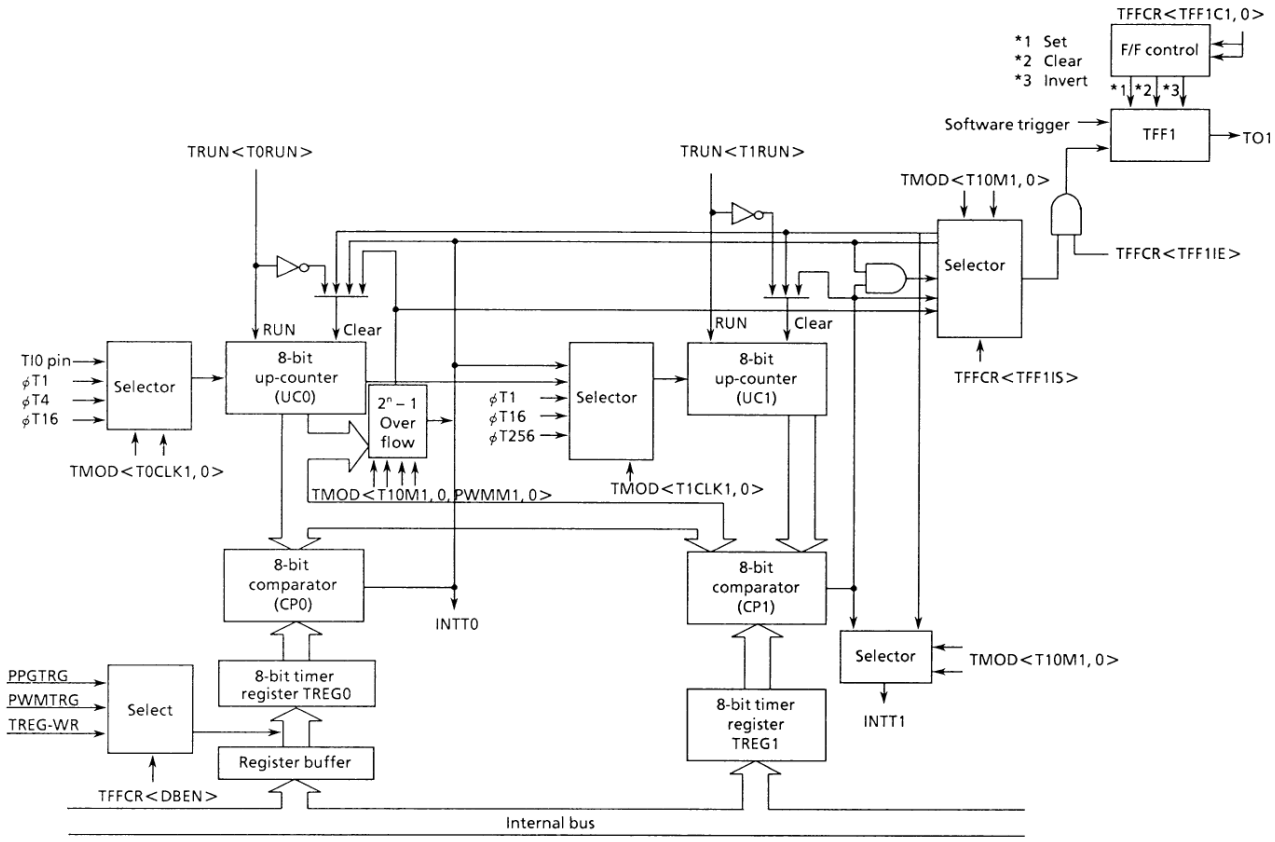


Figure 3.7 (1). Block Diagram of 8-Bit Timers (Timers 0 and 1)

① Prescaler

There are 9 bit prescaler and prescaler clock selection register to generate input clock for 8 bit Timer 0, 1, 16 bit Timer 4, 5 and Serial Interface.

Figure 3.7 (2) shows the block digram. Table 3.7 (1) shows prescaler clock resolution to 8, 16 bit Timer.

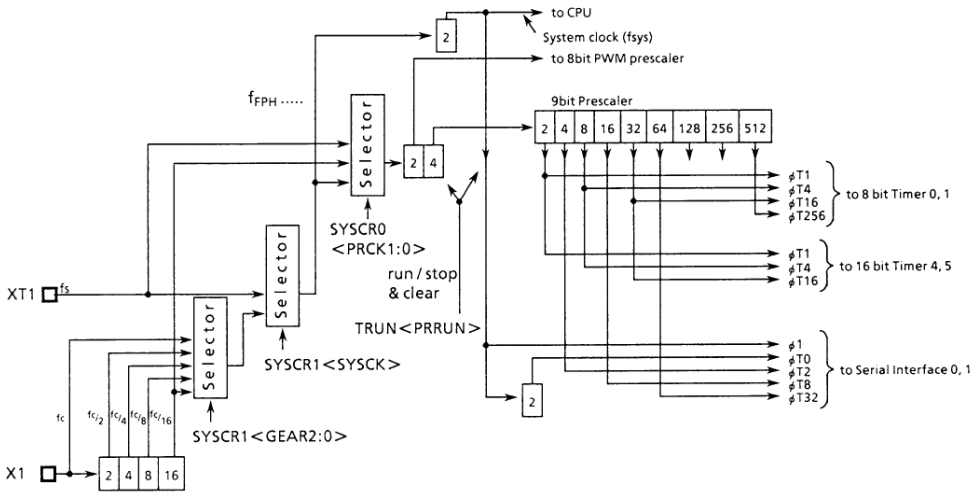


Figure 3.7 (2). Prescaler Block Diagram

Table 3.7 (1) Prescaler Clock Resolution to 8, 16 bit Timer

at fc = 16 MHz, fs = 32 kHz

Select system clock <SYSCK>	Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 : 0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^3$ (250 μs)	$fs/2^5$ (1 ms)	$fs/2^7$ (4 ms)	$fs/2^{11}$ (64 ms)
0 (fc)		000 (fc)	$fc/2^3$ (0.5 μs)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)	$fc/2^{11}$ (128 μs)
		001 ($fc/2$)	$fc/2^4$ (1 μs)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{12}$ (256 μs)
		010 ($fc/4$)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{13}$ (512 μs)
		011 ($fc/8$)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{10}$ (64 μs)	$fc/2^{14}$(1.024 ms)
		100 ($fc/16$)	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{11}$ (128 μs)	$fc/2^{15}$(2.048 ms)
XXX	01 (low frequency clock)	XXX	$fs/2^3$ (250 μs)	$fs/2^5$ (1 ms)	$fs/2^7$ (4 ms)	$fs/2^{11}$ (64 ms)
XXX	10 (note) ($fc/16$ clock)	XXX	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{11}$ (128 μs)	$fc/2^{15}$(2.048 ms)
XXX : don't care			← 16 bit Timer →			
			← 8 bit Timer →			

(Note) The $fc/16$ clock as a prescaler clock can not be used when the fs is used as a system clock.

The 1/4 times clock selected among f_{FPH} clock is input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1 : 0>.

Resetting sets <PRCK1 : 0> to "00", therefore, f_{FPH} /4 clock is input.

The 8 bit Timer 0, 1 uses 4 types of clock: øT1, øT4, øT16, and øT256 among the prescaler output.

The prescaler can be run or stopped by the timer control register TRUN <PRRUN>. Counting starts when <PRRUN> is set to "1", while the prescaler is cleared to zero and stops operation when <PRRUN> is set to "0".

When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop this prescaler before "HALT" instruction is executed.

② Up-counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by TMOD.

The input clock of timer 0 is selected from the external clock from T10 pin and the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of TMOD register.

The input clock of timer 1 differs depending on the operation mode. When set to 16-bit timer mode, the overflow output of timer 0 is 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 timer 0 according to the set value of TMOD register.

Example : When TMOD <T10M1,0> = 01, the overflow output of timer 0 becomes the input clock of timer 1 (16 bit timer mode).

When TMOD <T10M1,0> = 00 and TMOD <T1CLK1, 0> = 01, ϕT1 becomes the input of timer 1 (8 bit timer mode).

Operation mode is also set by TMOD register. When reset, it is initialized to TMOD <T01M1, 0> = 00 whereby the up-counter is placed in the 8-bit timer mode.

The counting and stop and clear of up-counter can be controlled for each interval timer by the timer operation control register TRUN. When reset, all up-counters will be cleared to stop the timers.

③ Timer register

This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, 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.

Timer register TREG0 is of double buffer structure, each of which makes a pair with register buffer.

The timer flip-flop control register TFFCR <DBEN> bit controls whether the double buffer structure in the TREG0 should be enabled or disabled. It is disabled when <DBEN> = 0 and enabled when they are set to 1.

In the condition of double buffer enable state, the data is transferred from the register buffer to the timer register when the 2^n - 1 overflow occurs in PWM mode, or at the PPG cycle in PPG mode. Therefore, during timer mode, the double buffer cannot be used.

When reset, it will be initialized to <DBEN> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <DBEN> to 1, and write the following data in the register buffer.

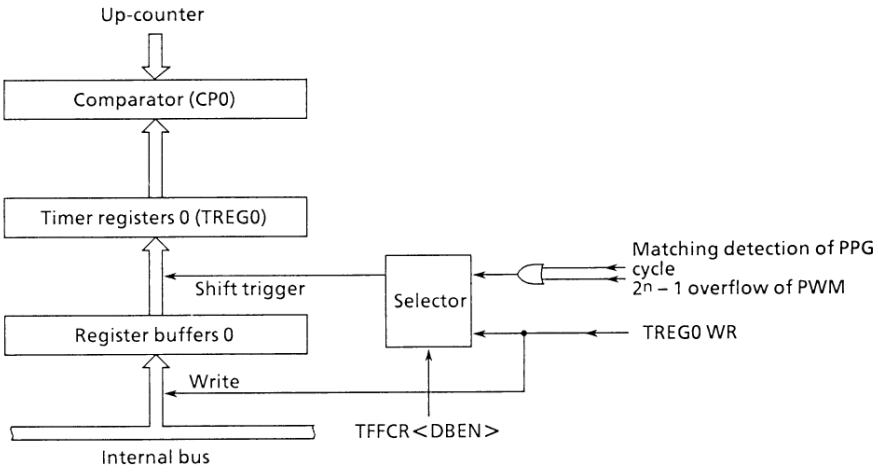


Figure 3.7 (3). Configuration of Timer Register 0

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

The memory address of each timer register is as follows.

TREG0: 000022H
TREG1: 000023H

All registers are write-only and cannot be read.

④ Comparator

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 (INTT0, INTT1) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

⑤ Timer flip-flop (timer F/F: TFF1)

The status of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer and the value can be output to the timer output pins TO1 (also used as P71).

A timer F/F is provided for a pair of timer 0 and timer 1 and is called TFF1. TFF1 is output to TO1 pin.

TRUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler & Timer Run / Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							

→ Count Operation

0	Stop and clear
1	Count

PRRUN : Operation of prescaler
T5RUN : Operation of 16-bit timer (timer5)
T4RUN : Operation of 16-bit timer (timer4)
P1RUN : Operation of PWM timer (PWM1 / timer3)
P0RUN : Operation of PWM timer (PWM0 / timer2)
T1RUN : Operation of 8-bit timer (timer1)
T0RUN : Operation of 8-bit timer (timer0)

SYSCR1 (006EH)

	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	0	0	0	0	0	0
Function	High Frequency oscillator (fc) 0 : stop 1 : oscillaton	Low Frequency oscillator (fs) 0 : stop 1 : oscillaton	High Frequency oscillator (fc) after released STOP mode 0 : stop 1 : oscillaton	Low Frequency oscillator (fs) after released STOP mode 0 : stop 1 : oscillaton	select clock after released STOP mode 0 : fc 1 : fs	Warming Up Timer 0 write : don't care 1 write : start timer 0 read : end warming up 1 read : not end warming up	select prescaler clock 00 : f_{FPH} 01 : fs 10 : fc/16 11 : (reserved)	

→ select gear value of high frequency

00	f_{FPH}
01	fs
10	fc / 16
11	(reserved)

} 1 / 4times clock

Figure 3.7 (4). Timer Operation Control Register/System Clock Control Register

TMOD (0024H)

	7	6	5	4	3	2	1	0
bit Symbol	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	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: – 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock of timer1 00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of timer0 00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Prohibit Read Modify Write

Input clock of timer 0

00	External Input (TI0)
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Input clock of timer 1

	TMOD<T10M1,0> ≠ 01	TMOD <T10M1,0> = 01
00	Comparator output of timer 0	Overflow output of timer 0
01	Internal clock ϕT1	
10	Internal clock ϕT16	(16-bit timer mode)
11	Internal clock ϕT256	

Select PWM cycle

00	——
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Set the operation mode of timer 0 and 1.

00	Two 8-bit timers (timer 0 and timer 1)
01	16-bit timer
10	8-bit PPG output
11	8-bit PWM output (timer 0) + 8-bit timer (timer 1)

Figure 3.7 (5). Timer Mode Control Register (TMOD)

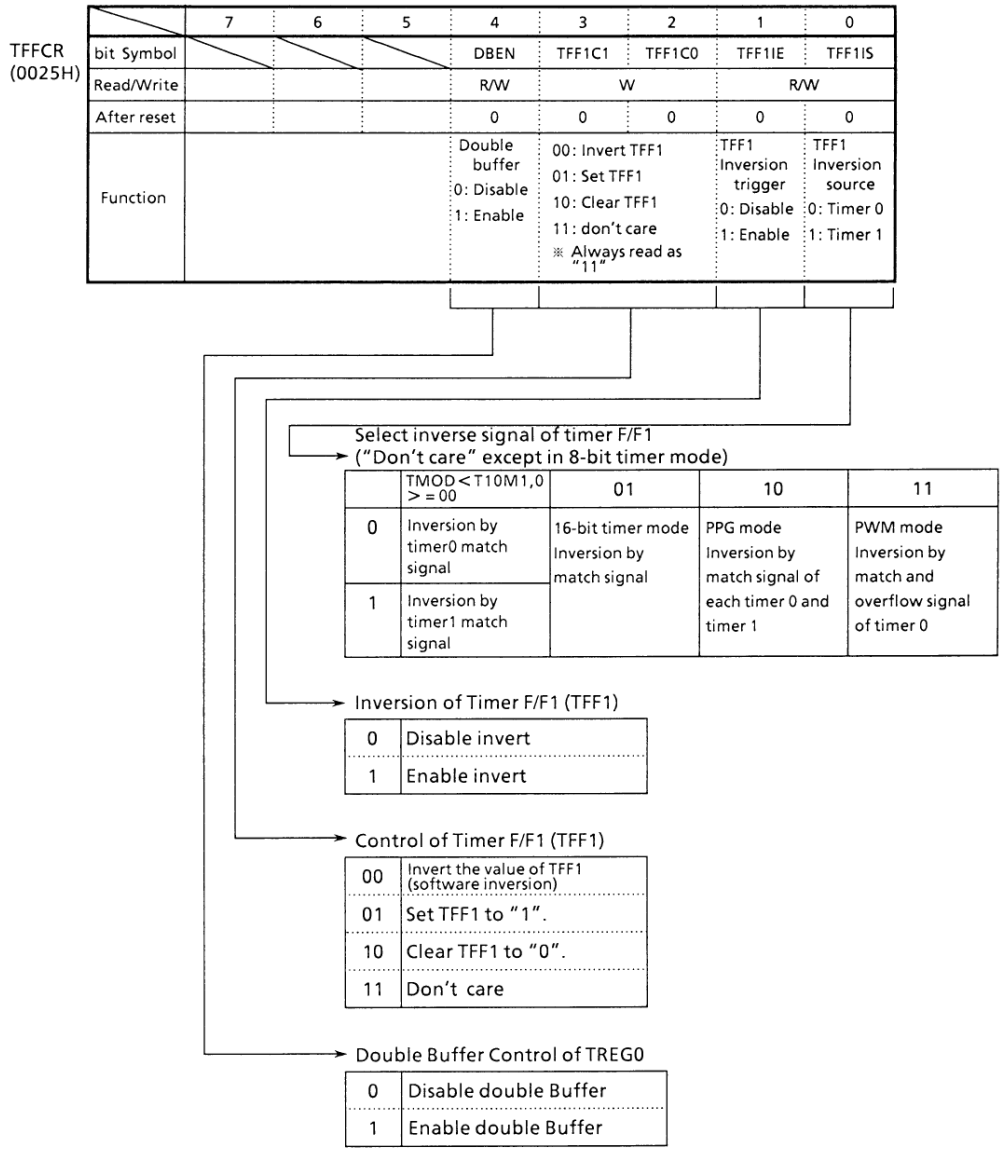


Figure 3.7 (6). Timer Flip-Flop Control Register (TFFCR)

The operation of 8-bit timers will be described below:

(1) 8-bit timer mode

Two interval timers 0, 1, can be used independently as 8-bit interval timer. All interval timers operate in the same manner, and thus only the operation of timer 1 will be explained below.

① Generating interrupts in a fixed cycle

To generate timer 1 interrupt at constant intervals using timer 1 (INTT1), first stop timer 1 then set the operation mode, input clock, and a cycle to TMOD and TREG1 register, respectively. Then, enable interrupt INTT1 and start the counting of timer 1.

Example: To generate timer 1 interrupt every 40 microseconds at fc = 16 MHz, set each register in the following manner.

※ Clock Condition

- system clock : low frequency (fs)
- clock gear : xxx
- prescaler clock : low frequency (fs)

```
                MSB          LSB
                  7 6 5 4 3 2 1 0
TRUN      ← - X - - - - 0 -      Stop timer 1, and clear it to "0".
TMOD      ← 0 0 X X 0 1 - -      Set the 8-bit timer mode, and select φT16 (4 ms at fs = 32 kHz)
                                 as the input clock.
TREG1     ← 1 1 1 1 1 0 1 0      Set the timer register 1 s ÷ φT16 = 250 = FAH
INTET10   ← 1 1 0 1 - - - -      Enable INTT1, and set it to "Level 5".
TRUN      ← 1 X - - - - 1 -      Start timer 1 counting.
```

Note : ×: don't care –; no change

Use the following table for selecting the input clock.

Table 3.7 (1) 8-Bit Timer Interrupt Cycle and Input Clock

Input Clock	Interrupt Cycle (at fc = 16MHz)	Resolution	Interrupt Cycle (at fc = 20MHz)	Resolution
φT1 (8/fc)	0.5μs ~ 128μs	0.5μs	0.4μs ~ 102.4μs	0.4μs
φT4 (32/fc)	2μs ~ 512μs	2μs	1.6μs ~ 409.6μs	1.6μs
φT16 (128/fc)	8μs ~ 2.048ms	8μs	6.4μs ~ 1.638ms	6.4μs
φT256 (2048/fc)	128μs ~ 32.708ms	128μs	102.4μs ~ 2.621ms	128μs

Note: The input clock of timer 0 and timer 1 are different from as follows:

Timer 0: TI0 input, φT1, φT4, φT16

Timer 1: Match Output of Timer 0, φT1, φT16, φT256

② Generating a 50% duty square wave pulse

The timer flip-flop (TFF1) is inverted at constant intervals, and its status is output to timer output pin (TO1).

Example: To output a 3.0μs square wave pulse from TO1 pin at fc = 16MHz, set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	system clock (fsys)

	7 6 5 4 3 2 1 0	
TRUN	← – X – – – – 0 –	Stop timer 1, and clear it to "0".
TMOD	← 0 0 X X 0 1 – –	Set the 8-bit timer mode, and select ϕT1 (0.5 μs at fc = 16 MHz) as the input clock.
TREG1	← 0 0 0 0 0 0 1 1	Set the timer register at 3.0 μs ÷ ϕT1 ÷ 2 = 3.
TFFCR	← – – – – 1 0 1 1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P7CR	← X X X X – – 1 –	Select P71 as TO1 pin.
P7FC	← X X X X – – 1 X	
TRUN	← 1 X – – – – 1 –	Start timer 1 counting.

Note : X ; don't care – ; no change

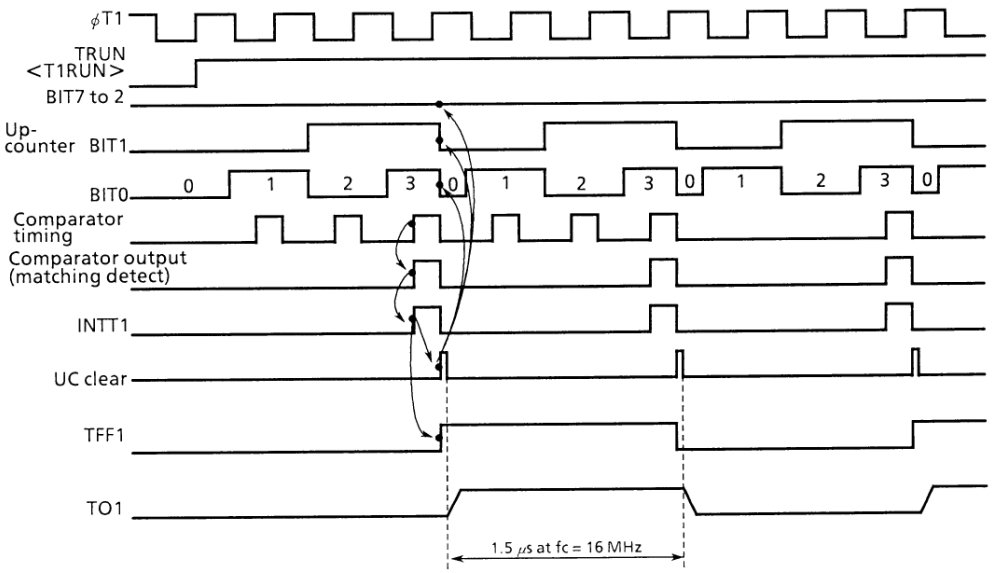


Figure 3.7 (7). Square Wave (50% Duty) Output Timing Chart

③ Making timer 1 count up by match signal from timer 0 comparator

Set the 8-bit timer mode, and set the comparator output of timer 0 as the input clock to timer 1.

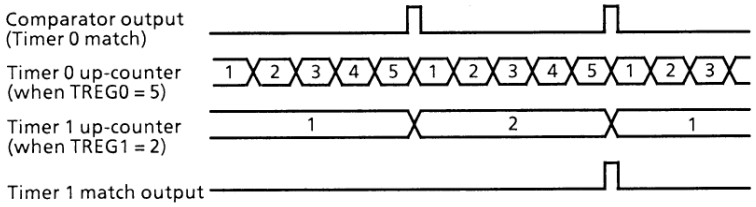


Figure 3.7 (8). Timer 1 Count Up by Timer 0

④ Output inversion with software

The value of timer flip-flop (TFF1) can be inverted, independent of timer operation.

Writing “00” into TFFCR <TFF1C1, 0> (memory address: 000025h of bit 3 and bit 2) inverts the value of TFF1.

⑤ Initial setting of timer flip-flop (TFF1)

The value of TFF1 can be initialized to “0” or “1”, independent of timer operation.

For example, write “10” in TFFCR <TFF1C1, 0> to clear TFF1 to “0”, while write “01” in TFFCR <TFF1C1, 0> to set TFF1 to “1”.

Note: The value of timer register cannot be read.

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1.

To make a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 0/timer 1 mode register TMOD <T10M1, 0> to “0, 1”.

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value of TMOD <T1CLK1, 0>. Table 3.7 (2) shows the relation between the cycle of timer (interrupt) and the selection of input clock.

Table 3.7 (2)

Input Clock	Interrupt Cycle (at fc = 16MHz)	Resolution	Interrupt Cycle (at fc = 20MHz)	Resolution
ϕT1 (8/fc)	0.5μs ~ 32.786ms	0.5μs	0.4μs ~ 26.214ms	0.4μs
ϕT4 (32/fc)	2μs ~ 131.072ms	2μs	1.6μs ~ 104.857ms	1.6μs
ϕT16 (128/fc)	8μs ~ 524.288ms	8μs	6.4μs ~ 419.430ms	6.4μs

The lower 8 bits of the timer (interrupt) cycle are set by the timer register TREG0, and the upper 8 bits are set by TREG1. Note that TREG0 always must be set first. (Writing data into TREG0 disables the comparator temporarily, and the comparator is restarted by writing data into TREG1.)

Setting example: To generate an interrupt INTT1 every 0.5 seconds at fc = 16MHz, set the following values for timer registers TREG0 and TREG1.

※ Clock Condition

system clock : High Frequency (fc)
clock gear : 1 (fc)
prescaler clock : system clock (fsys)

When counting with input clock of ϕT16 (8μs @ 16MHz)

0.5 sec ÷ 8μs = 62500 = F424H

Therefore, set TREG1 = F4H and TREG0 = 24H, respectively.

The comparator match signal is output from timer 0 each time the up-counter UC0 matches TREG0, where the up-counter UC0 is not to be cleared.

With the timer 1 comparator, the match detect signal is output at each comparator timing when up-counter UC1 and TREG1 values match. When the match detect signal is output simultaneously from both comparators of timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

Example: When TREG1 = 04H and TREG0 = 80H

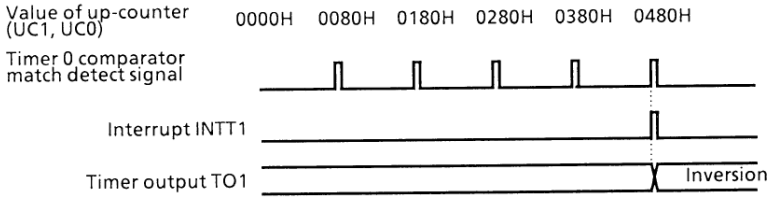


Figure 3.7 (9). Output Timer by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable Pulse Generation) Output mode

Square wave pulse can be generated at any frequency and duty by timer 0 and timer 1. The output pulse may be either low-active or high-active. In this mode, timer 1 cannot be used.

Timer 0 outputs pulse to TO1 pin (also used as P70).

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 TREG0 and TREG1.

However, it is required that the set value of TREG0 is smaller than that of TREG1.

Though the up-counter (UC1) of timer 1 is not used in this mode, UC1 should be set for counting by setting TRUN <T1RUN> to 1.

Figure 3.7 (11) shows the block diagram for this mode.

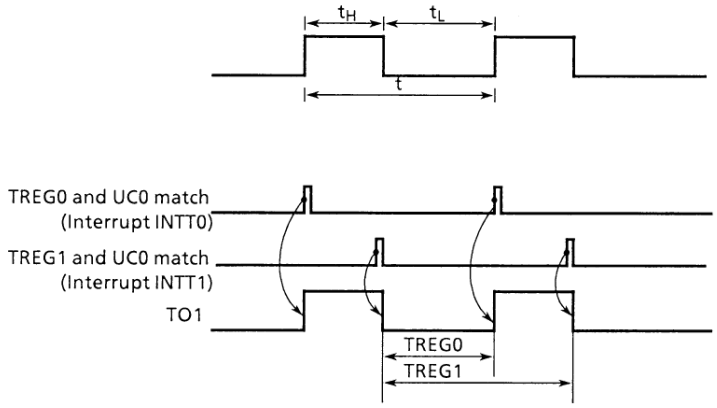


Figure 3.7 (10). 8-Bit PPG Output Waveforms

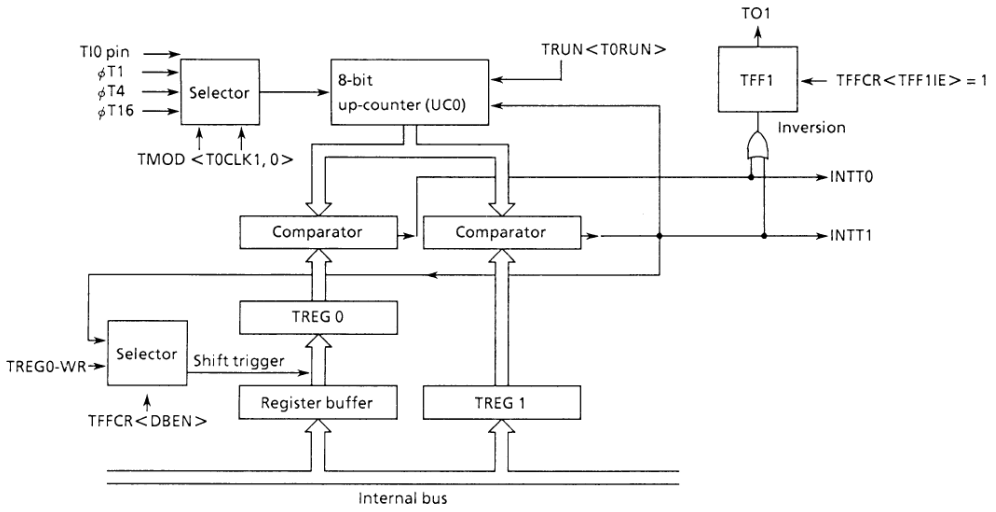


Figure 3.7 (11). Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TREG0 is enabled in this mode, the value of register buffer will be shifted in TREG0 each time TREG1 matches UC0.

Use of the double buffer makes easy handling of low duty waves (when duty is varied).

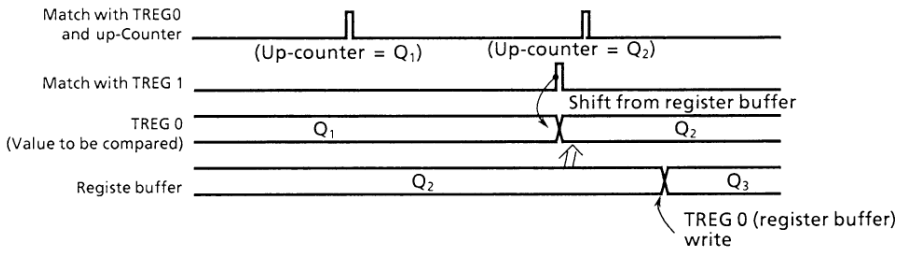
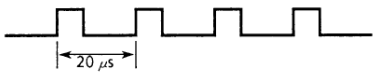


Figure 3.7 (12). Operation of Register Buffer

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



※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	system clock (fsys)

- Calculate the value to be set for timer register.
  To obtain the frequency 50kHz, the pulse cycle t should be: t = 1/50kHz = 20μs.
  Given ϕT1 = 0.5μs at 16MHz),
  20μs ÷ 0.5μs = 40
  Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H and then duty to 1/4, t x 1/4 = 20μs x 1/4 = 5μs
  5μs ÷ 0.5μs = 10
  Therefore, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

```
        7 6 5 4 3 2 1 0
TRUN  ← – X – – – – 0 0    Stop timer 0, and clear it to "0".
TMOD  ← 1 0 X X X X 0 1    Set the 8-bit PPG mode, and select φT1 as input clock.
TREG0 ← 0 0 0 0 1 0 1 0    Write "0AH".
TREG1 ← 0 0 1 0 1 0 0 0    Write "28H".
TFFCR ← – – – 1 0 1 1 X    Sets TFF1 and enable the inversion and double buffer
                           enable.
                           Writing "10" provides negative logic pulse.
P7CR  ← X X X X – – 1 –
P7FC  ← X X X X – – 1 X    Set P71 as the TO1 pin.
TRUN  ← 1 X – – – – 1 1    Start timer 0 and timer 1 counting.
```

Note : X ; don't care – ; no change

(4) 8-bit PWM Output mode

This mode is valid only for timer 0. In this mode, maximum 8-bit resolution of PWM pulse can be output.

PWM pulse is output to TO1 pin (also used as P71) when using timer 0. Timer 1 can also be used as 8-bit timer.

Timer output is inverted when up-counter (UC0) matches the set value of timer register TREG0 or when 2n - 1 (n = 6, 7, or 8; specified by T01MOD <PWM01, 0>) counter overflow occurs. Up-counter UC0 is cleared when 2n - 1 counter overflow occurs. For example, when n = 6, 6-bit PWM will be output, while when n = 7, 7-bit PWM will be output.

To use this PWM mode, the following conditions must be satisfied.

(Set value of timer register) <(Set value of 2^n - 1 counter overflow)

(Set value of timer register ≠ 0)

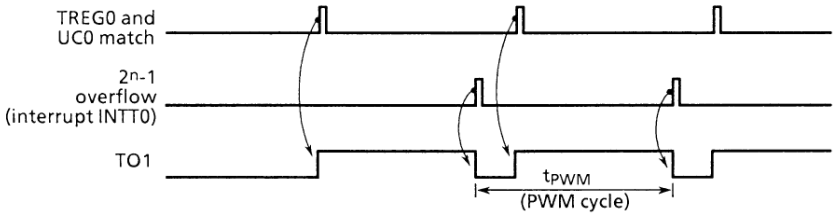


Figure 3.7 (13). 8-Bit PWM Waveforms

Figure 3.7 (14) shows the block diagram of this mode.

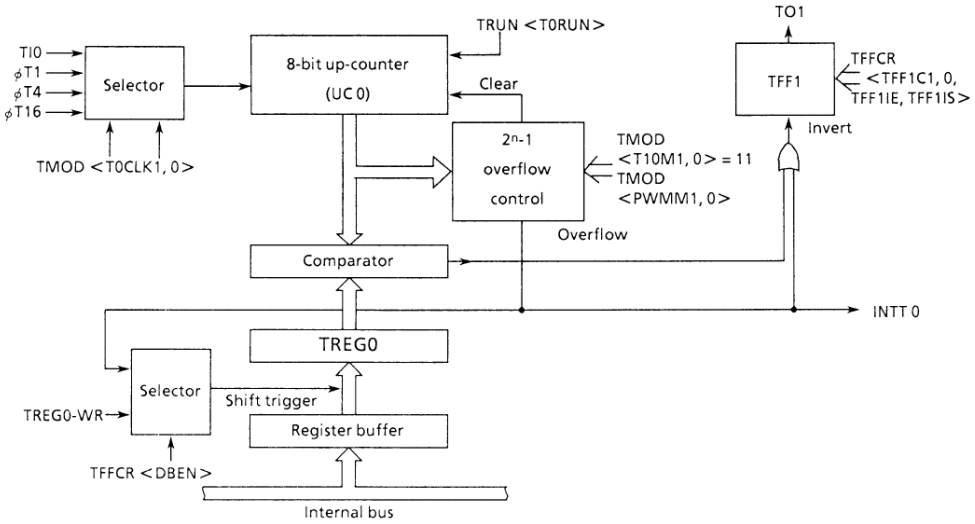


Figure 3.7 (14). Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TREG0 if 2^n - 1 overflow is detected when the double buffer of TREG0 is enabled.

Use of the double buffer makes easy the handling of small duty waves.

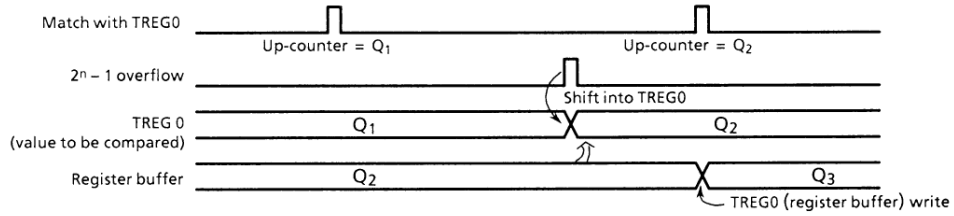
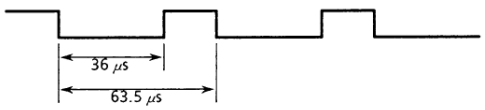


Figure 3.7 (15). Operation of Register Buffer

Example: To output the following PWM waves to TO1 pin at fc = 16MHz.



※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	system clock (fsys)

To realize 63.5μs of PWM cycle by ϕT1 = 0.5μs (@ fc = 16MHz),

$$63.5\mu s \div 0.5\mu s = 127 = 2^7 - 1$$

Consequently, n should be set to 7.

As the period of low level is 36μs, for ϕT1 = 0.5μs, set the following value for TREG0:

$$36\mu s \div 0.5\mu s = 72 = 48H$$

```
        MSB                LSB
          7 6 5 4 3 2 1 0
TRUN  ← – X – – – – – 0        Stop timer 0, and clear it to "0".
TMOD  ← 1 1 1 0 – – 0 1        Set 8-bit PWM mode (cycle: 2^7 – 1) and select φT1 as the input
                               clock.
TREG0 ← 0 1 0 0 1 0 0 0        Writes "48H".
TFFCR ← X X X X 1 0 1 X        Clears TFF1, enable the inversion and double buffer.

P7CR  ← X X X X – – 1 –  }     Set P71 as the TO1 pin.
P7FC  ← X X X X – – 1 X  }
TRUN  ← 1 X – – – – – 1        Start timer 0 counting.
```

Note : X ; don't care – ; no change

Table 3.7 (3) PWM Cycle

at fc = 16 MHz, fs = 32 kHz

select prescaler clock <PRCK1, 0>	select system clock <SYSCK>	Gear value <GEAR2 : 0>	PWM Cycle								
			2^6 - 1			2^7 - 1			2^8 - 1		
			φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
00 (f_{FPH})	1 (fs)	XXX	15.75 ms	63.00 ms	252.00 ms	31.75 ms	127.00 ms	508.00 ms	63.75 ms	255.00 ms	1.02 s
	0 (fc)	000 (fc)	31.5 μs	126.0 μs	504.0 μs	63.5 μs	254.0 μs	1.02 ms	127.5 μs	510.0 μs	2.04 ms
		001 ($fc/2$)	63.0 μs	252.0 μs	1.01 ms	127.0 μs	508.0 μs	2.03 ms	255.0 μs	1.02 ms	4.08 ms
		010 ($fc/4$)	126.0 μs	504.0 μs	2.02 ms	254.0 μs	1.02 ms	4.06 ms	510.0 μs	2.04 ms	8.16 ms
		011 ($fc/8$)	252.0 μs	1.01 ms	4.03 ms	508.0 μs	2.03 ms	8.13 ms	1.02 ms	4.08 ms	16.32 ms
		100 ($fc/16$)	504.0 μs	2.02 ms	8.06 ms	1.02 ms	4.06 ms	16.26 ms	2.04 ms	8.16 ms	32.64 ms
01 (low frequency clock)	XXX	XXX	15.75 ms	63.00 ms	252.00 ms	31.75 ms	127.00 ms	508.00 ms	63.75 ms	255.00 ms	1.02 s
10 ($fc/16$ clock)	XXX	XXX	504.0 μs	2.02 ms	8.06 ms	1.02 ms	4.06 ms	16.26 ms	2.04 ms	8.16 ms	32.64 ms

XXX : don't care

(5) Table 3.7 (4) shows the list of 8-bit timer modes.

Table 3.7 (4) Timer Mode Setting Registers

Register name	TMOD				TFFCR
Name of function in	T10M	PWMM	T1CLK	T0CLK	TFF1IS
Function	Timer mode	PWM0 cycle	Upper timer input clock	Lower timer input clock	Timer F/F invert signal select
16-bit timer mode	01	–	–	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 2 channels	00	–	Lower timer match: ϕT1, 16, 256 (00, 01, 10, 11)	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG × 1channel	10	–	–	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit PWM × 1channel	11	2^6-1, 2^7-1, 2^8-1 (01, 10, 11)	–	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 1channel	11	–	ϕT1, ϕT16, ϕT256 (01, 10, 11)	–	Output disabled

Note : - ; Don't care

3.8 8-Bit PWM Timer

The TMP93CM40/TMP93CM41 has two built-in 8-bit PWM timers (timers 2 and 3).

They have two operating modes.

- 8-bit PWM (pulse width modulation: variable duty at fixed interval) output mode
- 8-bit interval timer mode

Figure 3.8 (1), (2) are block diagrams of the 8-bit PWM timer (timers 2 and 3).

PWM timers consist of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Two timer flip-flops (TFF2 for timer 2 and TFF3 for timer 3) are provided.

Input clocks ϕP1, ϕP4, and ϕP16 for the PWM timers can be obtained using the built-in prescaler.

PWM timer operating mode and timer flip-flops are controlled by four control registers (P0MOD, P1MOD, PFFCR, and TRUN).

PWM timer 0 and 1 can be used independently.

All PWM timers operate in the same manner, thus, only the operation of PWM timer 0 will be explained below.

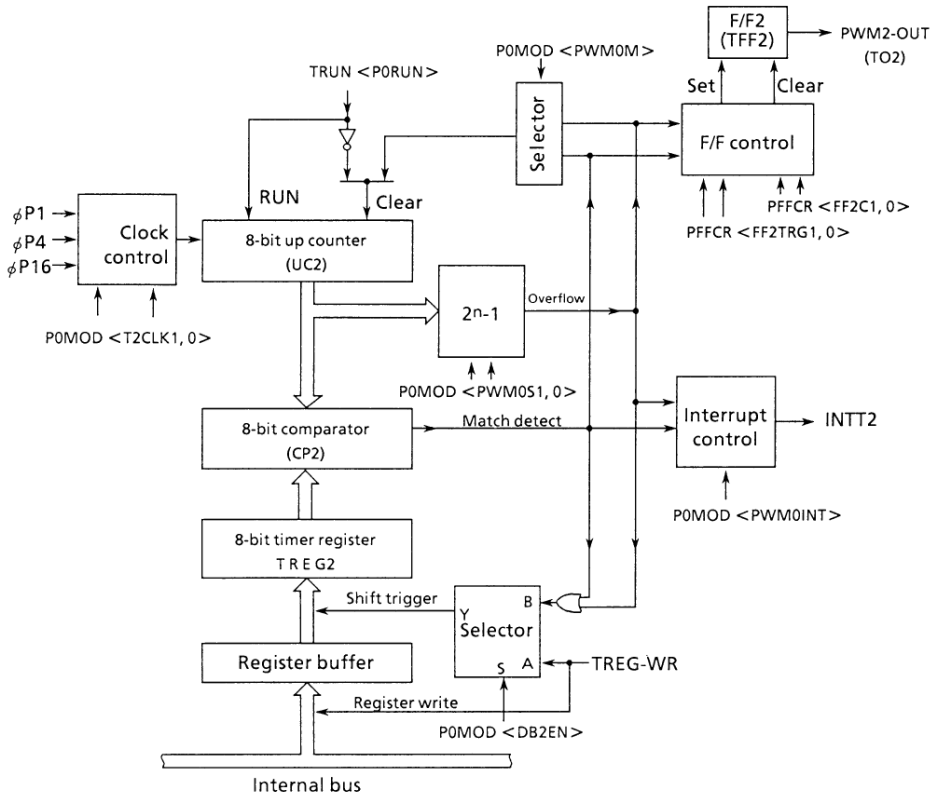


Figure 3.8 (1). Block Diagram of 8-Bit PWM Timer 0 (Timer 2)

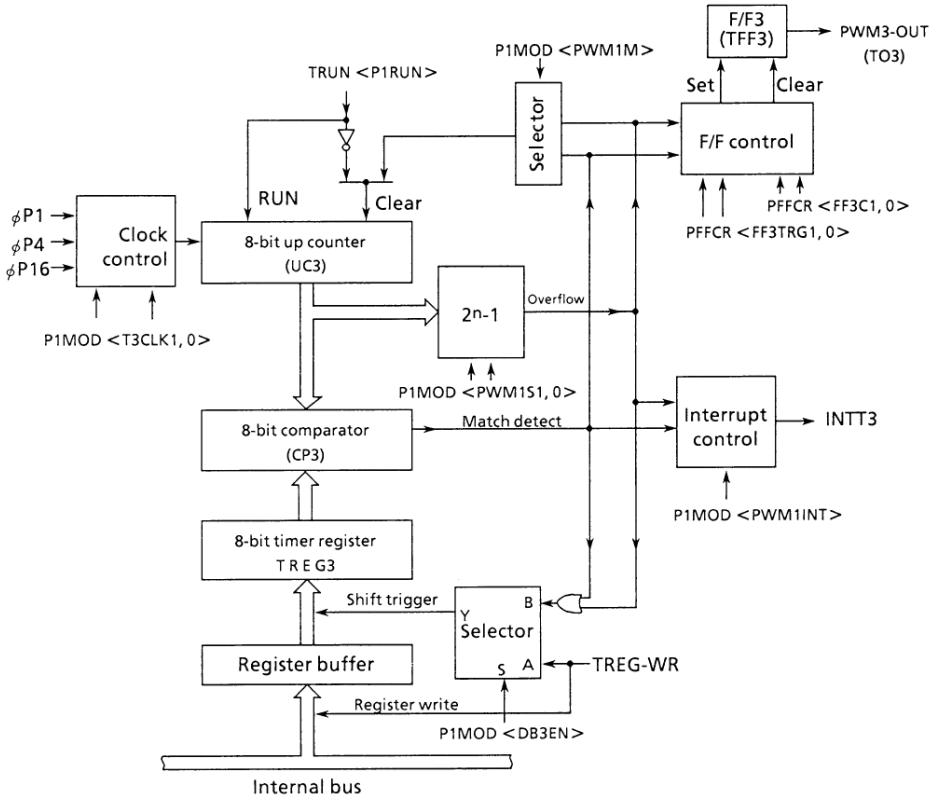


Figure 3.8 (2). Block Diagram of 8-Bit PWM Timer 1 (Timer 3)

① Prescaler

There are 5 bit prescaler and prescaler clock selection register to generate input clock for 8 bit PWM Timer 0, 1.

Figure 3.8 (3) shows the block diagram. Table 3.8 (1) shows prescaler clock resolution to 8 bit PWM Timer 0, 1.

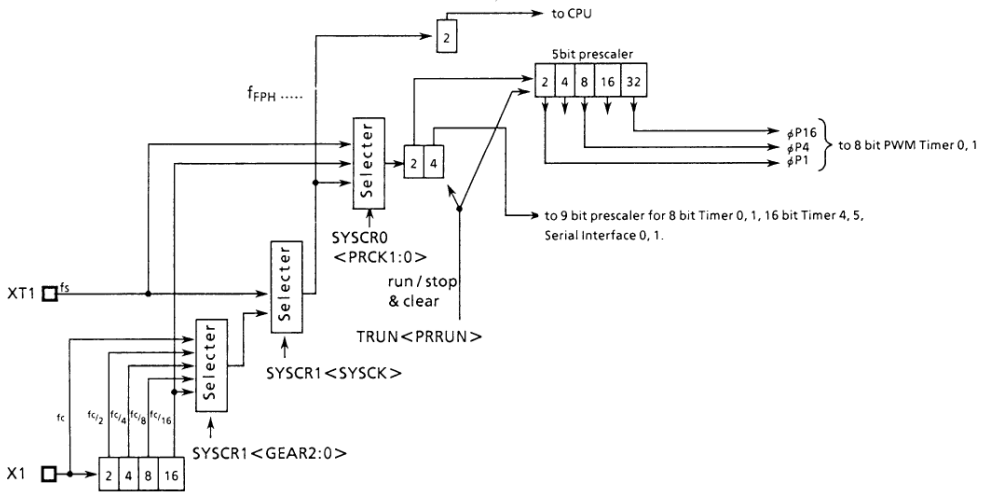


Figure 3.8 (3). Prescaler Block Diagram]

Table 3.8 (1) Prescaler Clock Resolution to 8 Bit PWM Timer 0, 1

at fc = 16 MHz, fs = 32 kHz

Select system clock <SYSCK>	Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 : 0>	Prescaler Clock Resolution		
			φP1	φP4	φP16
1 (fs)	00 (f_{FPH})	XXX	$fs/2^2$ (125 μs)	$fs/2^4$ (500 μs)	$fs/2^6$ (2 ms)
0 (fc)		000 (fc)	$fc/2^2$ (0.25 μs)	$fc/2^4$ (1 μs)	$fc/2^6$ (4 μs)
		001 ($fc/2$)	$fc/2^3$ (0.5 μs)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)
		010 ($fc/4$)	$fc/2^4$ (1 μs)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)
		011 ($fc/8$)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)
		100 ($fc/16$)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{10}$ (64 μs)
XXX	01 (low frequency clock)	XXX	$fs/2^2$ (125 μs)	$fs/2^4$ (500 μs)	$fs/2^6$ (2 ms)
XXX	10 ($fc/16$ clock)	XXX	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{10}$ (64 μs)

XXX : don't care

(Note) The fc/16 clock as a prescaler clock can not be used when the fs is used as a system clock.

The 1/2 times clock selected among f_{FPH} clock, fc/16 clock, and fs clock is input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1 : 0>.

Resetting sets <PRCK1 : 0> to "00", therefore, f_{FPH} /2 clock is input. The register TRUN <PRRUN> which controls this prescaler is also used at 9 bit prescaler. So, this prescaler cannot be operated independently.

The 8 bit Timer 0, 1 uses 3 types of clock: øP1, øP4, and øP16 among the prescaler outputs.

The prescaler can be run or stopped by TRUN <PRRUN> described of the 8 bit Timer.

Counting starts when <PRRUN> is set to "1", while the prescaler is cleared to zero and stops operation when <PRRUN> is set to "0".

When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop this prescaler before "HALT" instruction is executed.

② Up-counter

An 8-bit binary counter which counts up using the input clock specified by PWM mode register P0MOD <T2CLK1:0>.

The input clock for the PWM0 is selected from the internal clocks ϕP1, ϕP4, and ϕP16 (PWM dedicated prescaler output) depending on the <T2CLK1:0>.

Operating mode is also set by P0MOD <PWM0M>. At reset, they are initialized to "0", thus, the up-counter is in PWM mode. In PWM mode, the up-counter is cleared when a 2^n - 1 overflow occurs; in timer mode, the up-counter is cleared at compare and match.

Count/stop and clear of the up-counter can be controlled for each PWM timer using the timer operation control register TRUN. Resetting clears all up-counters and stops timers.

③ Timer registers

The 8-bit register is used for setting an interval time. When the value set in the timer register (TREG 2) matches the value in the up-counter, the match detect signal of the comparator becomes active.

Timer register TREG2 is paired with register buffer to make a double buffer structure.

TREG2 is a double buffer enable/disable controlled by P0MOD <DB2EN> : disabled when <DB2EN> = 0, enabled when <DB2EN> = 1.

Data is transferred from register buffer to timer when a 2^n - 1 overflow occurs in the PWM mode, or when compare and match occurs in 8-bit timer mode. That is, with a PWM timer, the timer mode can be operated in double buffer enable state, unlike timer mode for timers 0 and 1.

At reset, <DB2EN> is initialized to 0 to disable double buffer. To use double buffer, write the data in the timer register at first, then set <DB2EN> to 1, and write the following data in the register buffer.

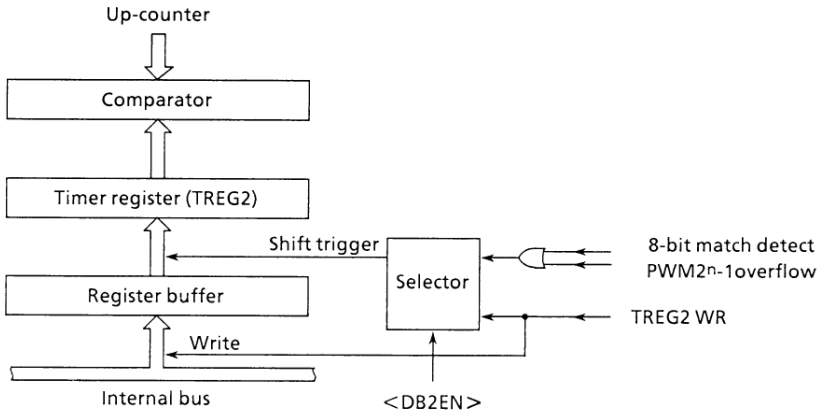


Figure 3.8 (4). Structure of Timer Registers 2

Note: The timer register and register buffer are allocated to the same memory address. When <DB2EN> = 0, the same value is written to both register buffer and timer register. When <DB2EN> = 1, the value is written to the register buffer only.

Memory addresses of the timer registers are as follows:

TREG2 : 000026H

TREG3 : 000027H

Both timer registers are write only; however, register buffer values can be read when reading the above addresses.

④ Comparator

Compares the value in the up-counter with the value in the timer register (TREG2). When they match, the comparator outputs the match detect signal. A timer interrupt (INTT2) is generated at compare and match if the interrupt select bit <PWM0INT> of the mode register (P0MOD) is set to 1. In timer mode, the comparator clears the up-counter to 0 at compare and match. It also inverts the value of the timer flip-flop if timer flip-flop invert is enabled.

⑤ Timer flip-flop

The value of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer or 2^n - 1 overflow. The value can be output to the timer output pin TO2 (also used as P72).

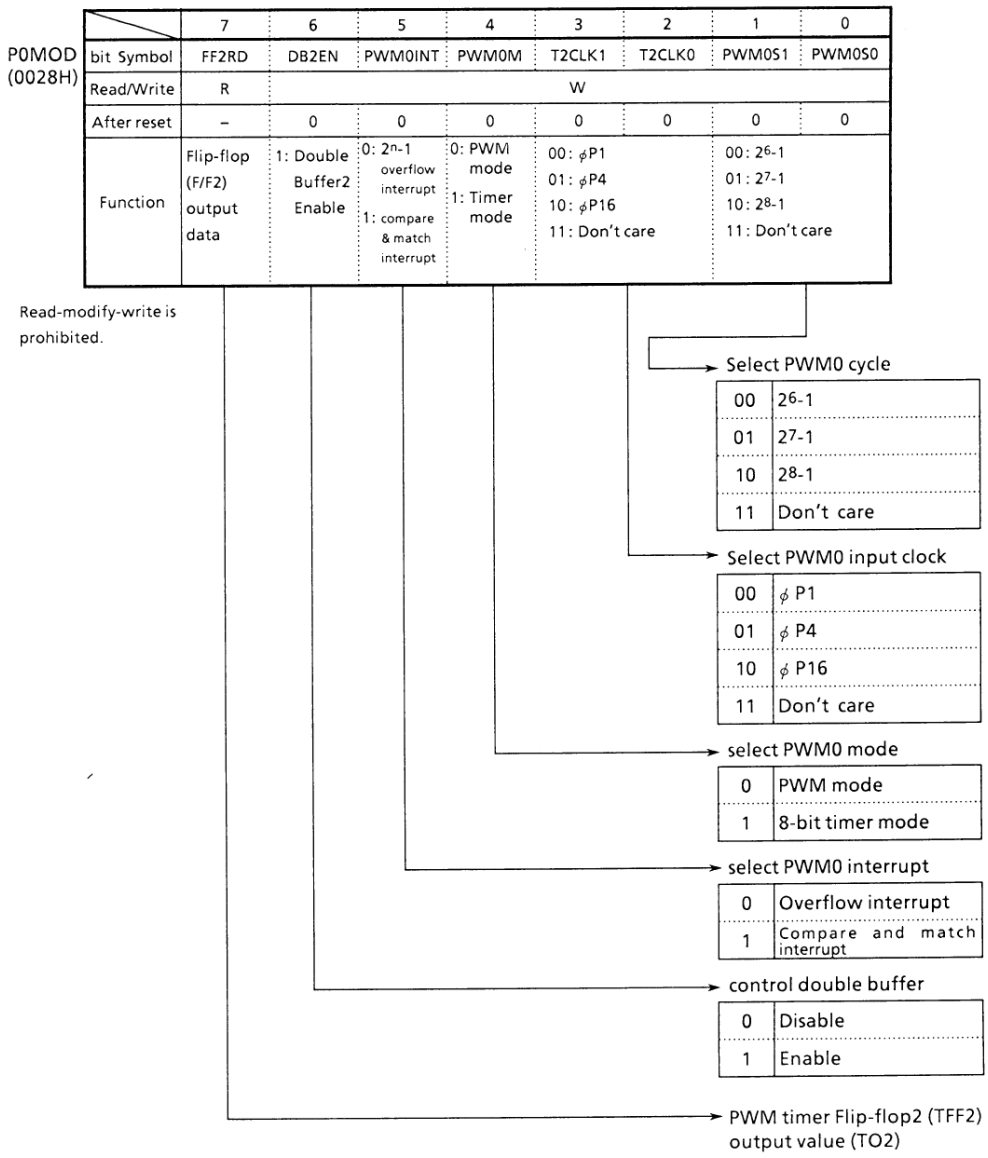


Figure 3.8 (5). 8-Bit PWM0 Mode Control Register

P1MOD (0029H)

	7	6	5	4	3	2	1	0
bit Symbol	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0
Read/Write	R	W						
After reset	–	0	0	0	0	0	0	0
Function	Flip-flop (F/F3) output data	1: Double Buffer3 Enable	0: 2^{n}-1 overflow interrupt 1: Compare and match interrupt	0: PWM mode 1: Timer mode	00: ϕP1 01: ϕP4 10: ϕP16 11: Don't care		00: 2^{6}-1 01: 2^{7}-1 10: 2^{8}-1 11: Don't care	

Read-modify-write is prohibited.

Select PWM1 cycle

00	2^{6}-1
01	2^{7}-1
10	2^{8}-1
11	Don't care

Select PWM1 input clock

00	ϕ P1
01	ϕ P4
10	ϕ P16
11	Don't care

Select PWM1 mode

0	PWMmode
1	8-bit timer mode

Select PWM1 interrupt

0	Overflow interrupy
1	Compare and match interrupt

Control double buffer

0	Disable
1	enable

PWM timer Flip-flop3 (TFF3) output value (TO3)

Figure 3.8 (6). 8-Bit PWM1 Mode Control Register

PFFCR (002AH)

	7	6	5	4	3	2	1	0
bit Symbol	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
Read/Write	W		R/W		W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00 : Don't care 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		00 : Disable TFF3 inverted. 01 : Invert by match. 10 : Set by match ; clear by overflow, 11 : Clear by match ; set by overflow.		00 : Don't care 01 : Set TFF2 10 : Clear TFF2 11 : Don't care		00 : Disable TFF2 inverted. 01 : Invert by match. 10 : Set by match ; clear by overflow, 11 : Clear by match ; set by overflow.	

Select PWM timer F/F2 (TFF2) trigger

00	Disable TFF2 trigger.
01	Invert by compare and match.
10	Set by compare and match. Clear by 2^{n-1} overflow.
11	Clear by compare and match. Set by 2^{n-1} overflow.

Control PWM timerF/F2 (TFF2)

00	Don't care
01	Set TFF2 to "1".
10	Clear TFF2 to "0".
11	Don't care

Select PWM timer F/F3 (TFF3) trigger

00	Disable TFF3 trigger.
01	Invert by compare and match.
10	Set by compare and match. Clear by 2^{n-1} overflow.
11	Clear by compare and match. Set by 2^{n-1} overflow.

Control PWM timerF/F3 (TFF3)

00	Don't care
01	Set TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care

Figure 3.8 (7). 8-Bit PWM F/F Control Register

TRUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler & Timer Run / Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							

Count Operation

0	Stop and clear
1	Count

PRRUN : Operation of prescaler
T5RUN : Operation of 16-bit timer (timer5)
T4RUN : Operation of 16-bit timer (timer4)
P1RUN : Operation of PWM timer (PWM1 / timer3)
P0RUN : Operation of PWM timer (PWM0 / timer2)
T1RUN : Operation of 8-bit timer (timer1)
T0RUN : Operation of 8-bit timer (timer0)

SYSCR0 (006EH)

	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	0	0	0	0	0	0
Function	High Frequency oscillator (fc) 0 : stop 1 : oscillator	Low Frequency oscillator (fc) 0 : stop 1 : oscillator	High Frequency oscillator (fc) after released STOP mode 0 : stop 1 : oscillator	Low Frequency oscillator (fc) after released STOP mode 0 : stop 1 : oscillator	select clock after released STOP mode 0 : fc 1 : fs	Warming Up Timer 0 write : don't care 1 write : start timer 0 read : end warming up 1 read : not end warming up	select gear value of high frequency (fs) 00 : f_{FPH} 01 : fs 10 : fc/16 11 : (reserved)	

select gear value of high frequency

00	f_{FPH}
01	fs
10	fc/16
11	(reserved)

} divided clock

Figure 3.8 (8). Timer Operation Control Register/System Clock Control Register

The following explains PWM timer operations.

(1) PWM timer mode

PWM output changes under the following two conditions.

Condition 1:

- TFF2 is cleared to 0 when the value in the up-counter (UC2) and the value set in the TREG2 match.
- TFF2 is set to 1 when a 2^n - 1 counter overflow (n = 6, 7, or 8) occurs.

Condition 2:

- TFF2 is set to 1 when the value in the up-counter (UC2) and the value set in TREG2 match.
- TFF2 is cleared to 0 when a 2^n - 1 counter overflow (n = 6, 7, or 8) occurs.

The up-counter (UC2) is cleared by a 2^n - 1 counter overflow.

The PWM timer can output 0% to 100% duty pulses because a 2^n - 1 counter overflow has a higher priority. That is, to obtain 0% output (always low), the mode used to set TFF2 to 0 due to overflow (PFFCR <FF2TRG1, 0> = 1, 0) must be set and 2^n - 1 (value for overflow) must be set in TREG2. To obtain 100% output (always high), the mode must be changed: PFFCR <FF2TRG1, 0> = 1,1 then the same operation is required.

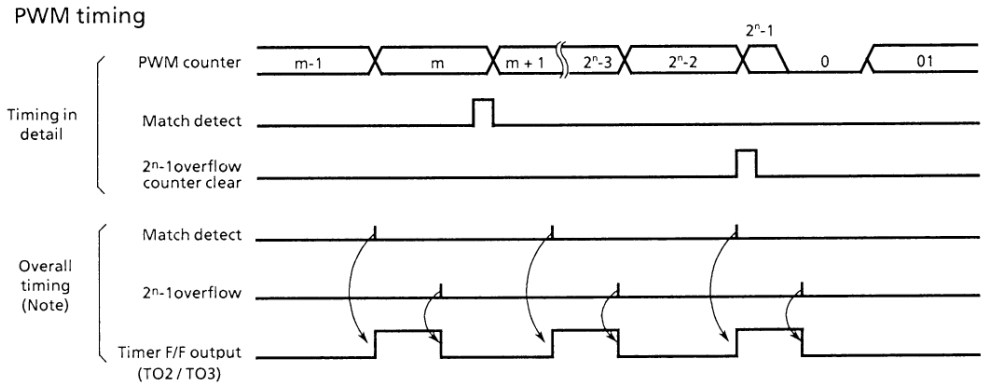


Figure 3.8 (9). Output Waves in PWM Timer Mode

Note: The above waves are obtained in a mode where the F/F is set by a match with the timer register (TREG) and reset by an overflow.

Figure 3.8 (10) is a block diagram of this mode.

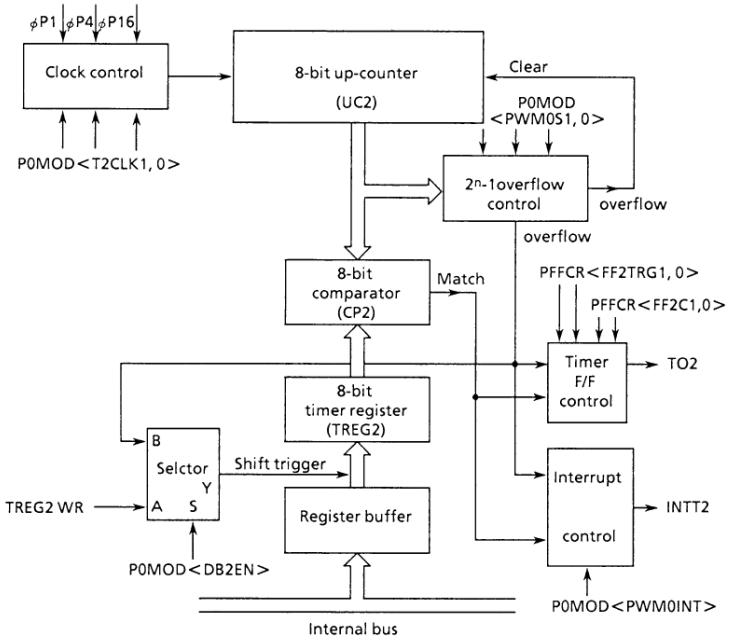


Figure 3.8 (10). Block Diagram of PWM Timer Mode (PWM0)

In this mode, enabling double buffer is very useful. The register buffer value shifts into TREG2 when a 2^n -1 overflow is detected, when double buffer is enabled.

Using double buffer makes handling small duty waves easy.

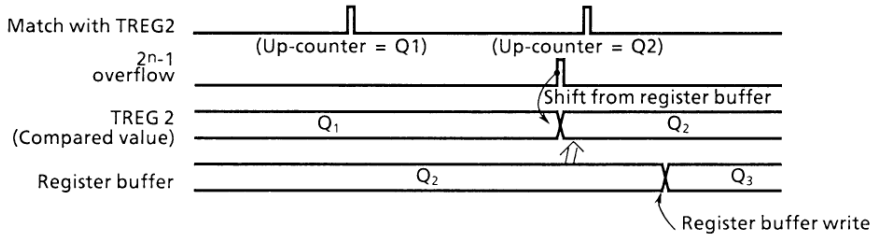
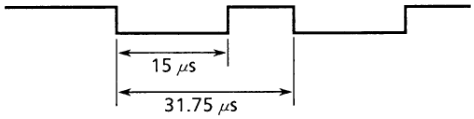


Figure 3.8 (11). Register Buffer Operation

Example: To output the following PWM waves to TO2 pin using PWM0 at fc = 16MHz.

※ Clock Condition

system clock : High frequency (fc)
clock gear : 1 (fc)
prescaler clock : system clock (fsys)



To implement 31.75μs PWM cycle by ϕP1 = 0.25μs (@ fc = 16MHz)

$$31.75\mu s \div 0.25\mu s = 127 = 2^7 - 1.$$

Consequently, set n to 7.

Since the low level cycle = 15μs; for ϕP1 = 0.25μs

$$15\mu s \div 0.25 = 60 = 3CH$$

set the 3CH in TREG2.

```
        7 6 5 4 3 2 1 0
TRUN  ← - X - - - 0 - -     Stops PWM0 and clears it to 0.
P0MOD ← - 0 0 0 0 0 0 1     Sets PWM (2^7-1) mode, input clock φP1, overflow
                            interrupt, and disables double buffer.
TREG2 ← 0 0 1 1 1 1 0 0     Writes 3CH.
P0MOD ← - 1 0 0 0 0 0 1     Enables double buffer.
PFFCR ← - - - - 0 1 1 0     Sets TFF2 and a mode where TFF2 is set by compare and
                            match, and cleared by overflow.
P7CR  ← X X X X - 1 - -  }  Sets P72 as TO2 pin
P7FC  ← X X X X - 1 - X  }
TRUN  ← 1 X - - - 1 - -     Starts PWM0 counting.
```

Note: X; don't care - ; no change

Table 3.8 (2) PWM Cycle and 2^n -1 Counter Setting

at fc = 16 MHz, fs = 32 kHz

select system clock <SYSCK>	select prescaler clock <PRCK1, 0>	Gear value <GEAR2 : 0>	PWM Cycle								
			2^6 - 1			2^7 - 1			2^8 - 1		
			φP1	φP4	φP16	φP1	φP4	φP16	φP1	φP4	φP16
1 (fs)	00 (f_{FPH})	XXX	7.88 ms	31.50 ms	126.00 ms	15.88 ms	63.50 ms	254.00 ms	31.88 ms	127.50 ms	510.00 ms
0 (fc)		000 (fc)	15.8 μs	63.0 μs	252.0 μs	31.8 μs	127.0 μs	508.0 μs	63.8 μs	255.0 μs	1.02 ms
		001 ($fc/2$)	31.5 μs	126.0 μs	504.0 μs	63.5 μs	254.0 μs	1.02 ms	127.5 μs	510.0 μs	2.04 ms
		010 ($fc/4$)	63.0 μs	252.0 μs	1.01 ms	127.0 μs	508.0 μs	2.03 ms	255.0 μs	1.02 ms	4.08 ms
		011 ($fc/8$)	127.0 μs	508.0 μs	2.03 ms	254 μs	1.02 ms	4.06 ms	510.0 μs	2.04 ms	8.16 ms
		100 ($fc/16$)	252.0 μs	1.01 ms	4.03 ms	508.0 μs	2.03 ms	8.13 ms	1.02 ms	4.08 ms	16.32 ms
XXX	01 (low frequency)	XXX	7.88 ms	31.50 ms	126.00 ms	15.88 ms	63.50 ms	254.00 ms	31.88 ms	127.50 ms	510.00 ms
XXX	10 ($fc/16$ clock)	XXX	252.0 μs	1.01 ms	4.03 ms	508.0 μs	2.03 ms	8.13 ms	1.02 ms	4.08 ms	16.32 ms

XXX : don't care

(2) 8-bit timer mode

Both PWM timers can be used independently as 8-bit interval timers. Since both timers operate in exactly the same way, PWM0 (timer 2) is used for the purposes of explanation.

① Generating interrupts at a fixed interval

To generate timer 2 interrupt (INTT2) at a fixed interval using PWM0 timer, first stop PWM0, then set the operating mode, input clock, and interval in the P0MOD and TREG2 registers. Next, enable INTT2 and start counting PWM0.

Example: To generate a timer 2 interrupt every 40μs at fc = 16MHz, set registers as follows:

※ Clock Condition

- system clock : High frequency (fc)
- clock gear : 1 (fc)
- prescaler clock : system clock (fsys)

```
                7 6 5 4 3 2 1 0
TRUN      ← - X - - - 0 - -      Stops PWM0 and clears it to 0.
P0MOD     ← X 0 1 1 0 0 X X      Sets 8-bit timer mode and selects φP1 (0.25 μs) and
                                 compare interrupt.
TREG2     ← 1 0 1 0 0 0 0 0      Sets 40 μs / 0.25 μs = A0H in timer register.
INTEPW10  ← - - - - 1 1 0 0      Enables INTT2 and sets interrupt level 4.
TRUN      ← 1 X - - - 1 - -      Starts counting PWM0.
```

Note : X ; don't care - ; no change

Select an input clock using Table 3.8 (1).

Note: To generate interrupts in 8-bit timer mode, bit 5 (interrupt control bit <PWM0INT> must be set to 1.

② Generating a 50% square wave

To generate a 50% square wave, invert the timer flip-flop at a fixed interval and output the timer flip-flop value to the timer output pin (TO2).

Example: To output a 3.0μs square wave at fc = 16MHz from TO2 pin, set register as follows:

※ Clock Condition

- system clock : High frequency (fc)
- clock gear : 1 (fc)
- prescaler clock : system clock (fsys)

	7 6 5 4 3 2 1 0	
TRUN	← - X - - - 0 - -	Stops PWM0 and clears it to 0.
P0MOD	← X 0 1 1 0 0 X X	Sets 8-bit timer mode and selects ϕP1 (0.25 μs) as the input clock.
TREG2	← 0 0 0 0 0 1 1 0	Sets 3.0 μs / 0.25 μs / 2 = 6 in the timer register.
PFFCR	← - - - - 1 0 0 1	Clears TFF2 to 0 and inverts using comparator output.
P7CR	← X X X X - 1 - -	Sets P72 as TO2 pin.
P7FC	← X X X X - 1 - X	
TRUN	← 1 X - - - 1 - -	Starts counting PWM0.

Note: X; don't care -; no change

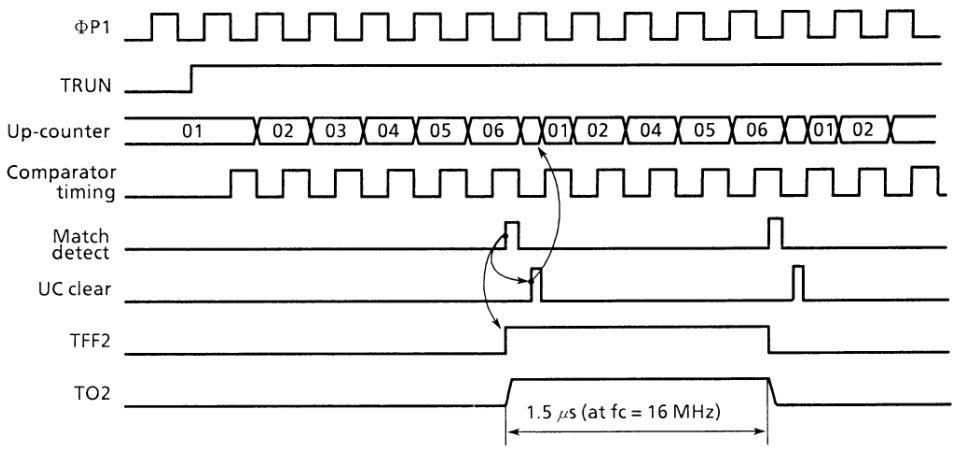


Figure 3.8 (12). Square Wave (50% Duty) Output Timing Chart

This mode is as shown in Figure 3.8 (13) below.

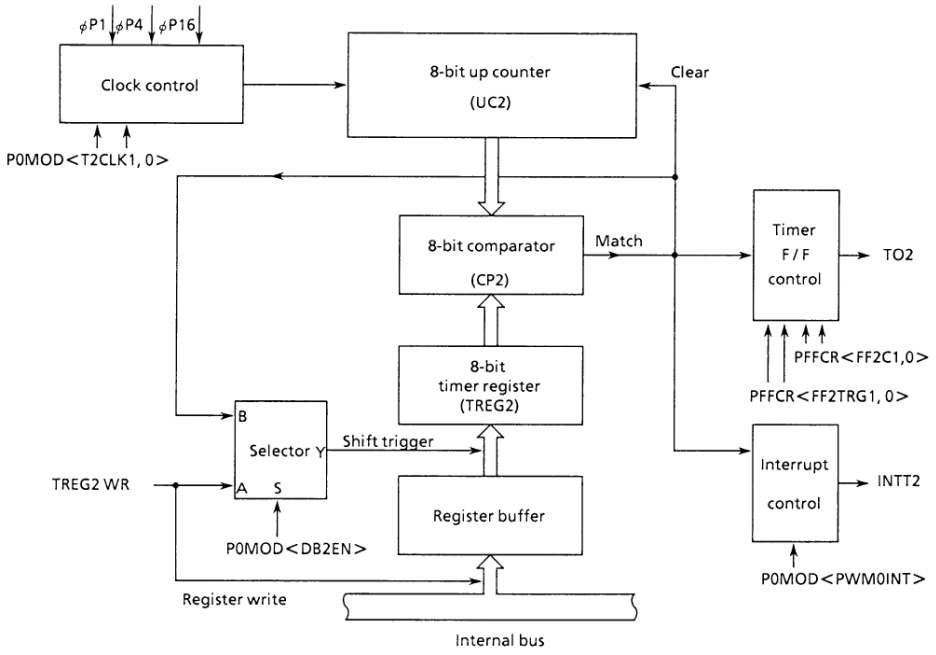


Figure 3.8 (13). Block Diagram of 8-Bit Timer Mode

3.9 16-Bit Timer

TMP93CM40/TMP93CM41 contains two (timer 4 and timer 5) 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
- 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, two 16-bit capture registers (one of them applies double-buffer), two comparators, capture input controller, and timer flip-flop and the control circuit.

Timer/event counter is controlled by four control registers: T4MOD/T5MOD, T4FFCR/T5FFCR, TRUN and T45CR.

Figure 3.9 (1), (2) show the block diagram of 16-bit timer/event counter (timer 4 and timer 5).

Timer 4 and 5 can be used independently.

All timers operate in the same manner except the following points, thus, only Timer 4 operation will be explained below.

Different Points Between Timer 4 and 5

	Timer 4	Timer 5
Timer Out Pin (for upper timer register)	TO5 pin (TFF5)	–
Different Phased Pulse Output Mode	Exist	Does not exist (not TO7 pin)

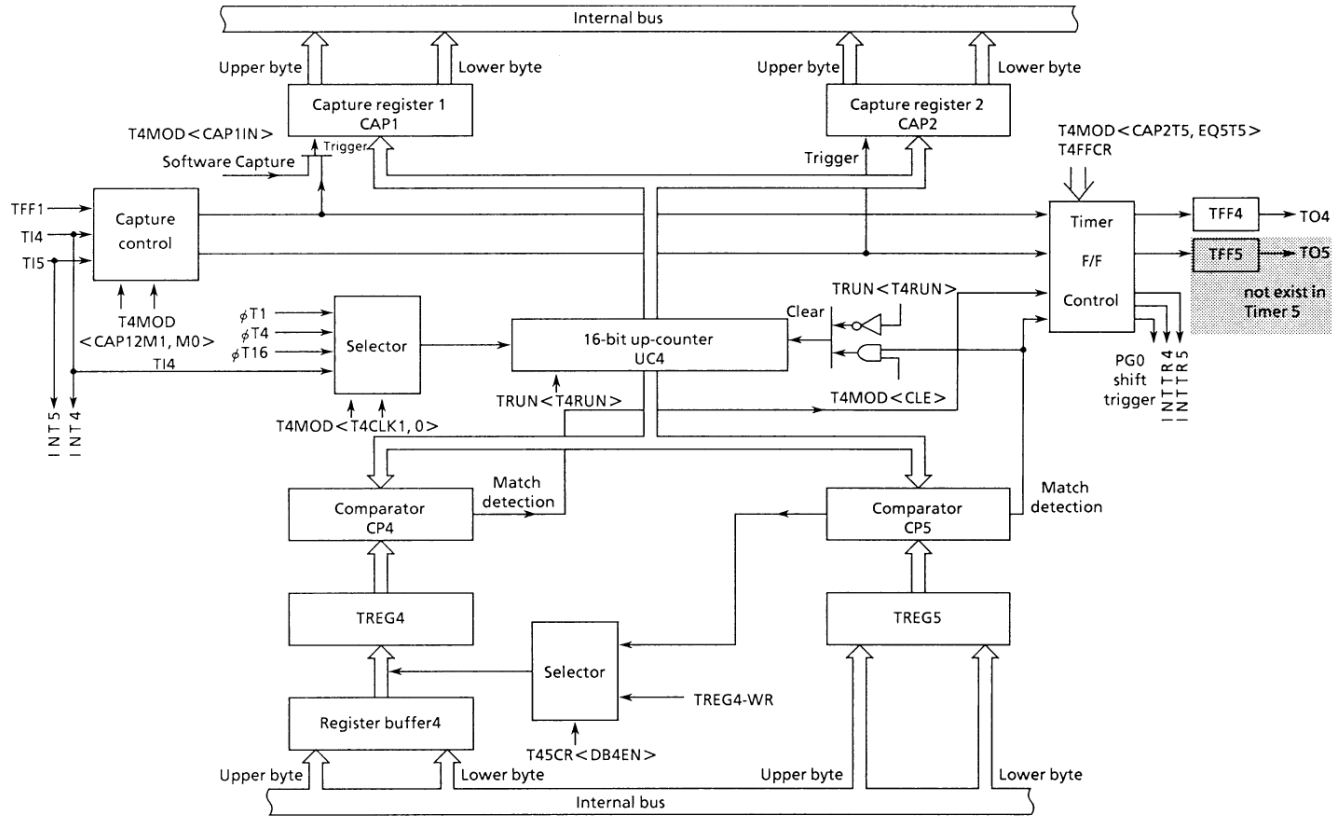


Figure 3.9 (1). Block Diagram of 16-Bit Timer (Timer 4)

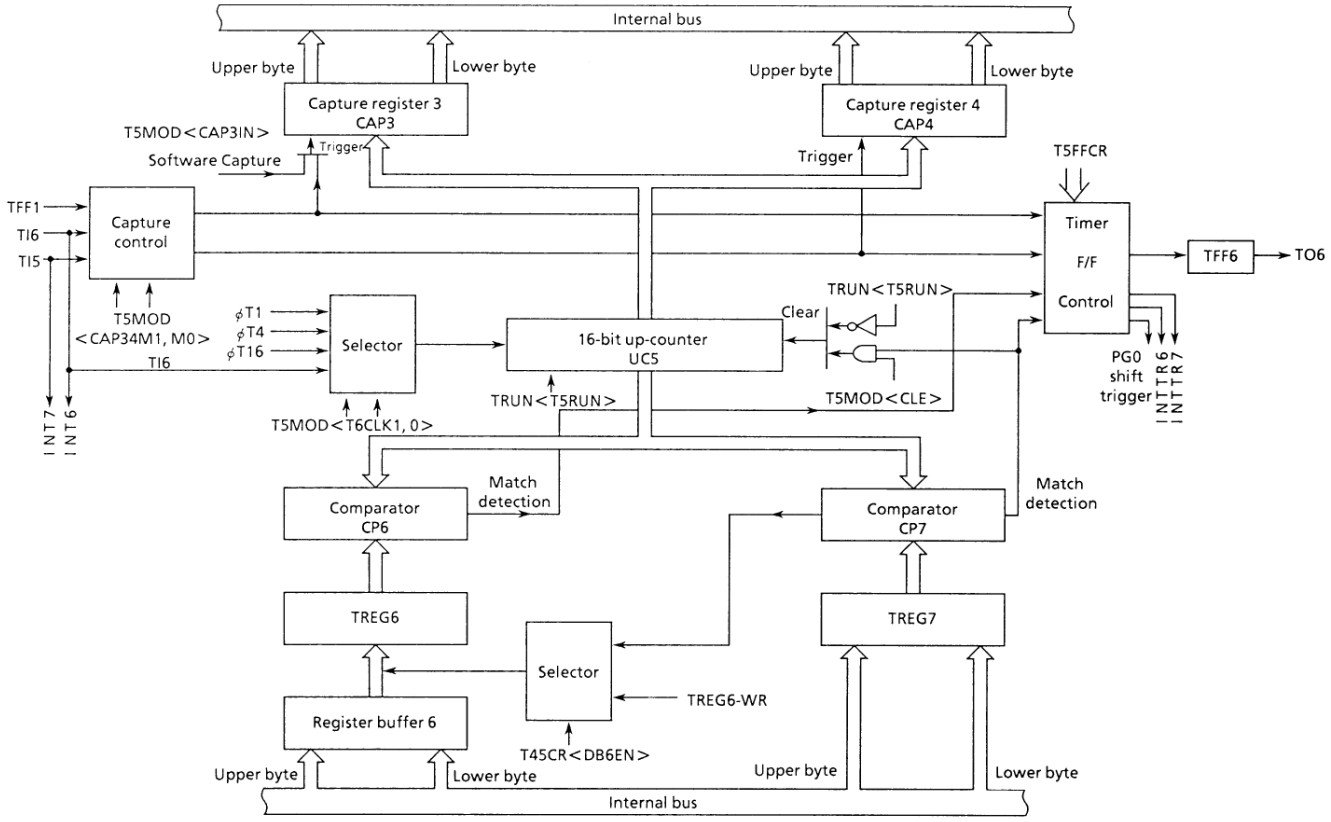


Figure 3.9 (2). Block Diagram of 16-Bit Timer (Timer 5)

T4MOD (0038H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT4 occurs at rise edge. 01: TI4↑ TI5↑ INT4 occurs at rise edge. 10: TI4↑ TI4↓ INT4 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC4 Clear Enable	Timer 4 source clock 00: TI4 01: φT1 10: φT4 11: φT16	
	Invert when the UC value is loaded to CAP2	Invert when the up-counter matches TREG5						

Timer 4 input clock

00	External clock (TI4)
01	φT1
10	φT4
11	φT16

Clearing the up-counter UC4

0	Clear disable
1	Clear by match with TREG5.

Figure 3.9 (3). 16-Bit Timer Mode Controller Register (T4MOD) (1/2)

T4MOD (0038H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT4 occurs at rise edge. 01: TI4↑ TI5↑ INT4 occurs at rise edge. 10: TI4↑ TI4↓ INT4 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC4 Clear Enable	Timer 4 source clock 00: TI4 01: φT1 10: φT4 11: φT16	
	Invert when the UC value is loaded to CAP2	Invert when the up-counter matches TREG5						

Capture timing of timer4

	Capture control	INT4 control
00	Capture disable	Interrupt occurs at the rise edge of TI4 (INT1) input.
01	CAP1 at TI4 rise CAP2 at TI5 rise	
10	CAP1 at TI4 rise CAP2 at TI4 fall	Interrupt occurs at the fall edge of TI4 (INT1) input.
11	CAP1 at TFF1 rise CAP2 at TFF1 fall	Interrupt occurs at the rise edge of TI4 (INT1) input.

Software capture

0	The up-counter4 value is loaded to CAP1 (software capture).
1	Always read as "1".

Timer flip-flop 5 (TFF5) invert trigger

0	Trigger disable (Invert Prohibition)
1	Trigger enable (Invert permission)

CAP2T5 : Invert when the up-counter value is loaded to CAP2
EQ5T5 : Invert when the up-counter matches TREG5

Figure 3.9 (4). 16-Bit Controller Register (T4MOD) (2/2)

T4FFCR (0039H)

	7	6	5	4	3	2	1	0
bit Symbol	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	0	0	0	0	0	0	0	0
Function	00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: don't care Always read as "11".		TFF4 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: don't care ※ Always read as "11"	
			Invert when the UC value is loaded to CAP2	Invert when the UC value is loaded to CAP1	Invert when the UC matches TREG5	Invert when the UC matches TREG4		

Timer flip-flop 4 (TFF4) control

00	Inverts the TFF4 value (software inversion).
01	Sets TFF4 to "1".
10	Clear TFF4 to "0".
11	Don't care (Always read as "11") .

Timer flip-flop 4 (TFF4) invert trigger

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP2T4 : Invert when the up-counter value is loaded to CAP2
CAP1T4 : Invert when the up-counter value is loaded to CAP1
EQ5T4 : Invert when up-counter matches TREG5
EQ4T4 : Invert when up-counter matches TREG4

Timer flip-flop 5 (TFF5) control

00	Inverts the TFF5 value (software inversion).
01	Set TFF5 to "1".
10	Clear TFF5 to "0".
11	Don't care (Always read as "11".)

Figure 3.9 (5). 16-Bit Timer 4 F/F Control (T4FFCR)

T5MOD (0048H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset			1	0	0	0	0	0
Function			0: Soft-Capture 1: don't care	Capture timing 00: Disable INT6 occurs at rise edge. 01: TI6↑ TI7↑ INT6 occurs at rise edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC5 Clear Enable	Timer 5 source clock 00: TI6 01: φT1 10: φT4 11: φT16	

Timer 5 input clock

00	External clock (TI4)
01	φT1
10	φT4
11	φT16

Clearing the up-counter UC5

0	Clear disable
1	Clear by match with TREG7

Figure 3.9 (6). 16-Bit Timer Mode Control Register (T5MOD) (1/2)

T5MOD (0048H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
Read/Write			W	R/W		R/W	R/W	
After reset			1	0	0	0	0	0
			0: Soft-Capture 1: don't care	Capture timing 00: Disable INT6 occurs at rise edge. 01: TI6↑ TI7↑ INT6 occurs at rise edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC5 Clear Enable	Timer 5 source clock 00: TI6 01: φT1 10: φT4 11: φT16	

Timer 5 Capture timing

	Capture control	INT6 Control
00	Capture disable	Interrupt occurs at the rise edge of TI6 (INT6) input.
01	CAP3 at TI6 rise CAP4 at TI7 rise	
10	CAP3 at TI6 rise CAP4 at TI6 fall	Interrupt occurs at the fall edge of TI6 (INT6) input.
11	CAP3 at TFF1 rise CAP4 at TFF1 fall	Interrupt occurs at the rise edge of TI6 (INT6) input.

Software capture

0	The up-counter 5 value is loaded to CAP3.
1	Always read as "1".

Figure 3.9 (7). 16-Bit Timer Control Register (T5MOD) (2/2)

T5FFCR (0049H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
Read/Write			R/W	R/W	R/W	R/W	W	
After reset			0	0	0	0	0	0
Function			TFF6 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: don't care	
			Invert when the UC value is loaded to CAP4	Invert when the UC value is loaded to CAP3	Invert when the UC matches TREG7	Invert when the UC matches TREG6	※ Always read as "11"	

Timer flip-flop 6 (TFF6) control

00	Inverts the TFF6 value (software inversion).
01	Sets TFF6 to "1".
10	Clear TFF6 to "0".
11	Don't care (Always read as "11") .

Timer flip-flop 6 (TFF6) invert trigger

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP4T6 : Invert when the up-counter value is loaded to CAP4
CAP3T6 : Invert when the up-counter value is loaded to CAP3
EQ7T6 : Invert when up-counter matches TREG7
EQ6T6 : Invert when up-counter matches TREG6

Figure 3.9 (8). 16-Bit Timer 5 F/F Control (T5FFCR)

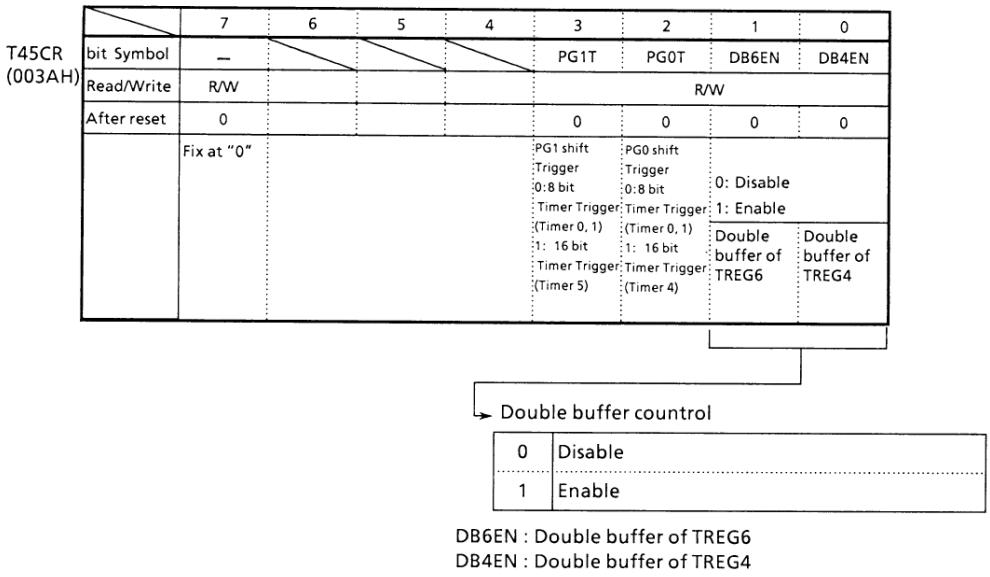


Figure 3.9 (9). 16-Bit Timer (Timer 4, 5) Control Register (T45CR)

TRUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler & Timer Run / Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							

Count Operation

0	Stop and clear
1	Count

PRRUN : Operation of prescaler
T5RUN : Operation of 16-bit timer (timer5)
T4RUN : Operation of 16-bit timer (timer4)
P1RUN : Operation of PWM timer (PWM1 / timer3)
P0RUN : Operation of PWM timer (PWM0 / timer2)
T1RUN : Operation of 8-bit timer (timer1)
T0RUN : Operation of 8-bit timer (timer0)

SYSCR1 (006EH)

	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	0	0	0	0	0	0
Function	High Frequency oscillator (fc) 0 : stop 1 : oscillaton	Low Frequency oscillator (fc) 0 : stop 1 : oscillaton	High Frequency oscillator (fc) after released STOP mode 0 : stop 1 : oscillaton	Low Frequency oscillator (fc) after released STOP mode 0 : stop 1 : oscillaton	select clock after released STOP mode 0 : fc 1 : fs	Warming Up Timer 0 write : don't care 1 write : start timer 0 read : end warming up 1 read : not end warming up	select gear value of high frequency (fs) 00 : f_{FPH} 01 : fs 10 : fc / 16 11 : (reserved)	

select gear value of high frequency

00	f_{FPH}
01	fs
10	fc / 16
11	(reserved)

1 / 4times clock

Figure 3.9 (10). Timer Operation Control Register/System Clock Control Register

① Prescaler

There are 9 bit prescaler and prescaler clock selection register to generate input clock for 8 bit Timer 0, 1, 16 bit Timer 4, 5 and Serial Inteface 0, 1.

Figure 3.9 (11) shows the block diagram. Table 3.7 (1) shows prescaler clock resolution to 8, 16 bit Timer.

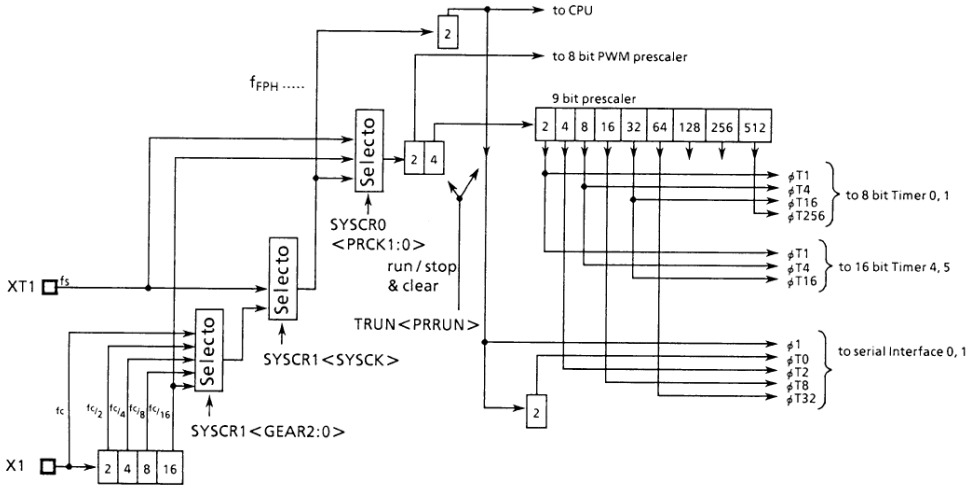


Figure 3.9 (11). Prescaler Block Diagram

Table 3.7 (1) Prescaler Clock Resolution to 8, 16 bit Timer

at fc = 16 MHz, fs = 32 kHz

Select system clock <SYSCK>	select prescaler clock <PRCK1, 0>	Gear value <GEAR2 : 0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^3$ (250 μs)	$fs/2^5$ (1 ms)	$fs/2^7$ (4 ms)	$fs/2^{11}$ (64 ms)
0 (fc)		000 (fc)	$fc/2^3$ (0.5 μs)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)	$fc/2^{11}$ (128 μs)
		001 ($fc/2$)	$fc/2^4$ (1 μs)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{12}$ (256 μs)
		010 ($fc/4$)	$fc/2^5$ (2 μs)	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{13}$ (512 μs)
		011 ($fc/8$)	$fc/2^6$ (4 μs)	$fc/2^8$ (16 μs)	$fc/2^{10}$ (64 μs)	$fc/2^{14}$(1.024 ms)
		100 ($fc/16$)	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{11}$ (128 μs)	$fc/2^{15}$(2.048 ms)
XXX	01 (low frequency clok)	XXX	$fs/2^3$ (250 μs)	$fs/2^5$ (1 ms)	$fs/2^7$ (4 ms)	$fs/2^{11}$ (64 ms)
XXX	10 ($fc/16$ clock)	XXX	$fc/2^7$ (8 μs)	$fc/2^9$ (32 μs)	$fc/2^{11}$ (128 μs)	$fc/2^{15}$(2.048 ms)

XXX : don't care

16 bit Timer

8 bit Timer

(Note) The fc/16 clock as a prescaler clock can not be used when the fs is used as a system clock.

The 1/4 times clock selected among 2 times system clock, fc/16 clock, and fs clock is input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1 : 0>.

Resetting sets <PRCK1 : 0> to "00", therefore, 2 times system clock is input.

The 16 bit Timer 4, 5 uses 3 types of clock: øT1, øT4, and øT16 among the prescaler outputs.

The prescaler can be run or stopped by the timer operation control register TRUN <PRRUN>. Counting starts when <PRRUN> is set to "1", while the prescaler is cleared to zero and stops operation when <PRRUN> is set to "0".

When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop this prescaler before "HALT" instruction is executed.

② Up-counter

UC4 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD <T4CLK1, 0> or T5MOD <T5CLK1, 0> register.

As the input clock, one of the internal clocks ϕT1, ϕT4, and ϕT16 from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P80/INT4 pin) can be selected. When reset, it will be initialized to <T4CLK1, 0> = 00 to select TI4 input mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN>.

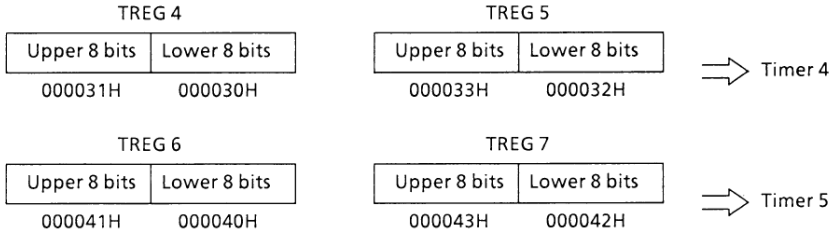
When clearing is enabled, up-counter UC4/UC5 will be cleared to zero each time it coincides matches the timer register TREG5, TREG7. The "clear enable/disable" is set by T4MOD <CLE>.

If clearing is disabled, the counter operates as a free-running counter.

③ Timer Registers

These two 16-bit registers are used to set the interval time. When the value of up-counter UC4 matches the set value of this timer register, the comparator match detect signal will be active.

Setting data for timer register (TREG4 and TREG5) is executed using 2 byte date transfer instruction or using 1 byte date transfer instruction twice for lower 8 bits and upper 1 bits in order.



TREG4 timer register is of double buffer structure, which is paired with register buffer. The timer control register T45CR <DB4EN> controls whether the double buffer structure should be enabled or disabled. : disabled when <DB4EN> = 0, while enabled when <DB4EN> = 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 (UC4) and timer register TREG5.

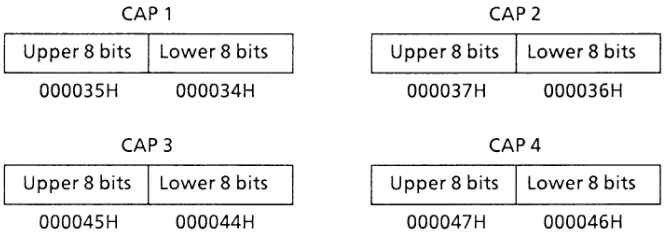
When reset, it will be initialized to <DB4EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN> = 1, and then write the following data in the register buffer.

TREG4, TREG6 and register buffer are allocated to the same memory addresses 000030H/000031H. When <DB4EN> = 0, same value will be written in both the timer register and register buffer. When <DB4EN> = 1, the value is written into only the register buffer.

④ Capture Register

These 16-bit registers are used to hold the values of the up-counter.

Data in the capture registers should be read by a 2-byte data load instruction or two 1-byte data load instruction, from the lower 8 bits followed by the upper 8 bits.



⑤ Capture Input Control

This circuit controls the timing to latch the value of up-counter UC4 into (CAP1, CAP2). The latch timing of capture register is controlled by register T4MOD <CAP12M1, 0>.

- When T4MOD <CAP12M1, 0> = 00

Capture function is disabled. Disable is the default on reset.

- When T4MOD <CAP12M1, 0> = 01

Data is loaded to CAP1 at the rise edge of TI4 pin (also used as P80/INT4) input, while data is loaded to CAP2 at the rise edge of TI5 pin (also used as P81/INT5) input. (Time difference measurement)

- When T4MOD <CAP12M1, 0> = 10

Data is loaded to CAP1 at the rise edge of TI4 pin input, while to CAP2 at the fall edge. Only in this setting, interrupt INT4 occurs at fall edge. (Pulse width measurement)

- When T4MOD <CAP12M1, 0> = 11

Data is loaded to CAP1 at the rise edge of timer flip-flop TFF1, while to CAP2 at the fall edge.

Besides, the value of up-counter can be loaded to capture registers by software. Whenever “0” is written in T4MOD <CAPIN>, the current value of up-counter will be loaded to capture register CAP1. It is necessary to keep the prescaler in RUN mode (TRUN <PRRUN> to be “1”).

⑥ Comparator

These are 16-bit comparators which compare the up-counter UC4 value with the set value of (TREG4, TREG5) to detect the match. When a match is detected, the comparators generate an interrupt (INTT4, INTT5), respectively. The up-counter UC4 is cleared only when UC4 matches TREG5. (The clearing of up-counter UC4 can be disabled by setting T4MOD <CLE> = 0.)

⑦ Timer Flip-Flop (TFF4)

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 T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4>. TFF4 will be inverted when “00” is written in T4FFCR <TFF4C1, 0>. Also it is set to “1” when “10” is written, and cleared to “0” when “10” is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P82).

⑧ Timer Flip-Flop (TFF5)

This flip-flop is inverted by the match detect signal from the comparator and the latch signal to the capture register CAP2. TFF5 will be inverted when “00” is written in T4FFCR <TFF5C1, 0>. Also it is set to “1” when “10” is written, and cleared to “0” when “10” is written. The value of TFF5 can be output to the timer output pin TO5 (also used as P82).

Note: This flip-flop (TFF5) is contained only in the 16-bit timer 4.

(1) 16-bit Timer Mode

Generating interrupts at fixed intervals:

In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTTR5.

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - 0 - - - -     Stop timer 4.
INTET54 ← 1 1 0 0 1 0 0 0     Enable INTTR5 and sets interrupt level 4. Disable
                              INTTR4.
T4FFCR  ← 1 1 0 0 0 0 1 1     Disable trigger.
T4MOD   ← 0 0 1 0 0 1 * *     Select internal clock for input and
            (** = 01, 10, 11) disable the capture function.
TREG5   ← * * * * * * * *     Set the interval time (16 bits).
          * * * * * * * *
TRUN    ← 1 X - 1 - - - -     Start timer 4.
```

Note : 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 (TI4 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.

The counter counts at the rise edge of TI4 pin input.

TI4 pin can also be used as P80/INT4.

Since both timers operate in exactly the same way, timer 4 is used for the purposes of explanation.

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - 0 - - - -     Stop timer 4.
P8CR    ← - - - - - - - 0     Set P80 to input mode
INTET54 ← 1 1 0 0 1 0 0 0     Enable INTTR5 and sets interrupt level 4, while
                              disables INTTR4.
T4FFCR  ← 1 1 0 0 0 0 1 1     Disable trigger.
T4MOD   ← 0 0 1 0 0 1 0 0     Select TI4 as the input clock.
TREG5   ← * * * * * * * *     Set the number of counts (16 bits).
TRUN    ← 1 X - 1 - - - -     Start timer 4.
```

Note : When used as an event counter, set the prescaler in RUN mode.

(3) 16-bit Programmable Pulse Generation (PPG) Output Mode

Since both timers operate in exactly the same way, timer 4 is used for the purposes of explanation.

The PPG mode is obtained by inversion of the timer flip-flop TFF4 that is to be enabled by the match of the up-counter UC4 with the timer register TREG4 or 5 and to be output to TO4 (also used as P82). In this mode, the following conditions must be satisfied.

(Set value of TREG4) < (Set value of TREG5)

	7 6 5 4 3 2 1 0	
TRUN	← - X - 0 - - - -	Stop timer 4.
TREG4	← * * * * * * * *	Set the duty. (16-Bit)
TREG5	← * * * * * * * *	Set the cycle. (16-Bit)
T45CR	← 0 X X X - - - 1	Double Buffer of TREG4 enable (Change the duty and cycle at the interrupt INTTR5)
T4FFCR	← 1 1 0 0 1 1 0 0	Set the mode to invert TFF4 at the match with TREG4 / TREG5, and also set the TFF4 to "0".
T4MOD	← 0 0 1 0 0 1 * *	Select the internal clock for the input, and disable
	(** = 01, 10, 11)	the capture function.
P8CR	← - - - - - - 1 - -	} Assign P82 as TO4.
P8FC	← X - X X - 1 X X	
TRUN	← 1 X - 1 - - - -	Start timer 4.

Note : X ; don't care - ; no change

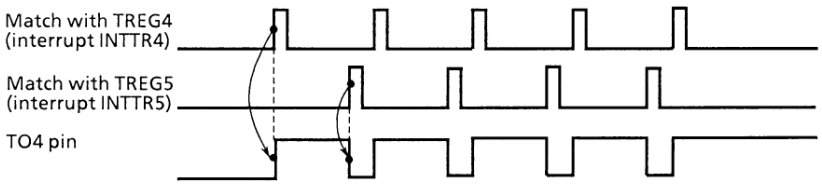


Figure 3.9 (11). Programmable Pulse Generation (PPG) Output Waveforms

When the double buffer of TREG4 is enabled in this mode, the value of register buffer 4 will be shifted in TREG4 at match with TREG5. This feature makes easy the handling of low duty waves.

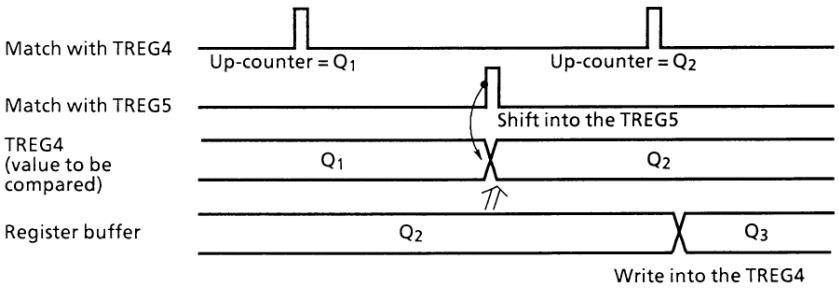


Figure 3.9 (12). Operation of Register Buffer

Shows the block diagram of this mode.

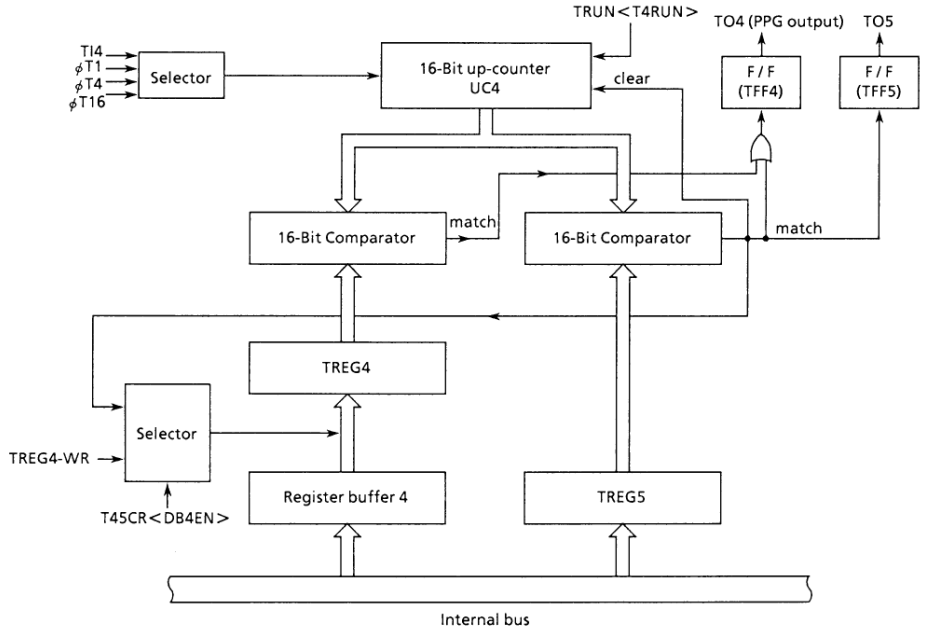


Figure 3.9 (13). Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

The loading of up-counter (UC4) values into the capture registers CAP1 and CAP2, the timer flip-flop TFF4 inversion due to the match detection by comparators CP4 and CP5, and the output of TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt 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 UC4 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up-counter into capture register CAP1 at the rise edge of the TI4 pin. Then set to T4MOD <CAP12M1, 0> = 01.

When the interrupt INT4 is generated at the rise edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 (= c + d), and set the above set value (c + d) plus a one-shot pulse width (p) to TREG5 (= c + d + p). When the interrupt INT4 occurs the T4FFCR <EQ5T4, EQ4T4> register should be set that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or TREG5. When interrupt INTTR5 occurs, this inversion will be disabled.

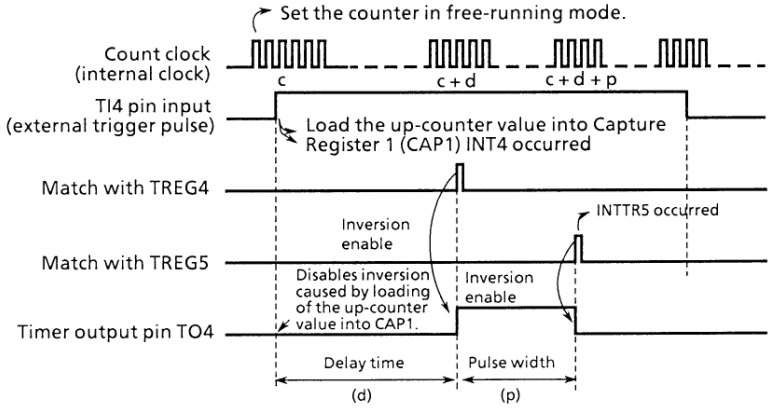


Figure 3.9 (14). One-Shot Pulse Output (with Delay)

Setting Example: To output 2ms one-shot pulse with 3ms delay to the external trigger pulse to TI4 pin.

※ Clock Condition

- system clock : High frequency (fc)
- clock gear : 1 (fc)
- prescaler clock : system clock (fsys)

Main setting

```
T4MOD  ←  - - 1 0 1 0 0 1
T4FFCR ←  1 1 0 0 0 0 1 0
P8CR   ←  - - - - - - 1 -
P8FC   ←  X - X X - 1 X X
INTE45 ←  - - - - 1 1 0 0
INTET54←  1 0 0 0 1 0 0 0
TRUN   ←  1 X - 1 - - - -
```

- T4MOD: Keep counting (Free-runnig); Count with φT1.
- T4FFCR: Load the up-counter value into CAP1 at the rise edge of TI4 pin input; Clear TFF4 to zero; Disable TFF4 inversion.
- P8CR / P8FC: Select P82 as the TO4 pin.
- INTE45: Enable INT4, and disable INTTR4 and INTTR5.
- INTET54: (no change)
- TRUN: Start timer 4.

Setting of INT4

```
TREG4  ←  CAP1+3ms/φT1
TREG5  ←  TREG4+2ms/φT1
T4FFCR ←  - - - - 1 1 - -
INTET54←  1 1 0 0 - - - -
```

- T4FFCR: Enable TFF4 inversion when the up-counter value matches TREG4 or 5.
- INTET54: Enable INTTR5.

Setting of INT5

```
T4FFCR ←  - - - - 0 0 - -
INTET54←  1 0 0 0 - - - -
```

- T4FFCR: Disable TFF4 inversion when the up-counter value matches TREG4 or 5.
- INTET54: Disable INTTR5.

Note: X ; don't care - ; no change

When delay time is unnecessary, invert timer flip-flop TFF4 when the up-counter value is loaded into capture register 1 (CAP1), and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt INT4 occurs. The TFF4 inversion should be enabled when the up-counter (UC4) value matches TREG5, and disabled when generating the interrupt INTTR5.

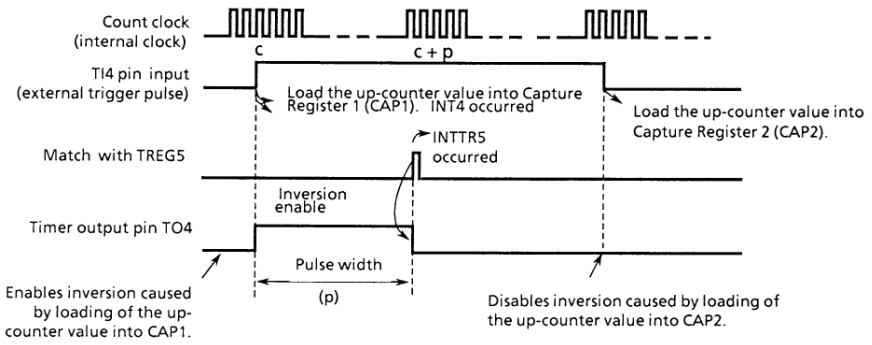


Figure 3.9 (15). One-Shot Pulse Output (without Delay)

② Frequency Measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TI4 pin, and its frequency is measured by the 8-bit timers (Timer 0 and Timer 1) and the 16-bit timer/event counter (Timer 4).

The TI4 pin input should be selected for the input clock of Timer 4. The value of the up-counter is loaded into the capture register CAP1 at the rise edge of the timer flip-flop TFF1 of 8-bit timers (Timer 0 and Timer 1), and into CAP2 at its fall edge.

The frequency is calculated by the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.

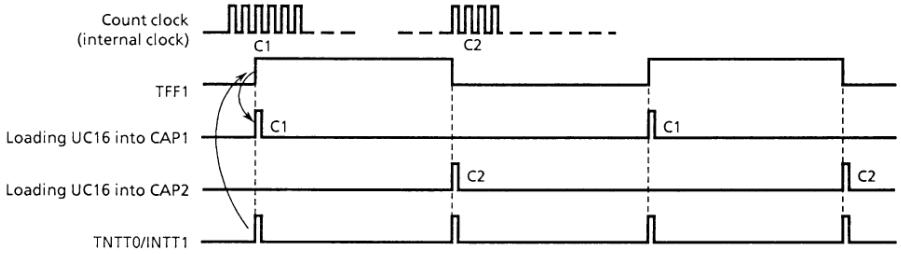


Figure 3.9 (16). Frequency Measurement

For example, if the value for the level “1” width of TFF1 of the 8-bit timer is set to 0.5 sec. and the difference between CAP1 and CAP2 is 100, the frequency will be 100/0.5 [sec.] = 200 [Hz].

③ Pulse Width Measurement

This mode allows measuring 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 TI4 pin. Then the capture function is used to load the UC4 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT4 occurs at the falling edge of TI4.

The pulse width is obtained from the difference between the values of CAP1 and CAP2 and the internal clock cycle.

For example, if the internal clock is 0.8 microseconds and the difference between CAP1 and CAP2 is 100, the pulse width will be 100 x 0.8μs = 80μs.

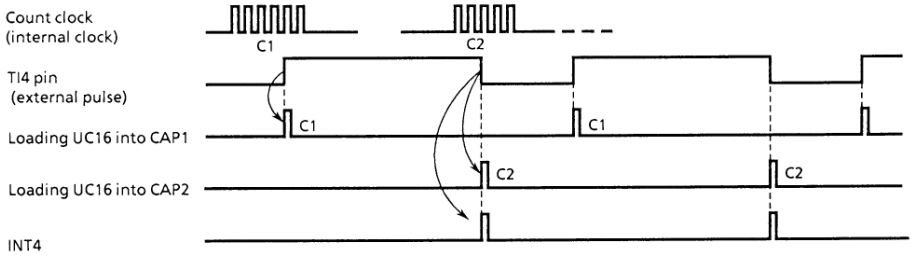


Figure 3.9 (17). Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD <CAP12M1, 0> = 10), external interrupt INT4 occurs at the falling edge of TI4 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 C2 and the second C1 at the second INT4 interrupt.

④ Time Difference Measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through TI4 and TI5.

Keep the 16-bit timer/event counter (Timer 4) counting (free-running) with the internal clock, and load the UC4 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT4 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT5.

The time difference between these pulses can be obtained from the difference between the time counts at which loading the up-counter value into CAP1 and CAP2 has been done.

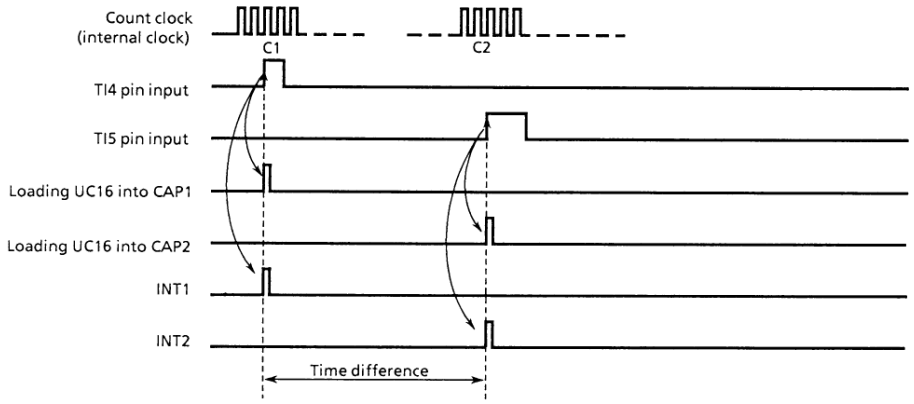


Figure 3.9 (18). Time Difference Measurement

(5) Different Phased Pulses Output Mode (This mode can only be used in Timer 4)

In this mode, signals with any different phase can be output by free-running up-counter UC4.

When the value in up-counter UC4 and the value in TREG4 (TREG5) match, the value in TFF4 (TFF5) is inverted and output to TO4 (TO5).

This mode can only be used by 16-bit timer 4.

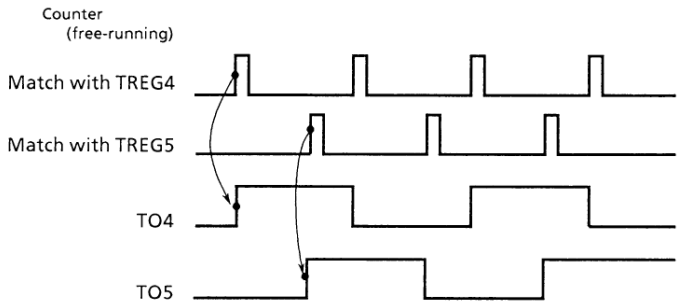


Figure 3.9 (19). Phase Output

Cycles (counter overflow time) of the above output waves are listed below.

Table 3.9 (2) Timer Output Cycle on the Different Phased Pulse Output Mode

at fc = 16 MHz, fs = 32kHz

select system clock <SYSCK>	select prescaler clock <PRCK1, 0>	Gear value <GEAR2 : 0>	Counter Overflow Time		
			ϕT1	ϕT4	ϕT16
1 (fs)	00 (f_{FPH})	XXX	16.384 s	65.536 s	262.144 s
0 (fc)		000 (fc)	32.768 ms	131.072 ms	524.288 ms
		001 ($fc/2$)	65.536 ms	262.144 ms	1.049 s
		010 ($fc/4$)	131.072 ms	524.288 ms	2.097 s
		011 ($fc/8$)	262.144 ms	1.049 s	4.194 s
		100 ($fc/16$)	524.288 ms	2.097 s	8.389 s
XXX	01 (low frrqency clock)	XXX	16.384 s	65.536 s	262.144 s
XXX	10 ($fc/16$ clock)	XXX	524.288 ms	2.097 s	8.389 s

xxx : don't care

3.10 Stepping Motor Control/Pattern Generation Port

TMP93CM40/M41 has two channels (PG0 and PG1) of 4-bit hardware stepping motor control/pattern generation (herein after called PG) which actuate in synchronization with the (8-bit/16-bit) timers. The PG (PG0 and PG1) are shared in 8-bit I/O ports P6.

Channel 0 (PG0) is synchronous with 8-bit timer 0 or timer 1, 16-bit timer 5, to update the output.

The PG ports are controlled by control registers (PG01CR) and can select either stepping motor control mode or pattern generation mode. Each bit of the P6 can be used as the PG port.

PG0 and PG1 can be used independently.

All PG operate in the same manner except the following points, and thus only the operation of PG0 will be explained below.

Different Points Between PG0 and PG1

	PG0	PG1
Trigger Signal	from Timer 4	from Timer 5

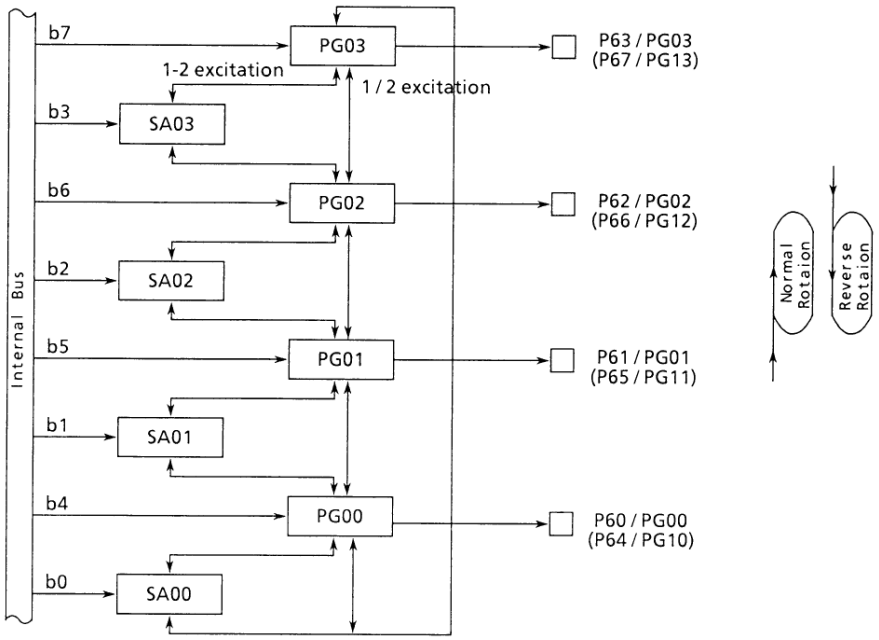


Figure 3.10 (1). PG Block Diagram

PG01CR (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R/W				R/W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8bit write 1: 4bit write	PG1 Rotaing direction 0: Normal rotation 1: Reverse rotation	PG1 mode (excitation) 0: 1excitation or 2excitation 1: 1-2 excitation	PG1 trigger input enable 0: disable 1: enable	PG0 write mode 0: 8bit write 1: 4bit write	PG0 Rotaing direction 0: Normal rotation 1: Reverse rotation	PG0 mode (excitation) 0: 1excitation or 2excitation 1: 1-2 excitation	PG0 trigger input enable 0: disable 1: enable

PG0 Trigger input enable

0	Trigger input disable to PG0
1	Trigger input enable to PG0

Set the operation mode of PG0

0	1 or 2 excitation (full step)
1	1-2 excitation (half step) / PG mode

PG0 (stepping motor control)
Rotaing direction control

0	Normal rotation / PG mode
1	Reverse rotation

Selecting PG0 write mode

0	8-bit write
1	4-bit write / PG mode (Only shifter alternate register can be written.)

Figure 3.10 (2a). Pattern Generation Control Register (PG01CR)

PG01CR (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R / W				R / W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8bit write 1: 4bit write	PG1 Rotaing direction 0: Normal rotation 1: Reverse rotation	PG1 mode (excitation) 0: 1excitation or 2excitation 1: 1-2 excitation	PG1 trigger input enable 0: disable 1: enable	PG0 write mode 0: 8bit write 1: 4bit write	PG0 Rotaing direction 0: Normal rotation 1: Reverse rotation	PG0 mode (excitation) 0: 1excitation or 2excitation 1: 1-2 excitation	PG0 trigger input enable 0: disable 1: enable

PG1 trigger input enable

0	Trigger input disable to PG1
1	Trigger input enable to PG1

Set the operation mode of PG1

0	1 or 2 excition (full step)
1	1-2 excitation (half step) / PG mode

PG1 (stepping motor control)
Rotaing direction control

0	Normal rotation / PG mode
1	Reverse rotation

Selecting PG1 write mode

0	8-bit write
1	4-bit write / PG mode (Only shifter alternate register can be written.)

Figure 3.10 (2b). Pattern Generation Control Register (PG01CR)

PG0REG (004CH)

	7	6	5	4	3	2	1	0
bit Symbol	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
Read/Write	W				R/W			
After reset	0	0	0	0	Undefined			
Function	Pattern Generation 0 (PG0) output latch register (Reading the P6 that is set to the PG port allows to read-out.)				Shift alternate register 0 For the PG mode (4-bit write) register			

Prohibit Read modify write

Figure 3.10 (3). Pattern Generation 0 Register (PG0REG)

PG1REG (004DH)

	7	6	5	4	3	2	1	0
bit Symbol	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
Read/Write	W				R/W			
After reset	0	0	0	0	Undefined			
Function	Pattern Generation 1 (PG1) output latch register (Reading the P6 that is set to the PG port allows to read-out.)				Shift alternate register 1 For the PG mode (4-bit write) register			

Prohibit Read modify write

Figure 3.10 (4). Pattern Generation 1 Register (PG1REG)

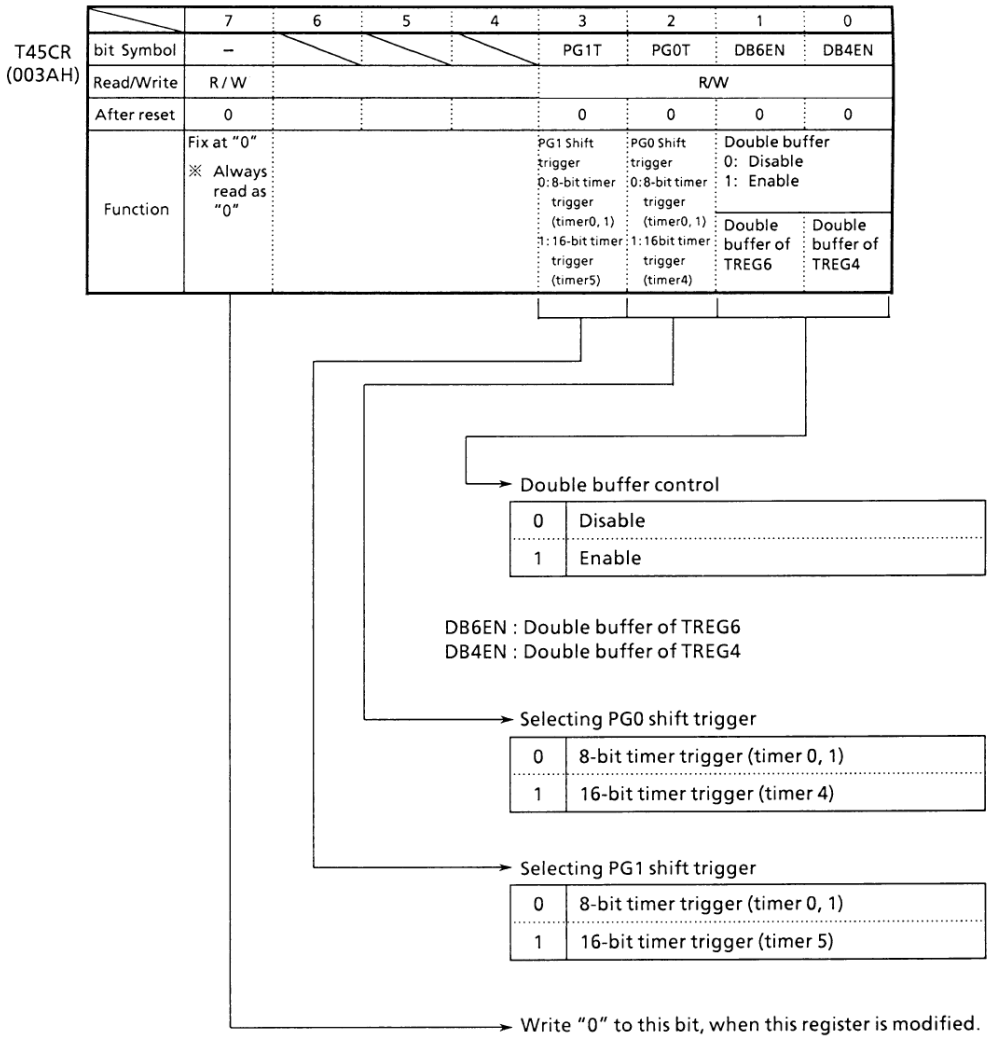


Figure 3.10 (5). 16-bit Timer Trigger Control Register (T45CR)

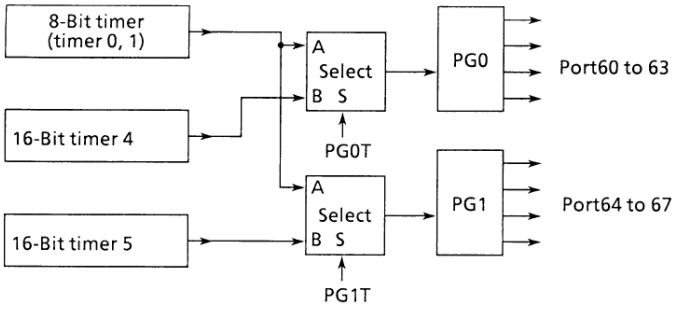


Figure 3.10 (6). Connection of Timer and Pattern Generator

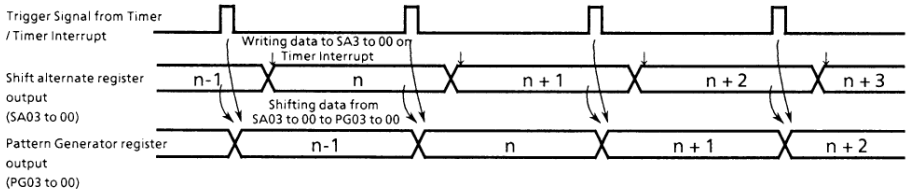
(1) Pattern Generation Mode

PG functions as a pattern generation according to the setting of PG01CR <PAT1>. In this mode, writing from CPU is executed only on the shifter alternate register. Writing a new data should be done during the interrupt operation of the timer for shift trigger, and a pattern can be output synchronous with the timer.

In this mode, set PG01CR <PG0M> to 1, and PG01CR <CCW0> to 0.

The output of this pattern generator is output to port 6; since port and functions can be switched on a bit basis using port function control register P6FC, any port pin can be assigned to pattern generator output.

Figure 3.10 (7) shows the block diagram of this mode.



Example of pattern generation mode

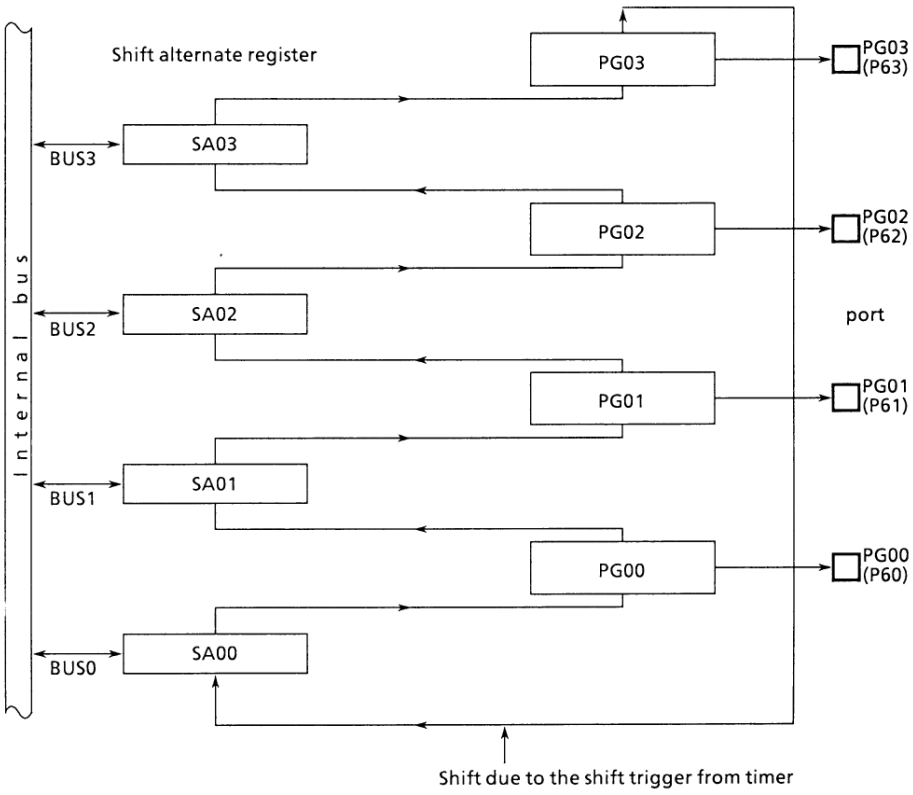


Figure 3.10 (7). Pattern Generation Mode Block Diagram (PG0)

In this pattern generation mode, only writing the output latch is disabled by hardware, but other functions do the same operation as 1-2 excitation in stepping motor control port mode. Accordingly, the data shifted by trigger signal from a timer must be written before the next trigger signal is output.

(2) Stepping Motor Control Mode

① 4-phase 1-Step/2-Step Excitation

Figure 3.10 (8) and Figure 3.10 (9) show the output waveforms of 4-phase 1 excitation and 4-phase 2 excitation, respectively when channel 0 (PG0) is selected.

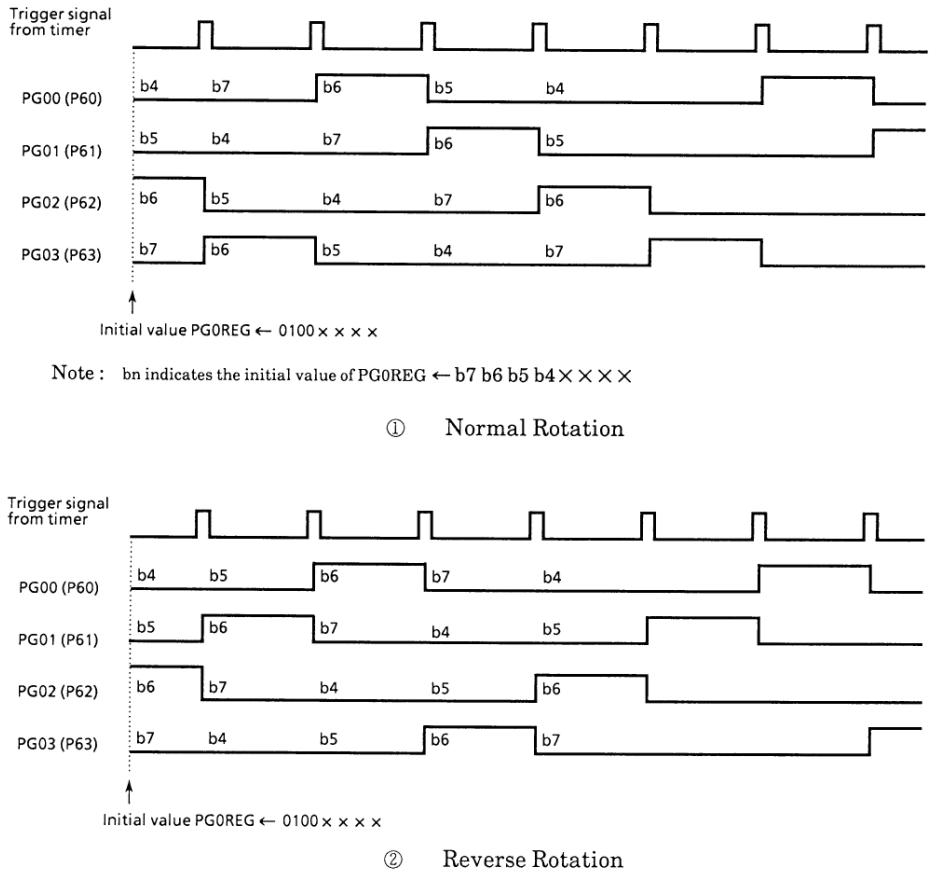


Figure 3.10 (8). Output Waveforms of 4-Phase 1-Step Excitation (Normal Rotation and Reverse Rotation)

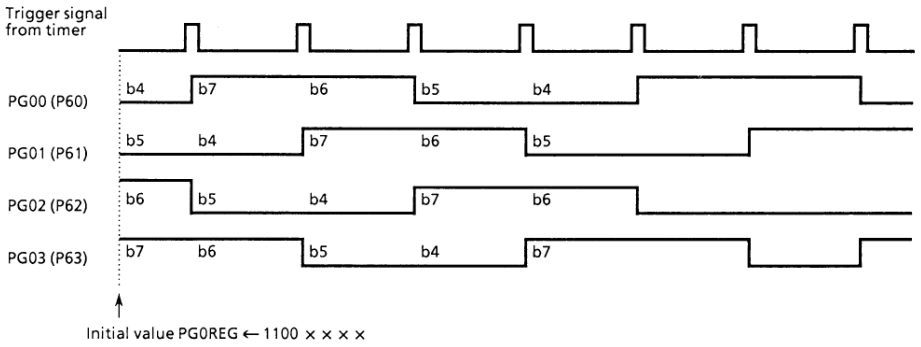


Figure 3.10 (9). Output Waveforms of 4-Phase 2-Step Excitation (Normal Rotation)

The operation when channel 0 is selected is explained below.

The output latch of PG0 (also used as P6) is shifted at the rising edge of the trigger signal from the timer to be output to the port.

The direction of shift is specified by PG01CR <CCW0>: Normal rotation (PG00 → PG01 → PG02 → PG03) when <CCW0> is set to "0"; reverse rotation (PG00 ← PG01 ← PG02 ← PG03) when "1". Four-phase 1-step excitation will be selected when only one bit is set to "1" during the initialization of PG, while 4-phase 2-step excitation will be selected when two consecutive bits are set to "1".

The value in the shift alternate registers are ignored when the 4-phase 1-step/2-step excitation mode is selected.

Figure 3.10 (10) shows the block diagram.

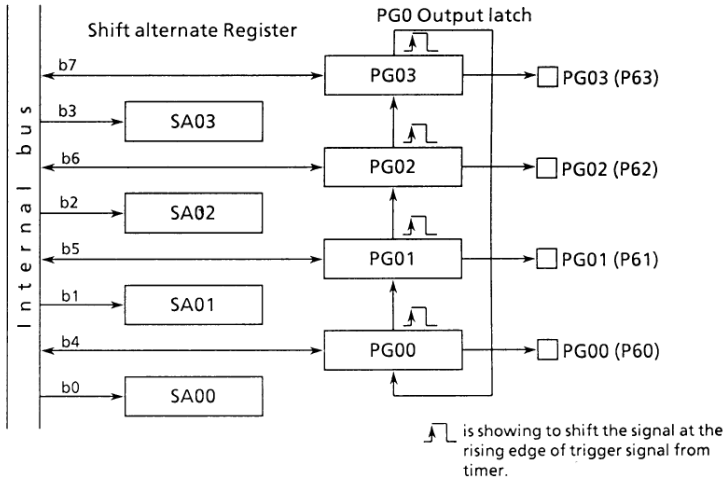


Figure 3.10 (10). Block Diagram of 4-Phase 1-Step Excitation/2-Step Excitation (Normal Rotation)

② 4-Phase 1-2 Step Excitation

Figure 3.10 (11) shows the output waveforms of 4-phase 1 -2 step excitation when channel 0 is selected.

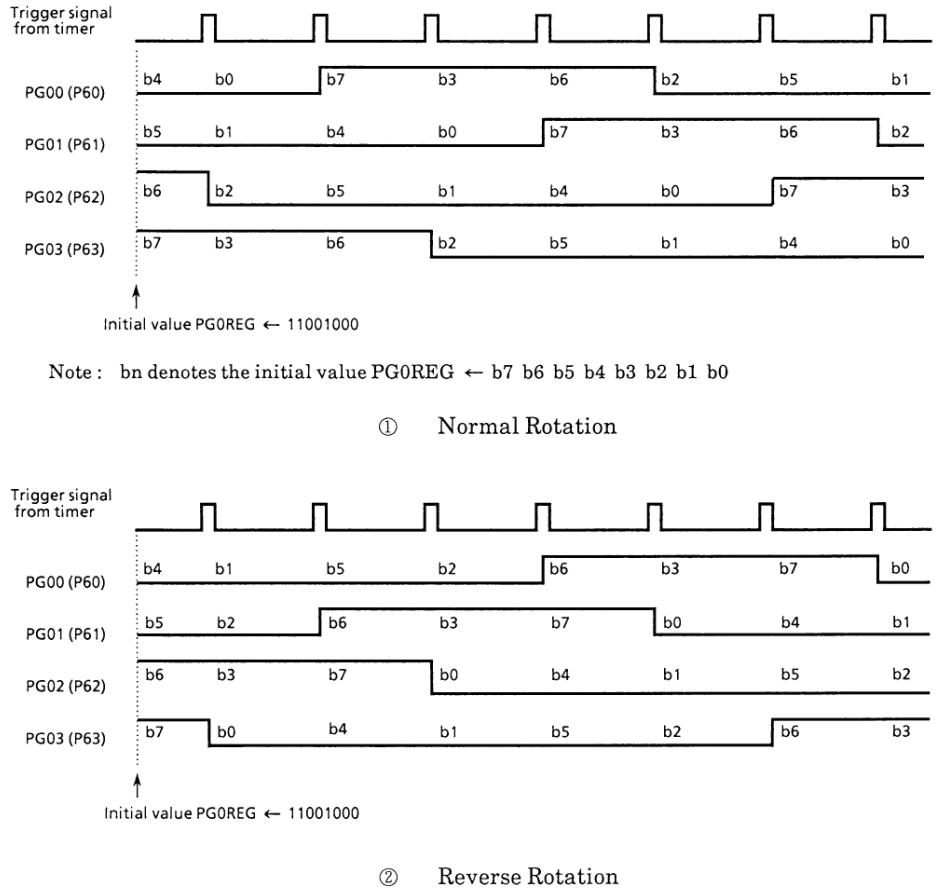


Figure 3.10 (11). Output Waveforms of 4-Phase 1-2 Step Excitation (Normal Rotation and Reverse Rotation)

The initialization for 4-phase 1-2 step excitation is as follows:

By rearranging the initial value “b7 b6 b5 b4 b3 b2 b1 b0” to “b7 b3 b6 b2 b5 b1 b4 b0”, the consecutive 3 bits are set to “1” and other bits are set to “0” (positive logic).

For example, if b7, b3, and b6 are set to “1", the initial value becomes “11001000”, obtaining the output waveforms as shown in Figure 3.10 (11).

To get an output waveform of negative logic, set values 1s and 0’s of the initial value should be inverted. For example, to change the output waveform shown in Figure 3.10 (11) into negative logic, change the initial value to “00110111”.

The operation will be explained below for channel 0.

The output latch of PG0 (shared by P6) and the shifter alternate register (SA0) for Pattern Generation are shifted at the rising edge of trigger signal from the timer to be output to the port. The direction of shift is set by PG01CR <CCW0>.

Figure 3.10 (12) shows the block diagram.

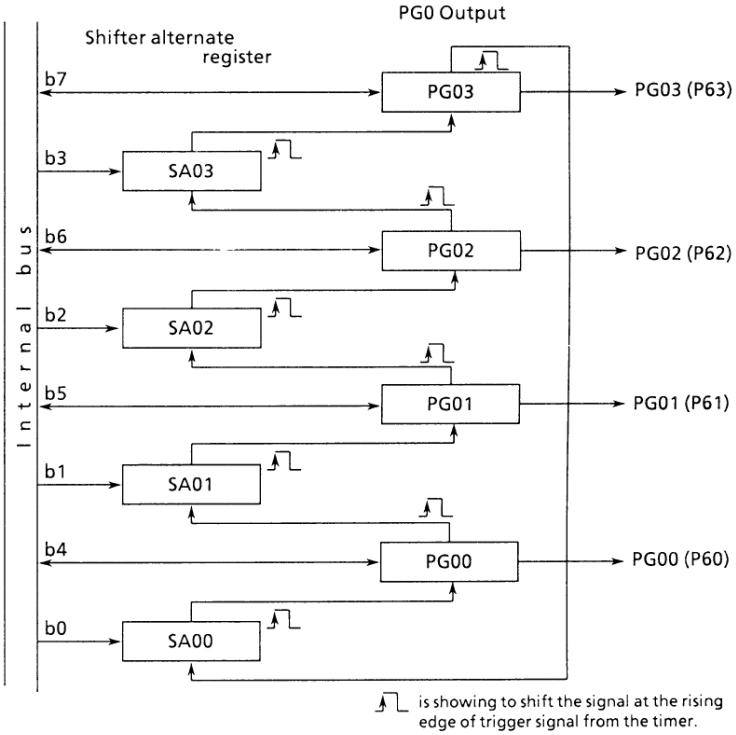


Figure 3.10 (12). Block Diagram of 4-Phase 1-2 Step Excitation (Normal Rotation)

Setting example: To drive channel 0 (PG0) by 4-phase 1-2 step excitation (normal rotation) when timer 0 is selected, set each register as follows:

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - - - - - 0     Stop timer 0, and clear it to zero.
TMOD    ← 0 0 X X - - 0 1     Set 8-bit timer mode and select φT1 as the input clock of timer 0.
TFFCR   ← X X X 0 1 0 1 0     Clear TFF1 to zero and enable the inversion trigger by timer 0.
TREG0   ← * * * * * * * *     Set the cycle in timer register.
P6CR    ← - - - - - 1 1 1 1   Set P60 to P63 bits to the output mode.
P6FC    ← - - - - - 1 1 1 1   Set P60 to P63 bits to the PG output.
PG01CR  ← - - - - - 0 0 1 1   Select PG0 4-phase 1-2 step excitation mode and normal rotation .
PG0REG  ← 1 1 0 0 1 0 0 0     Set an initial value.
TRUN    ← 1 X - - - - - 1     Start timer 0.
```

Note : X ; don't care - ; no change

(3) Trigger Signal From Timer

The trigger signal from the timer which is used by PG is not equal to the trigger signal of timer flip-flop (TFF1, TFF4, TFF5, and TFF6) and differs as shown in Table 3.10 (1) depending on the operation mode of the timer.

Table 3.10 (1) Select of Trigger Signal

	TFF1 Inversion	PG Shift
8-bit timer mode	Selected by TFFCR <TFF1IS> when the up-counter value matches TREG0 or TREG1 value.	←
16-bit timer mode	When the up-counter value matches with both TREG0 and TREG1 values. (The value of up-counter = TREG1*2^8 + TREG0)	←
PPG output mode	When the up-counter value matches with both TREG0 and TREG1.	When the up-counter value matches TREG1 value (PPG cycle).
PWM output mode	When the up-counter value matches TREG0 value and PWM cycle.	Trigger signal for PG is not generated.

Note: To shift PG, TFFCR <TFF1IE> must be set to "1" to enable TFF1 inversion.

Channel 1 of PG can be synchronized with the 16-bit timer Timer 4/Timer 5. In this case, the PG shift trigger signal from the 16-bit timer is output only when the up-counter UC4/UC5 value matches TREG5/TREG7.

When using a trigger signal from Timer 4, set either T4FFCR <EQ5T4> or T4MOD <EQ5T5> to "1" and a trigger is generated when the value in UC4 and the value in TREG5 match. When using a trigger signal from Timer 5, set T5FFCR <EQ7T6> to 1. Generates a trigger when the value in UC5 and the value in TREG7 match.

(4) Application of PG and Timer Output

As explained in “Trigger signal from timer”, the timing to shift PG and invert TFF differs depending on the mode of timer. An application to operate PG while operating an 8-bit timer in PPG mode will be explained below.

To drive a stepping motor, in addition to the value of each phase (PG output), synchronizing signal is often required at the timing when excitation is changed over. In this application, port 6 is used as a stepping motor control port to output a synchronizing signal to the TO1 pin (shared by P71).

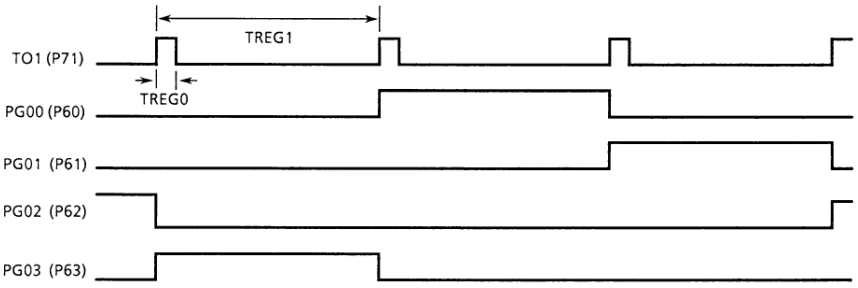


Figure 3.10 (13). Output Waveforms of 4-Phase 1-Step Excitation

Setting example:

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - - - - 0 0   Stop timer 0, and clear it to zero.
TMOD    ← 1 0 X X X X 0 1   Set timer 0 and timer 1 in PPG output mode and select
                            φT1 as the input clock.
TFFCR   ← X X X 0 0 1 1 X   Enable TFF1 inversion and set TFF1 to "1".
TREG0   ← * * * * * * * *   Set the duty of TO1 to TREG0.
TREG1   ← * * * * * * * *   Set the cycle of TO1 to TREG1.
P7CR    ← X X X X - - 1 -
P7FC    ← X X X X - - 1 X   } Assign P71 as TO1.
P6CR    ← - - - - 1 1 1 1
P6FC    ← - - - - 1 1 1 1   } Assign P60 to 63 as PG0.
PG01CR  ← - - - - 0 0 0 1   Set PG0 in 4-phase 1-step excitation mode.
PG0REG  ← * * * * * * * *   Set an initial value.
TRUN    ← 1 X - - - - 1 1   Start timer 0 and timer 1.
```

Note: X; don't care - ; no change

3.11 Serial Channel

TMP93CM40/TMP93CM41 contains two serial I/O channels for full duplex asynchronous transmission (UART) as well as for I/O extension.

The serial channel has the following operation modes:

- I/O interface mode (channel 1 only)
  - Mode 0: To transmit and receive I/O data as well as the synchronizing signal SCLK for extending I/O.
- Asynchronous transmission (UART) mode (channel 0 and 1)
  - 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 wake-up function for making the master controller start slave controllers in serial link (multi-controller system).

Figure 3.11 (1) shows the data format (for one frame) in each mode.

Serial Channel 0 and 1 can be used independently.

All channels operate in the same manner except the following points, thus, only the channel 0 will be explained below.

Different Points Between Channel 0 and Channel 1

	Channel 0	Channel 1
Pin Name	TXD0 (P90), RXD0 (P91), $\overline{CTS0}$/SCLK0 (P92)	TXD01 (P93), RXD1 (P94), SCLK1 (P95)
Handshake Function	Exist	Does Not Exist (Not for $\overline{CTS}$ pin)

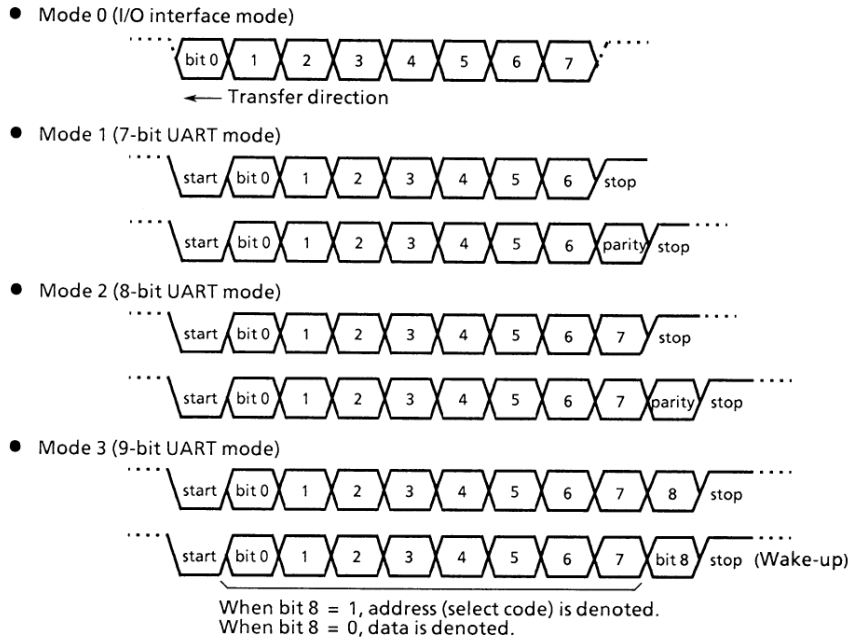


Figure 3.11 (1). Data Formats

The serial channel has a buffer register for transmitting and receiving operations, in order to temporarily store transmitted or received data, so that transmitting and receiving operations can be done independently (full duplex).

However, in I/O interface mode, SCLK (serial clock) pin is used for both transmission and receiving, the channel becomes half-duplex.

The receiving data register is of a double buffer structure to prevent the occurrence of overrun error and provides one frame of margin before CPU reads the received data. The receiving data register stores the already received data while the buffer register receives the next frame data.

By using CTS and RTS (there is no RTS pin, so any one port must be controlled by software), it is possible to halt data send until CPU finishes reading receive data every time a frame is received (Handshake function).

In the UART mode, a check function is added not to start the receiving operation by error start bits due to noise. The channel starts receiving data only when the start bit is detected to be normal at least twice in three samplings.

When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during receiving operation, flag SC0CR/SC1CR <OERR, PERR, FERR> will be set.

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of four clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the internal prescaler (shared by 8-bit/16-bit timer) by the value 2 to 16.

In I/O interface mode, it is possible to input synchronous signals as well as to transmit or receive data by external clock.

3.11.1 Control Registers

The serial channel is controlled by three control registers SC0CR, SC0MOD, and BR0CR. Transmitted and received data is stored in register SC0BUF.

SC0MOD (0052H)

	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 Bit 8	Hand shake 0: CTS disable 1: CTS enable	Receiving Function 0: Receive disable 1: Receive enable	Wake up 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: TO0 Trigger 01: baud rate generator 10: Internal clock Φ1 11: don't care	

Serial transmission clock (UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	don't care

Serial transmission mode

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

Wake-up Function

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

Receiving Function

0	Receive Disable
1	Receive Enable

Hand shake function ($\overline{CTS}$ Pin) enable

0	Disable (always Transferable)
1	Enable

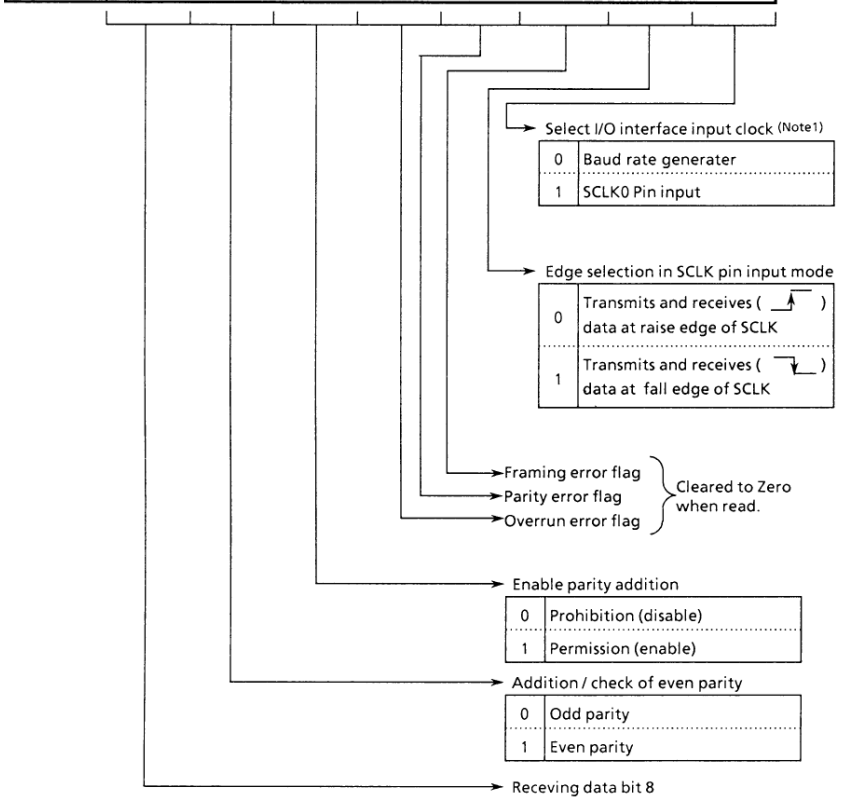
Transmission data bit 8

Note : There is SC1MOD (56H) in Channel1

Figure 3.11 (2). Serial Mode Control Register (Channel 0, SC0MOD)

SC0CR (0051H)

	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 Zero when read)			R/W	
After reset		0	0	0	0	0	0	0
Function	Received data Bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error Overrun	Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud Rate Generator 1: SCLK0 Pin input



Note : Serial control register for channel 1 is SC1CR (55H).
Note : As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.11 (3). Serial Control Register (Channel, SC0CR)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	—		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

0000	16 divisions
0001	Don't set
0010 to 1111	2 to 15 divisions

Selecting the input clock of baud rate generator

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

Note : Set TRUN<PRRUN> to "1" when the baud rate generator is used.

Figure 3.11 (4). Serial Channel Control (Channel 0, BR0CR)

SC0BUF (50H)

	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)

Figure 3.11 (5). Serial Transmission/Receiving Buffer Registers (Channel 0, SC0BUF)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	—	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfered data Bit 8	Fix at "0"	Receiving Function 0: Receive disable 1: Receive Enable	Wake up 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: TO0 Trigger 01: Baud rate generator 10: Internal clock Φ1 11: don't care	

Serial transmission clock (For UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	don't care

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC1CR).

Serial transmission mode

00	I/O interface mode	
01	UART mode	7-Bit length
10		8-Bit length
11		9-Bit length

Wake up Function

	9-Bit UART	Other mode
0	Interrupt when data are received	don't care
1	Interrupt only when RB8 = 1	

Receiving control

0	Receive disable
1	Receive enable

Transmission data bit 8

Figure 3.11 (6). Serial Mode Control Register (Channel 1, SC1MOD)

SC1CR (0055H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (clear to Zero when read)			R/W	
After reset		0	0	0	0	0	0	0
Function	Received data Bit 8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error / Overrun	1: error / Parity	1: error / Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	0: Baud rate generator 1: SCLK1 Pin input

Select I/O interface input clock (Note1)

0	Baud rate generate
1	SCLK1 Pin input

Edge selection in SCLK pin input mode

0	Transmits and receives (↑) data at raise edge of SCLK
1	Transmits and receives (↓) data at fall edge of SCLK

Framing error flag
Parity error flag
Overrun error flag
→ Cleared to Zero when read

Enable parity addition

0	Disable
1	Enable

Addition / check of even parity

0	Odd Parity
1	Even Parity

Receiving data bit 8

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

Figure 3.11 (7). Serial Control Register (Channel 1, SC1CR)

BR1CR (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	—		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

0000	16 divided
0001	Don't set
0010 to 1111	2 to 5 divisions

Selecting the input clock of baud rate generator

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

Note : To use baud rate generator, set TRUN<PRRUN>to "1", putting the prescaler in RUN mode.

Figure 3.11 (8). Baud Rate Generator Control Register (Channel 0, BR0CR)

SC1BUF (0054H)

	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)

Figure 3.11 (9). Serial Transmission/Receiving Buffer Registers (Channel 1, SC1BUF)

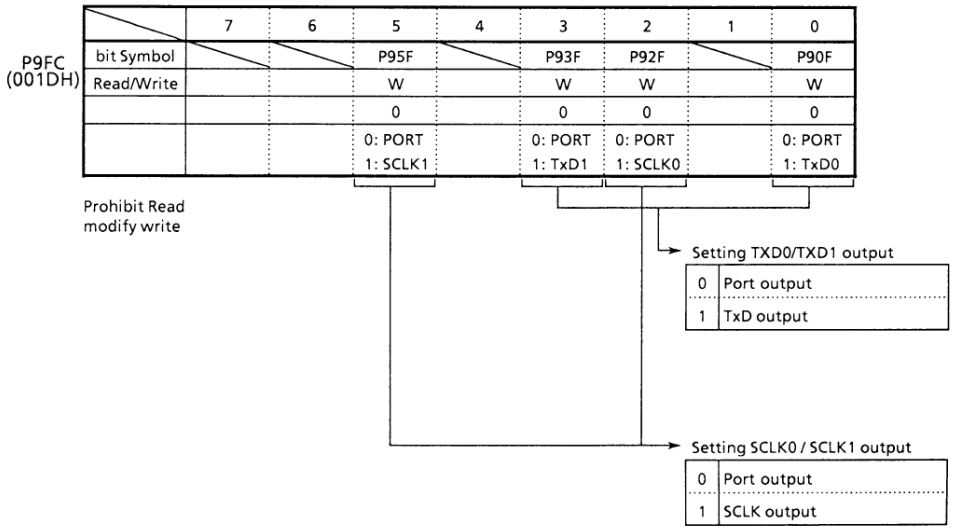
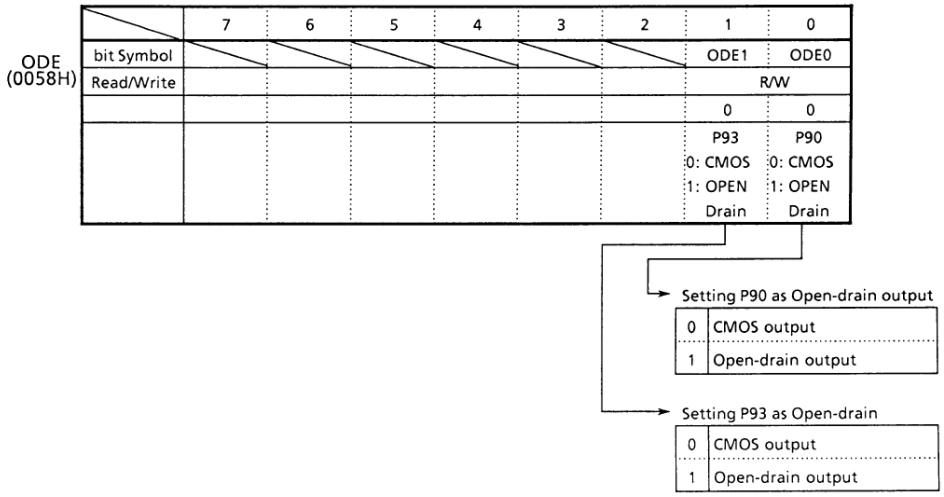


Figure 3.11 (10). Port 9 Function Register (P9FC)



Port 3.11 (11). Port 9 Open Drain Enable Register (ODE)

3.11.2 Configuration

Figure 3.11 (12) shows the block diagram of the serial channel 0.

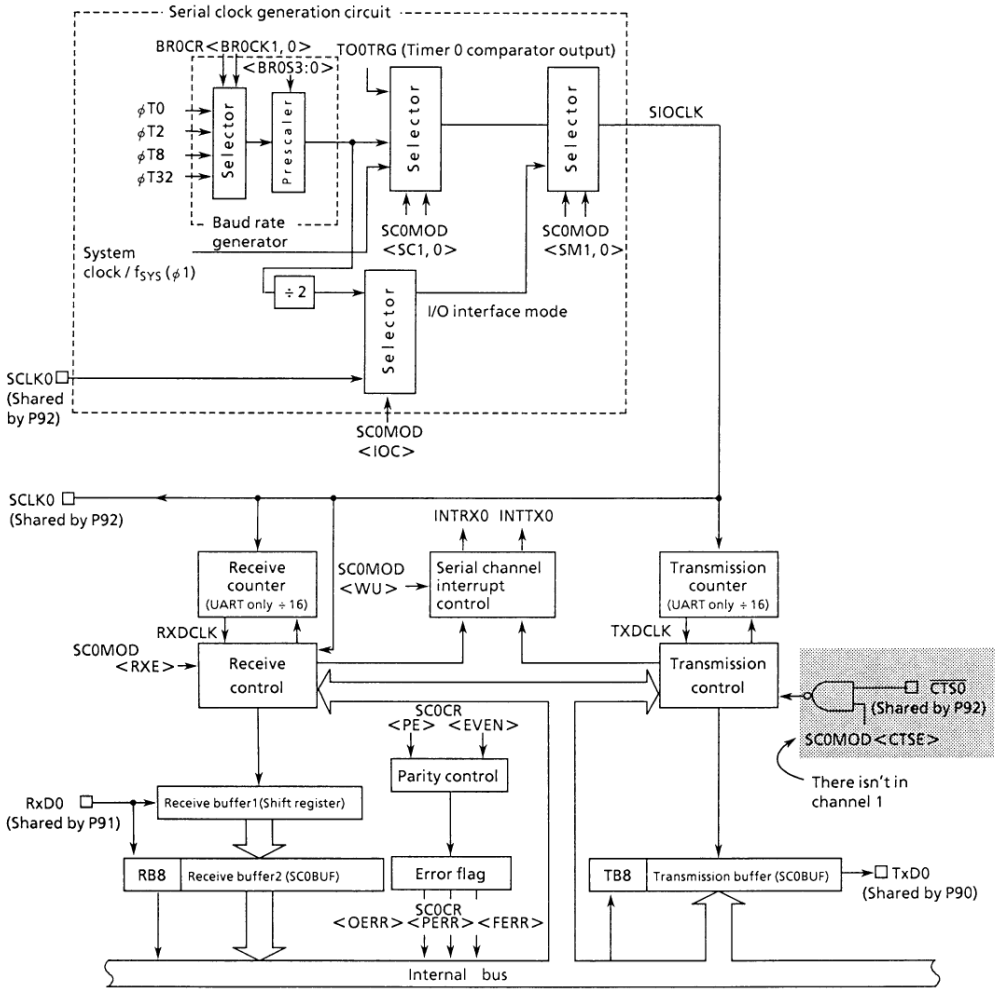


Figure 3.11 (12). Block Diagram of the Serial Channel 0

Figure 3.11 (13) shows the block diagram of the serial channel 1.

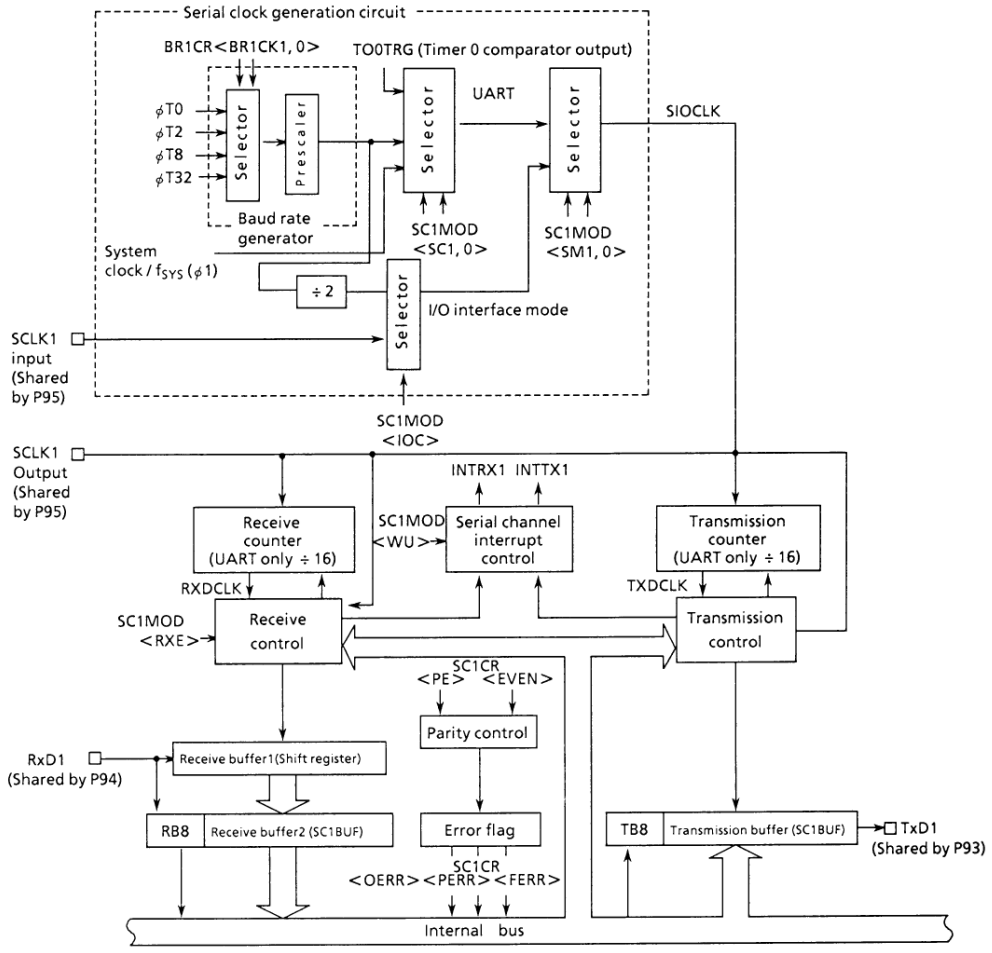


Figure 3.11 (13). Block Diagram of the Serial Channel 1

① Prescaler

There are 9 bit prescaler and prescaler clock selection to generate input clock for 8 bit Timer 0, 1, 16 bit Timer 4, 5 and Serial Interface 0, 1.

Figure 3.11 (14) shows the block diagram. Table 3.11 (1) shows prescaler clock resolution to the baud generator.

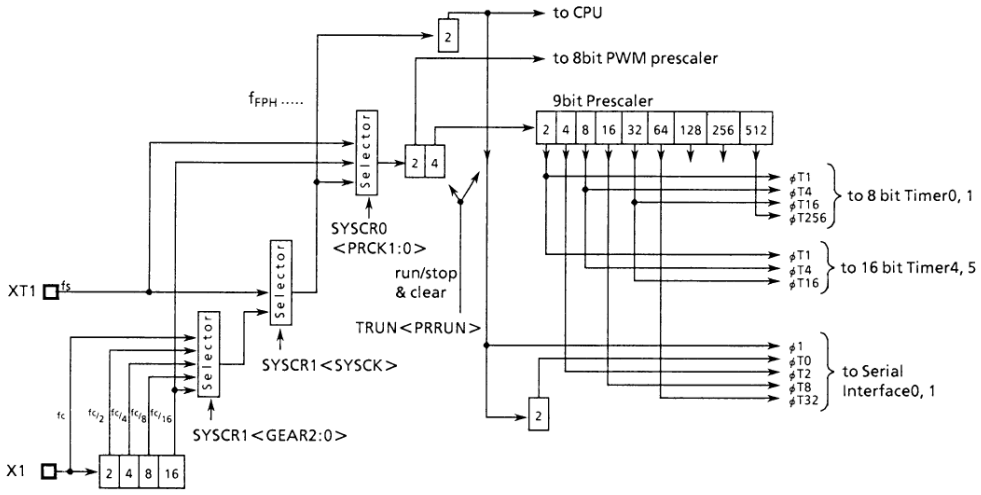


Figure 3.11 (14). Prescaler Block Diagram

Table 3.11 (1) Prescaler Clock Resolution to Baud Rate Generator

at fc = 16 MHz, fs = 32 kHz

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 ($fc/16$)	$fc/2^6$	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$
XXX	01 (low frequency clock)	XXX	—	$fs/2^4$	$fs/2^6$	$fs/2^8$
XXX	10 ($fc/16$ clock)	XXX	—	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$

XXX : don't care – : can not use

(Note) The fc / 16 clock as a prescaler prescaler clock can not be used when the fs is used as a system clock.

The 1/4 times clock selected among f_{FPH} clock, fc/16 clock, and fs clock is input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1 : 0>.

Resetting sets <PRCK1 : 0> to "00", therefore, f_{FPH} /4 clock is input.

The Baud Rate Generator uses 4 types of clock: øT0, øT2, øT8, and øT32 among the prescaler output.

The prescaler can be run or stopped by the timer control register TRUN <PRRUN>. Counting starts when <PRRUN> is set to "1", while the prescaler is cleared to zero and stops operation when <PRRUN> is set to "0".

When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop this prescaler before "HALT" instruction is executed.

② Baud Rate Generator

Baud rate generator comprises 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, øT0, øT2, øT8, or øT32 is generated by the 9-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 4-bit frequency divider, which divides frequency by 2 to 16 values to determine the transfer rate.

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

- UART mode

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

- I/O interface mode

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

Accordingly, when source clock fc is 12.288 MHz, input clock is ϕT2 (fc/16), and frequency divisor is 5, the transfer rate in UART mode becomes as follows:

$$\text{Transfer rate} = \frac{fc/16}{5} \div 16$$

$$= 12.288 \times 10^6/16/5/16 = 9600 \text{ (bps)}$$

Table 3.11 (2) shows an example of the transfer rate in UART mode.

Also with 8-bit timer 0, the serial channel can get a transfer rate. Table 3.9 (3) shows an example of baud rate using timer 0.

Table 3.11 (2) Selection of Transfer Rate (1) (When Baud Rate Generator is Used)

Unit (kbps)

fc [Mhz]	Input Clock / Frequency Divisor	ϕT0 (fc/4)	ϕT2 (fc/16)	ϕT8 (fc/64)	ϕT32 (fc/256)
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	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 rate in I/O interface mode is 8 times as fast as the values given in the above table.
Note 2: This table is calculated when fc is selected as a system clock, 1 as a clock gear, and system clock as a prescaler clock.

Table 3.11 (3) Selection of Transfer Rate (1) (When Timer 0 (Input Clock ϕT1) is Used)

Unit (Kbps)

TREG0 \ fc	12.288MHz	12MHz	9.8304MHz	8MHz	6.144MHz
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 timer 0 is used):

$$\text{Transfer rate} = \frac{fc}{TREG0 \times 8 \times 16}$$

↑ (When Timer 0 (input clock ϕT1) is used)

Note 1: Timer 0 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, 1 as a clock gear, and system clock as a prescaler clock.

③ Serial Clock Generation Circuit

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

1) I/O interface mode (channel 1 only)

When in SCLK output mode with the setting of SC1CR <IOC> = “0", the basic clock will be generated by dividing by 2 the output of the baud rate generator as described before. When in SCLK input mode with the setting of SC1CR <IOC> = “1", the rising edge or falling edge will be detected according to the setting of SC1CR <SCLKS> register to generate the basic clock.

2) Asynchronous Communication (UART) mode

According to the setting of SC0CR <SC1, 0>, the above baud rate generator clock, internal clock ϕ1 (500 Kbps @ fc = 16 MHz), or the match detect signal from timer 0 will be selected to generate the basic clock SIOCLK.

④ Receiving Counter

The receiving counter is a 4-bit binary counter used in asynchronous communication (UART) mode and counts up by SIOCLK clock. Sixteen pulses of SIOCLK are used for receiving one bit of data, and the data bit is sampled three times at 7th, 8th and 9th clock.

With the three samples, the received data is evaluated by the rule of majority.

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 that the received data is “0”.

⑤ Receiving Control

1) I/O interface mode (channel 1 only)

When in SCLK0 output mode with the setting of SC0CR <IOC> = “0", RxD0 signal will be sampled at the rising edge of shift clock which is output to SCLK0 pin.

When in SCLK0 input mode with the setting SC0CR <IOC> = “1", RxD0 signal will be sampled at the rising edge or falling edge of SCLK0 input according to the setting of SC0CR <SCLKS> register.

2) Asynchronous Communication (UART) mode

The receiving control has a circuit for detecting the start bit by the rule of majority. When two or more “0” are detected during three samples, it is recognized as start bit and the receiving operation is started.

Data being received is also evaluated by the rule of majority.

⑥ Receiving Buffer

To prevent overrun error, the receiving buffer has a double buffer structure.

Received data is stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data are stored in the receiving buffer 1, the stored data is transferred to another receiving buffer 2 (SC0BUF), generating an interrupt INTRX0/INTRX1. The CPU reads only receiving buffer 2 (SC0BUF). Even before the CPU reads the receiving buffer 2 (SC0BUF), the received data can be stored in the receiving buffer 1. However, unless the receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the 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 controllers is enabled by setting SC0MOD <WU> to “1", and interrupt INTRX0 occurs only when SC0CR <RB8> is set to “1”.

⑦ Transmission Counter

Transmission counter is a 4-bit binary counter which is used in asynchronous communication (UART) mode and, like a receiving counter, counts by SIOCLK clock, generating TxDCLK every 16 clock pulses.

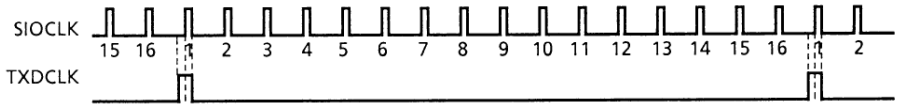


Figure 3.11 (15). Generation of Transmission Clock

⑧ Transmission Controller

1) I/O interface mode

In SCLK0 output mode with the setting of SC0CR <IOC> = “0", the data in the transmission buffer are output bit by bit to TxD0 pin at the rising edge of shift clock which is output from SCLK0 pin.

In SCLK0 input mode with the setting SC0CR <IOC> = “1", the data in the transmission buffer are output bit by bit to TxD0 pin at the rising edge or falling edge of SCLK0 input according to the setting of SC0CR <SCLKS> register.

2) Asynchronous Communication (UART) mode

When transmission data is written in 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 has a $\overline{\text{CTS0}}$ 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 SC0MOD <CTSE>.

When the $\overline{\text{CTS0}}$ pin goes high, after completion of the current data send, data send is halted until the $\overline{\text{CTS0}}$ pin goes low again. The INTTX0 Interrupts are generated, requests the next send data to the CPU.

Though there is no $\overline{\text{RTS}}$ pin, a handshake function can be easily configured by setting any port assigned to the $\overline{\text{RTS}}$ function. The $\overline{\text{RTS}}$ should be output “High” to request data send halt after data receive is completed by a software in the RXD interrupt routine.

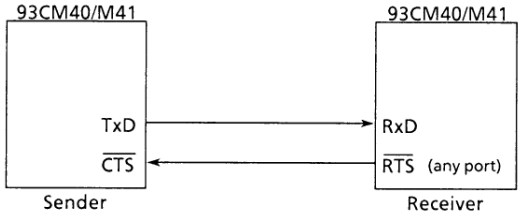
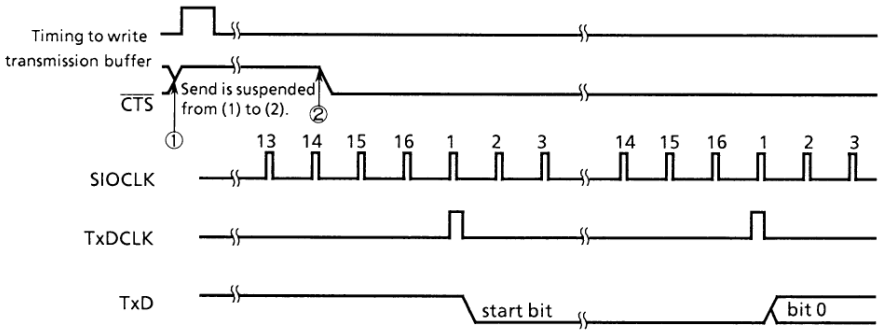


Figure 3.11 (16). Handshake Function



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 fall after the $\overline{CTS}$ signal falls.

Figure 3.11 (17). Timing of $\overline{CTS}$ (Clear to Send)

⑨ Transmission Buffer

Transmission buffer (SC0BUF) shifts to and sends the transmission data written from the CPU from the least significant bit (LSB) in order, using transmission shift clock TxDSFT which is generated by the transmission control. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0 interrupt.

⑩ Parity Control Circuit

When 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 mode. With SC0CR <EVEN> register, even (odd) parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer SCBUF, and data are transmitted after being stored in SC0BUF <TB7> when in 7-bit UART mode while in SC0MOD <TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before transmission data are written in the transmission buffer.

For receiving, data is shifted in the receiving buffer 1, and parity is added after the data is transferred in the receiving buffer 2 (SC0BUF/SC1BUF), and then compared with SC0BUF <RB7> when in 7-bit UART mode and with SC0MOD <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 (SCBUF0), an overrun error will occur.

2. Parity error <PERR>

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

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 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt timing	Center of last bit (Bit 8)	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 (Bit 8)	Center of last bit (parity bit)	Center of stop bit
Overrun error timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit

Note: Framing error occurs after an interrupt has occurred. Therefore, to check for framing error during interrupt operation, it is necessary to wait for 1 bit period of transfer rate.

Transmitting

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt timing	Just before last bit is transmitted.	←	←

2) I/O Interface mode

Transmission interrupt timing	SCLK output mode	Immediately after rise of last SCLK signal. (See Figure 3.11 (20))
	SCLK input mode	Immediately after rise of last SCLK signal (rising mode), or immediately after fall in falling mode. (See Figure 3.11 (21))
Receiving interrupt timing	SCLK output mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after last SCLK. (See Figure 3.11 (22))
	SCLK input mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after SCLK. (See Figure 3.11 (23))

3.11.3 Operational Description

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

This mode is used to increase the number of I/O pins for transmitting or receiving data to or from the external shifter register.

This mode includes SCLK output mode to output synchronous clock SCLK and SCLK input mode to input external synchronous clock SCLK.

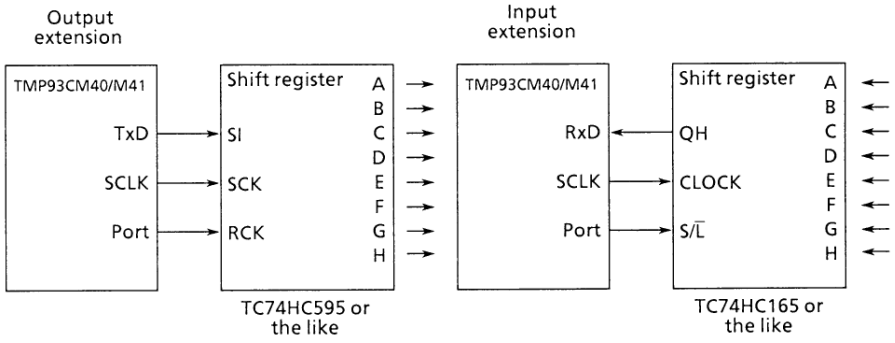


Figure 3.11 (18). Example of SCLK Output Mode Connection

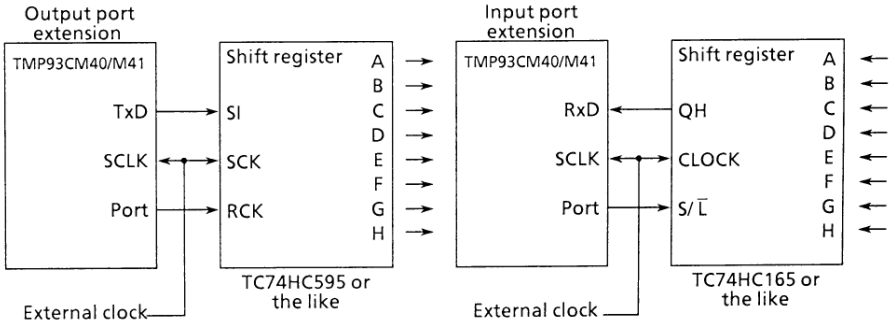


Figure 3.11 (19). Example of SCLK Input Mode Connection

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 in the transmission buffer. When all data is output, INTES0 <ITX0C0> will be set to generate INTTX0 interrupt.

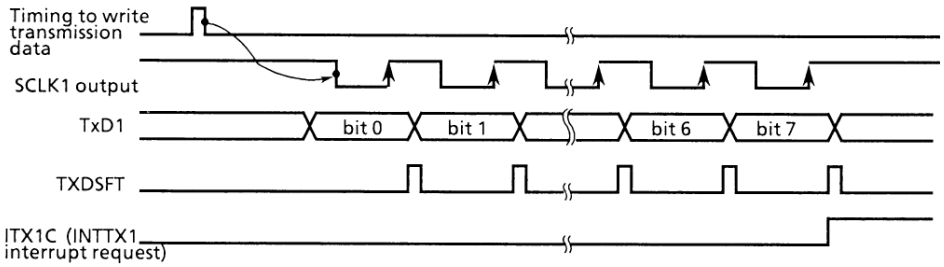


Figure 3.11 (20) Transmitting Operation in I/O Interface Mode (SCLK Output Mode) (Channel 1)

In SCLK output mode, 8-bit data are output from TxD0 pin when SCLK0 input becomes active while data are written in the transmission buffer by CPU. When all data are output, INTES0 <ITX0C> will be set to generate INTTX0 interrupt.

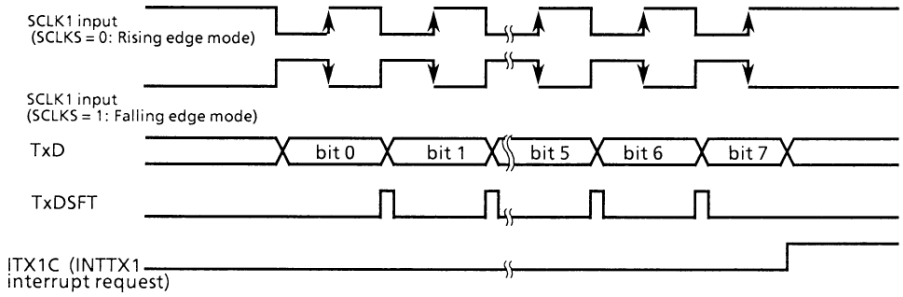


Figure 3.11 (21). Transmitting Operation in I/O Interface Mode (SCLK Input Mode) (Channel 1)

② Receiving

In SCLK output mode, synchronous clock is output from SCLK0 pin and the data is shifted in the receiving buffer 1 whenever the receive interrupt flag INTES0 <IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred in the receiving buffer 2 (SC0BUF) at the timing shown below, and INTES0 <IRX0C> will be set again to generate INTRX0 interrupt.

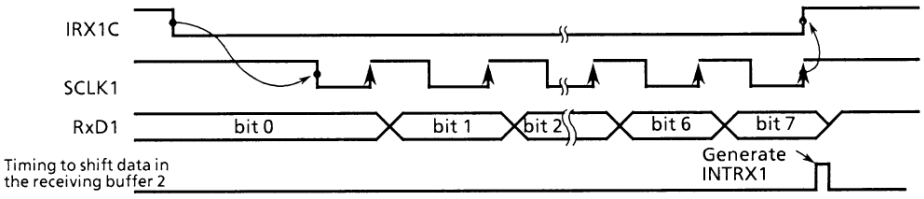


Figure 3.11 (22). Receiving Operation in I/O Interface Mode (SCLK1 Output Mode) (Channel 1)

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

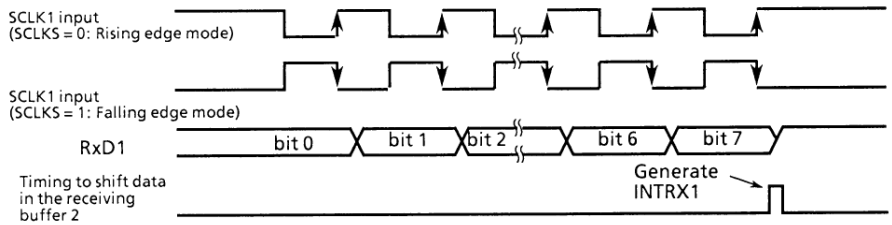


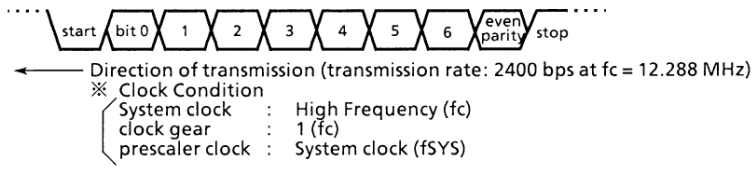
Figure 3.11 (23). Receiving Operation in I/O Interface Mode (SCLK Input Mode) (Channel 1)

Note: For data receiving, the system must be placed in the receive enable state (SC0MOD <RXE> = "1")

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

The 7-bit mode can be set by setting serial channel mode register SC0MOD <SM1, 0> to “01”.
In this mode, a parity bit can be added, and the addition of a 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.



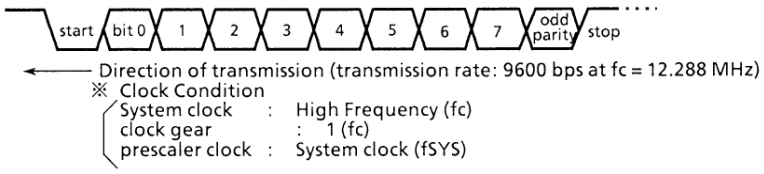

```
                7 6 5 4 3 2 1 0
P9CR    ← X X - - - - - 1
P9FC    ← X X - X - X X 1      } Select P90 as the TxD pin.
SC0MOD  ← X 0 - X 0 1 0 1      Set 7-bit UART mode.
SC0CR   ← X 1 1 X X X 0 0      Add an even parity.
BR0CR   ← 0 X 1 0 0 1 0 1      Set transfer rate at 2400 bps.
TRUN    ← 1 X - - - - - -      Start the prescaler for the baud rate generator.
INTES0  ← 1 1 0 0 - - - -      Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF  ← * * * * * * * *      Set data for transmission.
```

Note: X; don’t care - ; no change

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

The 8-bit UART mode can be specified by setting SC0MOD <SM1, 0> to “10”. In this mode, 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.



Main setting

```
                7 6 5 4 3 2 1 0
P9CR    ← X X - - - - 0 -       Select P91 (RxD) as the input pin.
SC0MOD  ← - 0 1 X 1 0 0 1       Enable receiving in 8-bit UART mode.
SC0CR   ← X 0 1 X X X 0 0       Add an odd parity.
BR0CR   ← 0 X 0 1 0 1 0 1       Set transfer rate at 9600 bps.
TRUN    ← 1 X - - - - - -       Start the prescaler for the baud rate generator.
INTES0  ← - - - - 1 1 0 0       Enable INTTX0 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.
```

Note : X ; don't care - ; no change

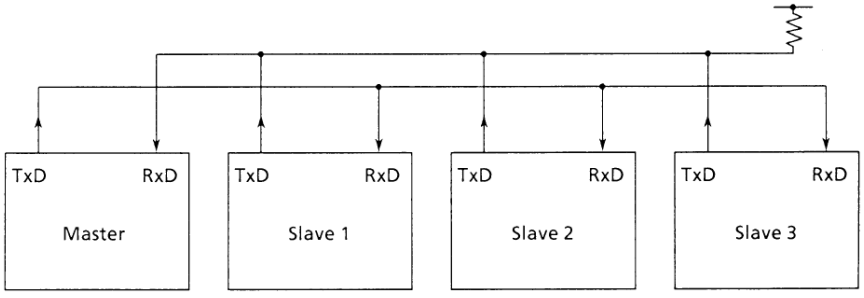
(4) Mode 3 (9-bit UART Mode)

The 9-bit UART mode can be specified by setting SC0MOD <SM1, 0> to "11". In this mode, parity bit cannot be added

For transmission, the MSB (9th bit) is written in SC0M0D <TB8>, while in receiving it is stored in SCCR <RB8>. For writing and reading the buffer, the MSB is read or written first, then SC0BUF.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SC0MOD <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.

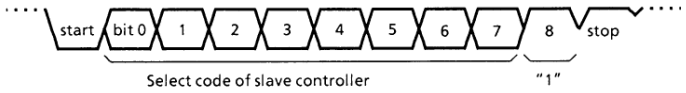
Figure 3.11 (24). Serial Link Using Wake-Up Function

Protocol

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

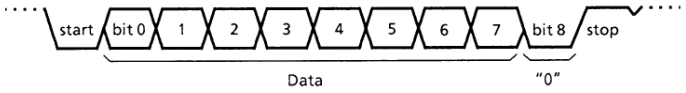
② Set SC0MOD <WU> bit of each slave controller to "1" to enable data receiving.

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



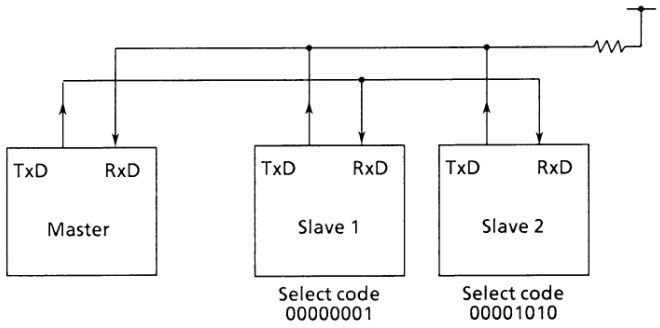
④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.

⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU> bit is cleared to "0." The MSB (bit 8) <TB8> is cleared to "0".



⑥ The other slave controllers (with the <WU> bit remaining at "1") ignore the receiving data because their MSBs (bit 8 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 ϕ1 (fc/2) 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   ← X X - - - - 0 1   } Select P90 as TxD0 pin and P91 as RxD0 pin.
P9FC   ← X X - X - X X 1   }
INTES0 ← 1 1 0 0 1 1 0 1     Enable INTTX0 and set the interrupt level 4.
                             Enable INTTX0 and set the interrupt level 5.
SC0MOD ← 1 0 1 0 1 1 1 0     Set φ1 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

SC0MOD ← 0 - - - - - - -     Sets TB8 to"0".
SC0BUF ← * * * * * * * *     Set data for transmission.
```

- Setting the slave controller 2

```
Main

P9CR   ← X X - - - - 0 1   } Select P91 as RxD0 pin and P90 as TxD0 pin (open drain
P9FC   ← X X - X - X X 1   } output).
ODE    ← X X X X X X - 1   }
INTES0 ← 1 1 0 1 1 1 1 0     Enable INTRX0 and INTTX0.
SC0MOD ← 0 0 1 1 1 1 1 0     Set <WU> to "1" in the 9-bit UART transmission mode
                             with transfer clock φ1.

INTRX0 interrupt

Acc ← SC0BUF
if Acc = Select code
Then SC0MOD ← - - - 0 - - - -   Clear <WU> to "0".
```

3.12 Analog/Digital Converter

TMP93CM40/M41 contains a high-speed analog/digital converter (A/D converter) with 8-channel analog input that features 10-bit successive approximation.

Figure 3.12 (1) shows the block diagram of the A/D converter. The 8-channel analog input pins (AN7 to AN0) are shared by input-only P5 and so can be used as input port.

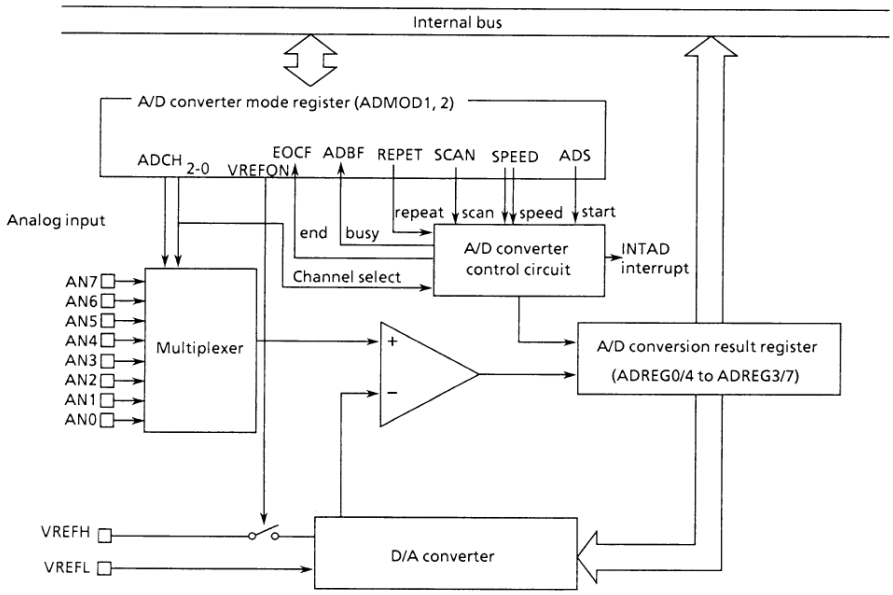


Figure 3.12 (1). Block Diagram of A/D Converter

Note: This A/D converter does not have a built-in sample and hold circuit. Therefore, when A/D converting high-frequency signals, connect a sample and hold circuit externally.

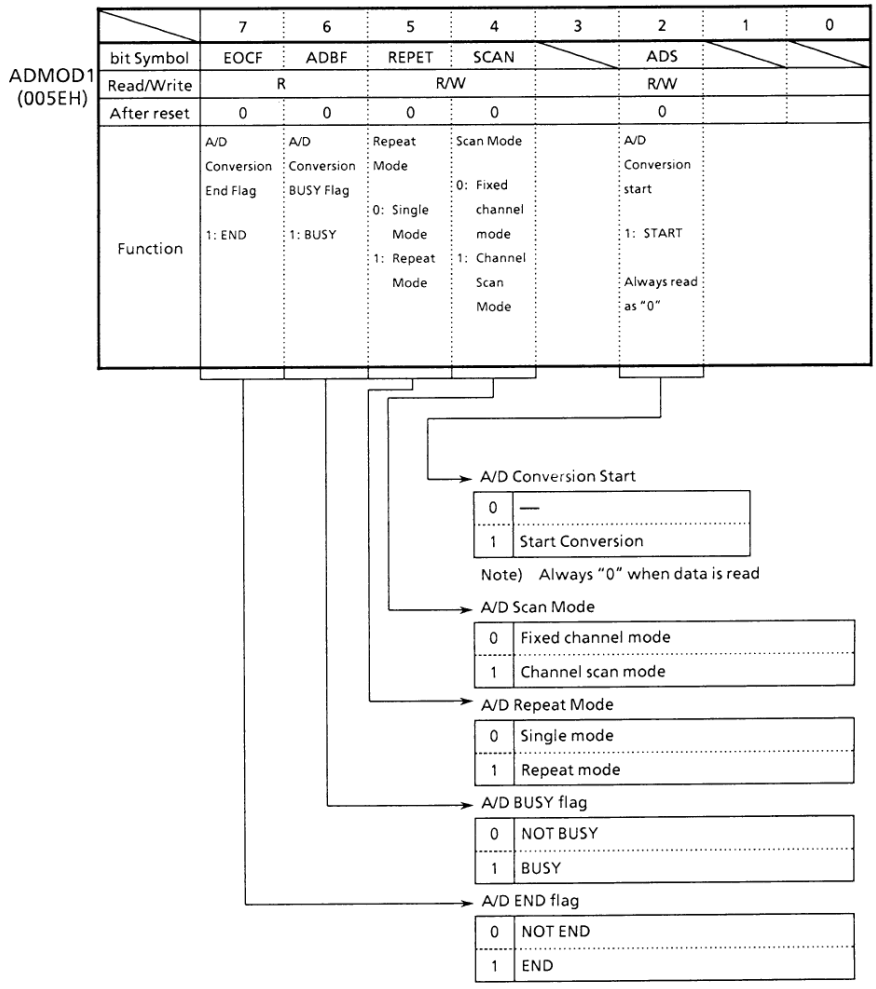


Figure 3.12 (2-1). A/D Control Register

ADMOD2 (005FH)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	1		0	0		0	0	0
Function	Ladder Resistance ON / OFF		Conversion Spead 00 : 160 states 01 : 320 states 10 : 640 states 11 : 1280 states			Analog Input Channel Selection		

Analog Input Channel Selection

<ADCH2,1,0> \ <SCAN>	0	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

Conversion Speed

00	160 states (20 μs at 16 MHz)
01	320 states (40 μs at 16 MHz)
10	640 states (80 μs at 16 MHz)
11	1280 states (160 μs at 16 MHz)

Ladder Resistance ON / OFF selection

0	Ladder Resistance OFF
1	Ladder Resistance ON

(Note) Set the <VREFON> register to "1" before starting (write <ADS> to "1") conversion.

Figure 3.12 (2-2). A/D Control Register

ADREG04L (0060H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN0 are stored.							

ADREG04H (0061H)

	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	Upper 8 bits of A/D result for AN0 are stored.							

ADREG15L (0062H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN1 are stored.							

ADREG15H (0063H)

	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	Upper 8 bits of A/D result for AN1 are stored.							

Note) The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7.

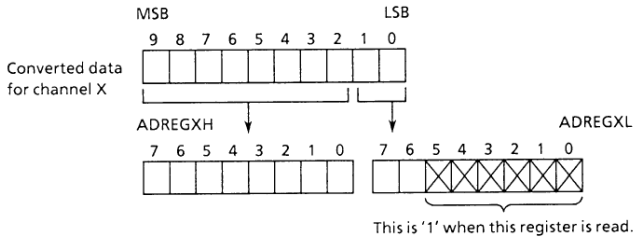


Figure 3.12 (3-1). A/D Conversion Result Register (ADREG04, 15)

ADREG26L (0064H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN2 are stored.							

ADREG26H (0065H)

	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	Upper 8 bits of A/D result for AN2 are stored.							

ADREG37L (0066H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN3 are stored.							

ADREG37H (0067H)

	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	Upper 8 bits of A/D result for AN3 are stored.							

Note) The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7.

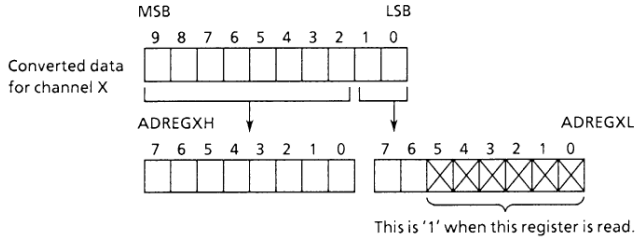


Figure 3.12 (3-2). A/D Conversion Result Register (ADREG04, 15)

3.12.1 Operation

(1) Analog Reference Voltage

High analog reference voltage is applied to the VREFH pin, and low analog reference voltage is applied to VREFL pin.

The reference voltage between VREFH and VREFL is divided by 1024 using ladder resistance, and compared with the analog input voltage for A/D conversion.

The switch between VREFH and VREFL can be cut (OFF) by writing "0" to <VREFON>.

When the conversion will be started at the <VREFON> = "0", write "1" to <VEFRON> before writing "1" to <ADS>.

(2) Analog Input Channels

Analog input channel is selected by ADMOD <ADCH2:0>. However, which channel to select depends on the operation mode of the A/D converter.

In fixed analog input mode, one channel is selected by ADMOD <ADCH2:0> among four pins: AN0 to AN7.

In analog input channel scan mode, the number of channels to be scanned from AN0 or AN4 is specified by <ADCH2:0>, such as AN0 → AN1, AN0 → AN1 → AN2, AN0 → AN1 → AN2 → AN3, AN4 → AN5, AN4 → AN5 → AN6, and AN4 → AN5 → AN6 → AN7.

When reset, A/D conversion channel register will be initialized to ADMOD <ADCH2:0> = 00, so that AN0 pin will be selected.

The pins which are not used as analog input channel can be used as ordinary input port P5.

(3) Starting A/D Conversion

A/D conversion starts when A/D conversion register ADMOD1 <ADS> is written "1". When A/D conversion starts, A/D conversion busy flag ADMOD1 <ADBF> which indicates "conversion is in progress" will be set to "1".

(4) A/D Conversion Mode

Both fixed A/D conversion channel mode and A/D conversion channel scan mode have two conversion modes, i.e., single and repeat conversion modes.

In fixed channel repeat mode, conversion of specified one channel is executed repeatedly.

In scan repeat mode, scanning from AN0, ··· → AN3 is executed repeatedly.

A/D conversion mode is selected by ADMOD1 <REPET, SCAN>.

(5) A/D Conversion Speed Selection

There are four A/D conversion speed modes. The selection is executed by ADMOD2 <SPEED1:0> register.

When reset, ADMOD <SPEED1:0> will be initialized to "00", so that high speed conversion mode will be selected.

(6) A/D Conversion End and Interrupt

- A/D conversion single mode

ADMOD1 <EOCF> for A/D conversion end will be set to "1," ADMOD1 <ADBF> flag will be reset to "0," and INTAD interrupt will be enabled when A/D conversion of specified channel ends in fixed conversion channel mode or when A/D conversion of the last channel ends in channel scan mode.

- A/D conversion repeat mode

For both fixed conversion channel mode and conversion channel scan mode, INTAD should be disabled when in repeat mode. Always set the INTE0AD at "000", so that it disables the interrupt request.

Write "0" to ADMOD2 <REPET> to end the repeat mode. Then, the repeat mode will be exited as soon as the conversion in progress is completed.

(7) Storing the A/D Conversion Result

The results of A/D conversion are stored in ADREG04 to ADREG37 registers for each channel.

The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7.

However, the current conversion data cannot be known which channels.

In repeat mode, the registers are updated whenever conversion ends.

ADREG04 to ADREG37 are read-only registers.

(8) Reading the A/D Conversion Result

The results of A/D conversion are stored in ADREG04 to ADREG37 registers.

When the contents of one of ADREG04L, ADREG15L, ADREG26L, and ADREG37L registers are read, ADMOD1 <EOCF> will be cleared to "0".

<EOCF> is not cleared to "0" when the contents of one ADREG04L, ADREG15L, ADREG26L, and ADREG37L for lower 2-bits is read.

Setting example: ① When the analog input voltage of the AN3 pin is A/D converted by 160 states speed and the result is transfered in the memory address 100H by A/D interrupt INTAD routine

Main setting

```
INTE0AD  ← 1 1 0 0 - - - -    Enable INTAD and set interrupt level 4.
ADMOD2   ← 1 X 0 0 X 0 1 1    Specify AN3 pin as an analog input channel and
ADMOD1   ← X X 0 0 X 1 X X    starts A/D conversion in 160 states speed mode.
```

INTAD routine

```
WA          ← ADREG37         Read ADREG37L and ADREG37H values and write
                              to WA (16 bit)
WA          >> 6              Right-shifts WA six times and writes 0 in upper
                              bits.
(000100H)←   WA               Writes contents of WA in memory at 100H
```

② When the analog input voltage of the AN4 to AN7 pins (4 pins) are A/D converted by 320 states speed and set the channel scan & repeat mode.

Main setting

```
INTE0AD  ← 1 0 0 0 - - - -    INTAD disenable.
ADMOD2   ← 1 X 0 1 0 1 1 1    Specify AN4 to AN7 pins as input channel and scan
ADMOD1   ← X X 1 1 X 1 0 0    & repeat mode and starts A/D conversion in 320
                              states speed mode .
```

```
Note)   X: don't care
        -: no change
```

3.13 Watchdog Timer (Runaway Detecting Timer)

TMP93CM40/M41 contain 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 to notify the CPU of the malfunction, and outputs "0" externally from watchdog timer out pin WDTOUT to notify the peripheral devices of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

This binary counter is also used as a warming up timer for the internal oscillator stabilization. This is used when the STOP releasing and before changing system clock.

3.13.1 Configuration

Figure 3.13 (1) shows the block diagram of the watchdog timer (WDT).

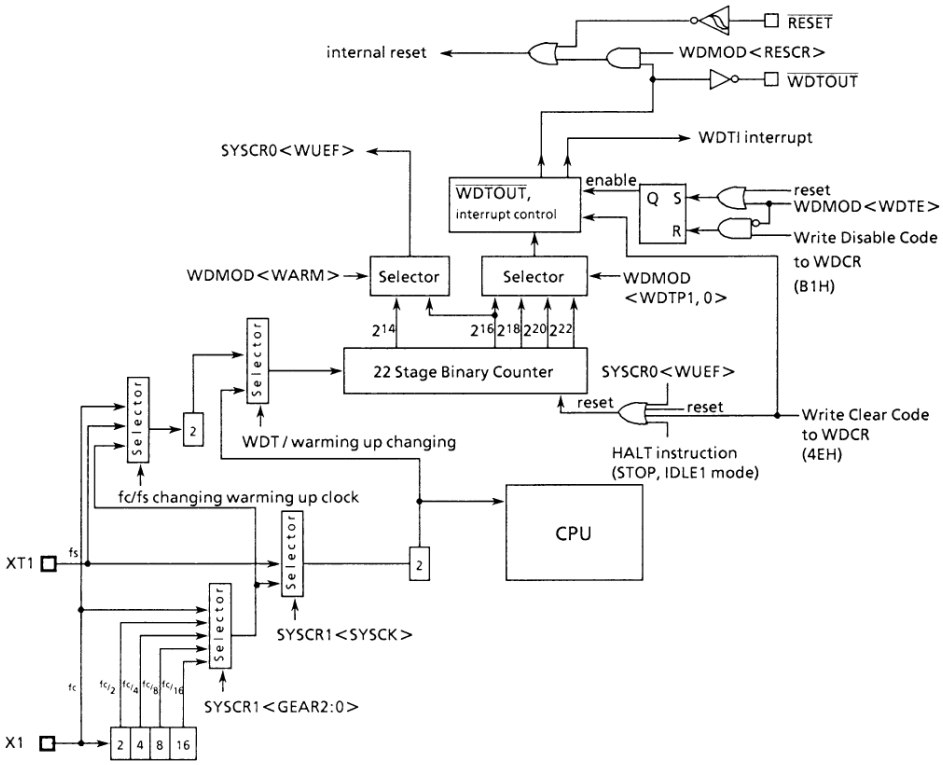


Figure 3.13 (1). Block Diagram of Watchdog Timer/Warming Up Timer

The watchdog timer is a 22-stage binary counter which uses ϕ (fc/2) as the input clock. There are four outputs from the binary counter: 2^{16}/fc, 2^{18}/fc, 2^{20}/fc, and 2^{22}/fc. Selecting one of the outputs with the WDMOD register generates a watchdog interrupt, and outputs watchdog timer out when an overflow occurs.

Since the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0" due to a watchdog timer overflow, the peripheral devices can be reset. The watchdog timer out pin is set to "1" after disabling WDT and clearing the watchdog timer (by writing a clear code 4EH in the WDCR register).

(Example)

```
LDW     (WDMOD), B100H    ;    disable
LD      (WDCR), 4EH       ;    write clear code
SET     7, (WDMOD)        ;    enable again
```

In other words, the $\overline{\text{WDTOUT}}$ keeps outputting "0" until the clear code is written.

The watchdog timer out pin can also be connected to the reset pin internally. The watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs 0 to 8 to 20 states (16 to 40 μs @ fc = 16MHz) and resets itself.

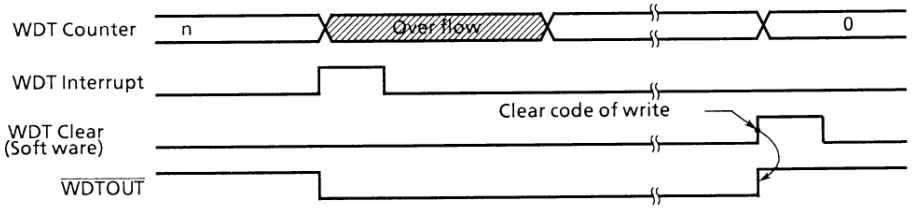


Figure 3.13 (2). Normal Mode

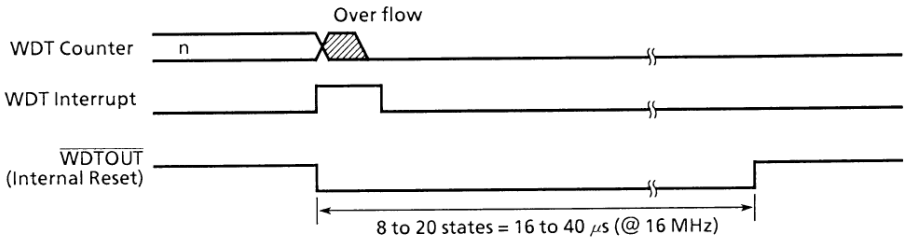


Figure 3.13 (3). Reset Mode

3.13.2 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, and therefore 2^{16}/fc is set. (The number of states is approximately 32,768).

The detecting time of WDT is shown in Figure 3.13 (6).

② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD <WDTE> is initialized to "1"

- Disable control

```
WDMOD ← 0 - - - - - X X    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".

To disable, it is necessary to clear 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 the binary counter of the watchdog timer function.

- 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 (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM0	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	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: $2^{21}/f_{SYS}$		Warming Up time 0: 2^{14}/ inputted frequency 1: 2^{16}/ inputted frequency	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDEL2 mode care		1: Internally connects WDT out to the reset pin	1: Drive the pin even in STOP mode

DRVE (explanation by stop mode)

Watchdog timer out control

0	—
1	Connects WDT out to a reset

Select the standby mode HALT instruction

00	RUN mode (Only the CPU stops)
01	STOP mode (All circuits stop)
10	IDLE1 mode (Only the oscillator operates)
11	IDLE2 mode (CPU and A/D stop)

Select the detecting period of watchdog timer

0	2^{14}/inputted frequency (approx. 1.0 ms at 16 MHz)
1	2^{16}/inputted frequency (approx. 4.1 ms at 16 MHz)

Select the detecting time of watchdog timer

00	$2^{15}/f_{SYS}$ (approx. 4.1 ms at 16 MHz)
01	$2^{17}/f_{SYS}$ (approx. 16 ms at 16 MHz)
10	$2^{19}/f_{SYS}$ (approx. 66 ms at 16 MHz)
11	$2^{21}/f_{SYS}$ (approx. 262 ms at 16 MHz)

Watchdog timer Enable / Disable control

0	Disable
1	Enable

Figure 3.13 (4). Watchdog Timer Mode Register

WDCR (005DH)

	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	—

Figure 3.13 (5). Watchdog Timer Control Register

at fc = 16 MHz, fs = 32 kHz

System clock selection <SYSCK>	Gear value <GEAR2 : 0>	Watch Dog Timer Detecting Time			
		$2^{15}/f_{SYS}$	$2^{17}/f_{SYS}$	$2^{19}/f_{SYS}$	$2^{21}/f_{SYS}$
1 (fs)	XXX	2.048 s	8.192 s	32.768 s	131.072 s
0 (fc)	000 (fc)	4.096 ms	16.384 ms	65.536 ms	262.1 ms
	001 ($f_c/2$)	8.192 ms	32.768 ms	131.072 ms	512 ms
	010 ($f_c/4$)	16.384 ms	65.536 ms	262.1 ms	1.024 s
	011 ($f_c/8$)	32.768 ms	131.072 ms	512 ms	2.048 s
	100 ($f_c/16$)	65.536 ms	262.1 ms	1.024 s	4.096 s

XXX : don't care

Figure 3.13 (6). Watchdog Detecting Time

3.13.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD <WDTP1, 0> register and outputs a low level signal. 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-malfunction program. By connecting the watchdog timer out pin to peripheral devices' resets, a CPU malfunction can also be acknowledged to other devices.

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE1 and STOP modes. In the RUN mode, the watchdog timer is enabled. When the bus is released ($\overline{BUSAK}$ = "L"), WDT continues counting up.

However, the function can be disabled when entering the RUN, IDLE2 modes.

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 $2^{18} / f_{SYS}$

```
WDMOD ← 1 0 1 - - - X X
```

③ Disable the watchdog timer.

```
WDMOD ← 0 - - - - - X X        Clear WDTE to "0".
WDCR  ← 1 0 1 1 0 0 0 1        Write disable code (B1H).
```

④ Set IDLE1 mode.

```
WDMOD ← 0 - - - 1 0 X X        Disables WDT and sets IDLE1 mode.
WDCR  ← 1 0 1 1 0 0 0 1
Executes HALT command          Set the standby mode
```

⑤ Set the STOP mode (warming up time: $2^{16} / f_{SYS}$)

```
WDMOD ← - - - 1 0 1 X X        Set the STOP mode.
Executes HALT command.         Execute HALT instruction. Set the standby
                               mode.
```

4. Electrical Characteristics

4.1 Absolute Maximum (TMP93CM40F/TMP93CM41F)

Symbol	Parameter	Rating	Unit
V_{cc}	Power Supply Voltage	-0.5 to 6.5	V
V IN	Input Voltage	-0.5 ~ V_{cc} + 0.5	V
Σ IOL	Output Current (total)	120	mA
Σ IOH	Output Current (total)	-80	mA
PD	Power Dissipation (Ta = 85°C)	600	mW
T SOLDER	Soldering Temperature (10s)	260	°C
T STG	Storage Temperature	-65 to 150	°C
T OPR	Operating Temperature	-40 to 85	°C

4.2 DC Characteristics (1/2)

Symbol	Parameter	Max	Typ. (Note 1)	Max	Unit	Test Condition
V_{cc}	Power Supply Voltage	4.5		5.5	V	fc = 4 to 16MHz fs = 30 to 34kHz (Ta = -40 to 85°C)
						fc = 4 to 20MHz fs = 30 to 34kHz (Ta = -20 to 70°C)
		2.7 (Note 2)				fc = 4 to 10MHz fs = 30 to 34kHz (Ta = -40 to 85°C)
V IL	Low High Voltage (AD0 to 15)			0.8 0.6	V	$V_{CC} \geq 4.5V$ $V_{CC} < 4.5V$
V IL1	Port 2 to A (except P87, P5)			0.3 V_{cc}		
V IL2	$\overline{RESET}$, $\overline{NMI}$, INT0			0.25 V_{cc}		V_{CC} = 2.7 to 5.5V
V IL3	$\overline{EA}$, AM8/$\overline{16}$			0.3		
V IL4	X1, P5			0.2 V_{cc}		
V IH	Low High Voltage (AD0 to 15)	2.2 2.0				$V_{CC} \geq 4.5V$ $V_{CC} < 4.5V$
V IH1	Port 2 to A (except P87)	0.7 V_{cc}		V_{cc} + 0.3		
V IH2	$\overline{RESET}$, $\overline{NMI}$, INT0	0.75 V_{cc}				V_{CC} = 2.7 to 5.5V
V IH3	$\overline{EA}$, AM8/$\overline{16}$	V_{cc} - 0.3				
V IH4	X1	0.8 V_{cc}				
V OL	Output Low Voltage			0.45	V	I OL = 1.6mA (V_{CC} = 2.7 to 5.5V)
V OH	Output High Voltage	2.4			V	I OH = -400μA (V_{CC} = 2.7 to 5.5V)
V OH1		0.75 V_{cc}			V	I OH = -100μA (V_{CC} = 5V±10%)
V OH2		0.9 V_{cc}			V	I OH = - 20μA (V_{CC} = 5V±10%)
I DAR (note 3)	Darlington Drive Current (8 Output Pins max.)	-1.0		-3.5	mA	V_{EXT} = 1.5V R_{EXT} = 1.1KΩ (V_{CC} = 5V±10%)
I LI	Input Leakage Current		0.02 (Typ)	±5	μA	$0.0 \leq V_{in} \leq V_{cc}$
I LO	Output Leakage Current		0.05 (Typ)	±10		$0.2 \leq V_{in} \leq V_{cc}$ - 0.2

(note 1) Typical values are for Ta = 25°C and V_{CC} = 5V unless otherwise.
(note 2) The operation of A/D converter is guaranteed at V_{CC} = 5V±10%.
(note 3) I-DAR is guaranteed for a total of up to 8 ports.
(note 4)The condition of measurement of ICC (Normal/Slow).
Operates only CPU, output ports are open and input ports fixed.

4.2 DC Characteristics (1(2/2)

Symbol	Parameter	Max	Typ. (Note 1)	Max	Unit	Test Condition
V STOP	Power Down Voltage (at STOP, RAM Back up)	2.0		6.0	V	V_{IL2} = 0.2V_{CC}, V_{IH2} = 0.8V_{CC}
R RST	$\overline{\text{RESET}}$ Pull Up Register	50 80		150 200	KΩ	V_{CC} = 5V±10% V_{CC} = 3V±10%
C IO	Pin Capacitance			10	pF	tosc = 1MHz
V TH	Schmitt Width $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INTO (P87)	0.4	1.0		V	
R KL	Programmable Pull Down Register	10 30		80 150	KΩ	V_{CC} = 5V±10% V_{CC} = 3V±10%
R KH	Programmable Pull Up Register	50 100		150 300	KΩ	V_{CC} = 5V±10% V_{CC} = 3V±10%
I_{cc}	Operating Current (NORMAL) RUN IDLE2 IDLE1		16 14 8.0 1.0	25 25 15 5	mA	V_{CC} = 5V±10% fc = 20MHz
	Operating Current (NORMAL) RUN IDLE2 IDLE1		6.0 5.0 3.0 0.4	8 7 4 1.1		V_{CC} = 3V±10% fc = 10MHz (Typ.: V_{CC} = 3.0V)
	Operating Current (SLOW) RUN IDLE2 IDLE1		30 28 20 15	35 30 20 15	μA	V_{CC} = 3V±10% fs = 32.768kHz (Typ.: V_{CC} = 3.0V)
	STOP		0.2	10	μA	V_{CC} = 2.7 to 5.5V

(note 1) Typical values are for Ta = 25°C and V_{CC} = 5V unless otherwise.
(note 2) The operation of A/D converter is guaranteed at V_{CC} = 5V±10%.
(note 3) I-DAR is guaranteed for a total of up to 8 ports.
(note 4)The condition of measurement of ICC (Normal/Slow).
Operates only CPU, output ports are open and input ports fixed.

4.3 AC Electrical Characteristics

(1) V_{cc} = 5V±10%

No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{OSC}	Osc. Period (= x)	50	250	62.5		50		ns
2	t_{CLK}	CLK width	2x - 40		85		60		ns
3	t_{AK}	A0 - 23 Valid→CLK Hold	0.5x - 20		11		5		ns
4	t_{KA}	CLK Valid→A0 - 23 Hold	1.5x - 70		24		5		ns
5	t_{AL}	A0-15 Valid→ALE fall	0.5x - 15		16		10		ns
6	t_{LA}	ALE fall→A0 - 15 Hold	0.5x - 15		11		5		ns
7	t_{LL}	ALE High width	x - 40		23		10		ns
8	t_{LC}	ALE fall→$\overline{RD}$/$\overline{WR}$ fall	0.5x - 30		6		0		ns
9	t_{CL}	$\overline{RD}$/$\overline{WR}$ rise→ALE rise	0.5x - 20		11		5		ns
10	t_{ACL}	A0 - 15 Valid→$\overline{RD}$/$\overline{WR}$ fall	x - 25		38		25		ns
11	t_{ACH}	A0 - 23 Valid→$\overline{RD}$/$\overline{WR}$ fall	1.5x - 50		44		25		ns
12	t_{CA}	$\overline{RD}$/$\overline{WR}$ rise→A0 - 23 Hold	0.5x - 25		6		0		ns
13	t_{ADL}	A0 - 15 Valid→D0 - 15 input		3.0x - 55		143		95	ns
14	t_{ADH}	A0 - 23 Valid→D0 - 15 input		3.5x - 65		154		110	ns
15	t_{RD}	$\overline{RD}$ fall→D0 - 15 input		2.0x - 60		75		40	ns
16	t_{RR}	$\overline{RD}$ Low width	2.0x - 40		85		60		ns
17	t_{HR}	$\overline{RD}$ rise→D0 - 15 Hold	0		0		0		ns
18	t_{RAE}	$\overline{RD}$ rise→A0 - 15 output	x - 15		48		35		ns
19	t_{WW}	$\overline{WR}$ Low width	2.0x - 40		85		60		ns
20	t_{DW}	D0 - 15 Valid→$\overline{WR}$ rise	2.0x - 55		70		45		ns
21	t_{WD}	$\overline{WR}$ rise→D0 - 15 Hold	0.5x - 15		16		10		ns
22	t_{AEH}	A0 - 23 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	t_{AWL}	A0 - 15 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	t_{CW}	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	t_{APH}	A0 - 23 Valid→PORT input		2.5x - 120		36		5	ns
26	t_{APH2}	A0 - 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t_{CP}	$\overline{WR}$ rise→PORT Valid		200		200		200	ns
28	t_{ASRH}	A0 - 23 Valid→$\overline{RAS}$ fall	1.0x - 40		23		10		ns
29	t_{ASRL}	A0 - 15 Valid→$\overline{RAS}$ fall	0.5x - 15		16		10		ns
30	t_{RAC}	$\overline{RAS}$ fall→D0 - 15 input		2.5x - 70		86		55	ns
31	t_{RAH}	$\overline{RAS}$ fall→A0 - 15 Hold	0.5x - 15		16		10		ns
32	t_{RAS}	$\overline{RAS}$ Low width	2.0x - 40		85		60		ns
33	t_{RP}	$\overline{RAS}$ High width	2.0x - 40		85		60		ns
34	t_{RSH}	$\overline{CAS}$ fall→$\overline{RAS}$ rise	1.0x - 35		23		10		ns
35	t_{RSC}	$\overline{RAS}$ rise→$\overline{CAS}$ rise	0.5x - 25		6		0		ns
36	t_{RCD}	$\overline{RAS}$ fall→$\overline{CAS}$ fall	1.0x - 40		23		10		ns
37	t_{CAC}	$\overline{CAS}$ fall→D0 - 15 input		1.5x - 65		29		10	ns
38	t_{CAS}	$\overline{CAS}$ Low width	1.5x - 30		64		40		ns

AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF
  (However CL = 100pF for AD0 ~ AD15, AD0 to AD23, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, CLK, $\overline{RAS}$, $\overline{CAS0}$ to $\overline{CAS2}$)
- Input Level: High 2.4V /Low 0.45V (AD0 to AD15)
  High 0.8Vcc /Low 0.2Vcc (Except for AD0 to AD15)

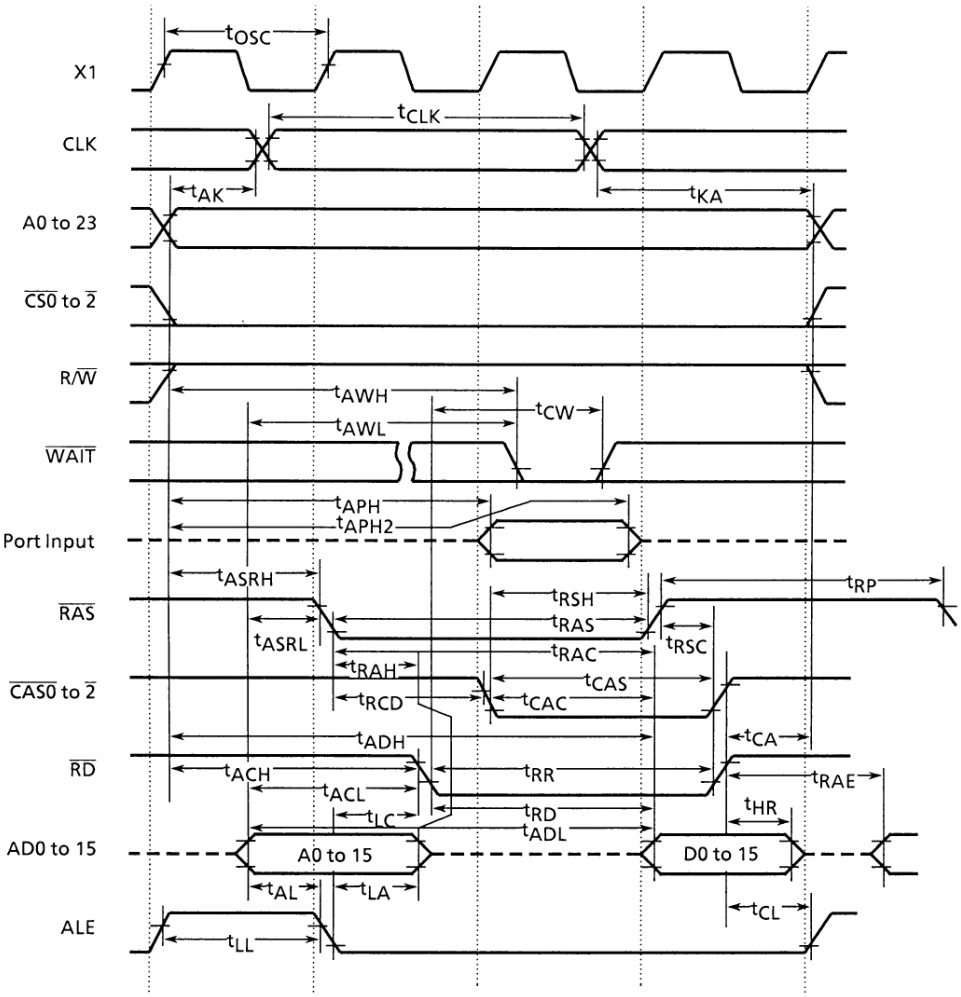
(2) V_{cc} = 3V±10% (TMP93CM40/M41F are guaranteed up to 10MHz operation)

No.	Symbol	Parameter	Variable		10MHz		Unit
			Min	Max	Min	Max	
1	t_{OSC}	Osc. Period (= x)	80	250	100		ns
2	t_{CLK}	CLK width	2x - 40		160		ns
3	t_{AK}	A0 - 23 Valid→CLK Hold	0.5x - 30		20		ns
4	t_{KA}	CLK Valid→A0 - 23 Hold	1.5x - 80		70		ns
5	t_{AL}	A0-15 Valid→ALE fall	0.5x - 35		15		ns
6	t_{LA}	ALE fall→A0 - 15 Hold	0.5x - 35		15		ns
7	t_{LL}	ALE High width	x - 60		40		ns
8	t_{LC}	ALE fall→$\overline{RD}$/$\overline{WR}$ fall	0.5x - 40		10		ns
9	t_{CL}	$\overline{RD}$/$\overline{WR}$ rise→ALE rise	0.5x - 40		10		ns
10	t_{ACL}	A0 - 15 Valid→$\overline{RD}$/$\overline{WR}$ fall	x - 50		50		ns
11	t_{ACH}	A0 - 23 Valid→$\overline{RD}$/$\overline{WR}$ fall	1.5x - 50		10		ns
12	t_{CA}	$\overline{RD}$/$\overline{WR}$ rise→A0 - 23 Hold	0.5x - 40		30		ns
13	t_{ADL}	A0 - 15 Valid→D0 - 15 input		3.0x - 110		100	ns
14	t_{ADH}	A0 - 23 Valid→D0 - 15 input		3.5x - 125		225	ns
15	t_{RD}	$\overline{RD}$ fall→D0 - 15 input		2.0x - 115		85	ns
16	t_{RR}	$\overline{RD}$ Low width	2.0x - 40		160		ns
17	t_{HR}	$\overline{RD}$ rise→D0 - 15 Hold	0		0		ns
18	t_{RAE}	$\overline{RD}$ rise→A0 - 15 output	x - 25		75		ns
19	t_{WW}	$\overline{WR}$ Low width	2.0x - 40		160		ns
20	t_{DW}	D0 - 15 Valid→$\overline{WR}$ rise	2.0x - 120		80		ns
21	t_{WD}	$\overline{WR}$ rise→D0 - 15 Hold	0.5x - 40		10		ns
22	t_{AEH}	A0 - 23 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.5x - 130		220	ns
23	t_{AWL}	A0 - 15 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.0x - 100		200	ns
24	t_{CW}	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ Hold (1WAIT + n mode)	2.0x + 0		200		ns
25	t_{APH}	A0 - 23 Valid→PORT input		2.5x - 120		130	ns
26	t_{APH2}	A0 - 23 Valid→PORT Hold	2.5x + 50		300		ns
27	t_{CP}	$\overline{WR}$ rise→PORT Valid		200		200	ns
28	t_{ASRH}	A0 - 23 Valid→$\overline{RAS}$ fall	1.0x - 60		40		ns
29	t_{ASRL}	A0 - 15 Valid→$\overline{RAS}$ fall	0.5x - 40		10		ns
30	t_{RAC}	$\overline{RAS}$ fall→D0 - 15 input		2.5x - 90		160	ns
31	t_{RAH}	$\overline{RAS}$ fall→A0 - 15 Hold	0.5x - 25		25		ns
32	t_{RAS}	$\overline{RAS}$ Low width	2.0x - 40		160		ns
33	t_{RP}	$\overline{RAS}$ High width	2.0x - 40		160		ns
34	t_{RSH}	$\overline{CAS}$ fall→$\overline{RAS}$ rise	1.0x - 55		45		ns
35	t_{RSC}	$\overline{RAS}$ rise→$\overline{CAS}$ rise	0.5x - 25		25		ns
36	t_{RCD}	$\overline{RAS}$ fall→$\overline{CAS}$ fall	1.0x - 40		60		ns
37	t_{CAC}	$\overline{CAS}$ fall→D0 - 15 input		1.5x - 120		30	ns
38	t_{CAS}	$\overline{CAS}$ Low width	1.5x - 30		120		ns

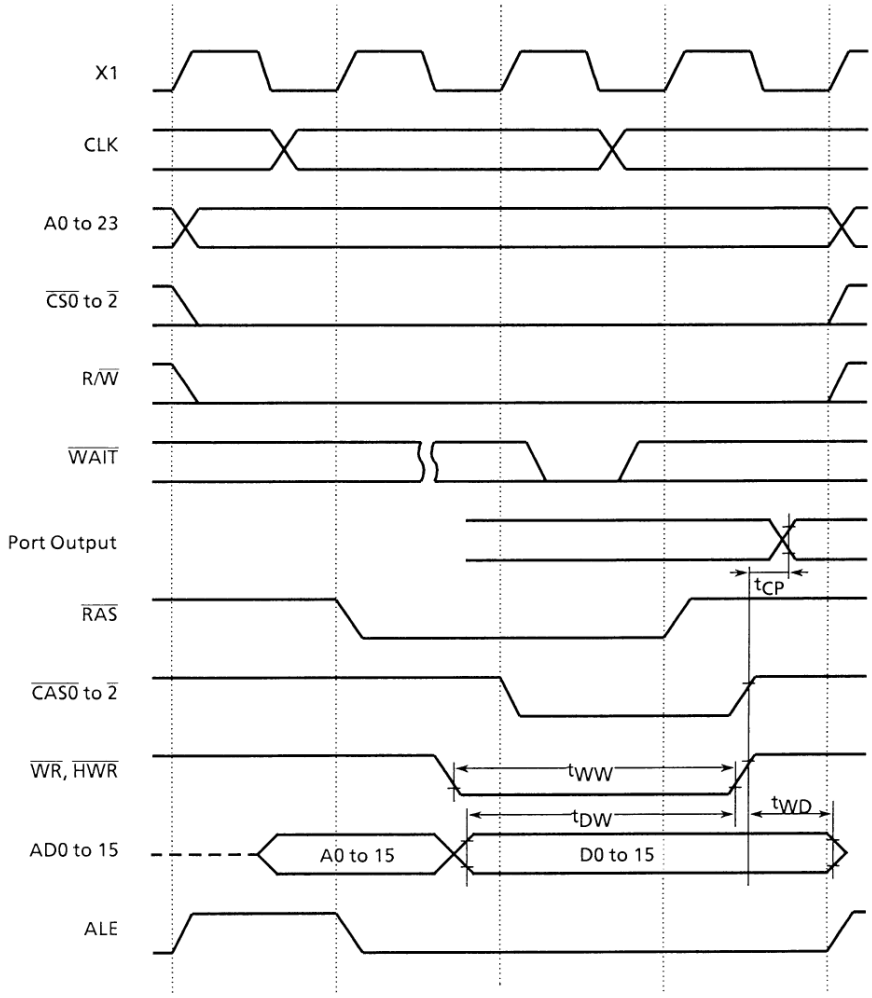
AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF
  (However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, CLK, $\overline{RAS}$, $\overline{CAS0}$ ~ $\overline{CAS2}$)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)
  High 0.8Vcc /Low 0.2Vcc (Except for AD0 ~ AD15)

(1) Read Cycle


tOSC
X1
tCLK
CLK
tAK
tKA
A0 to 23
CS0 to 2
R/W
tAWH
tCW
tAWL
WAIT
tAPH
tAPH2
Port Input
tASRH
tRSH
tRP
RAS
tRAS
tASRL
tRSC
tRAC
tRAH
tCAS
CAS0 to 2
tRCD
tCAC
tADH
tCA
RD
tACH
tRR
tRAE
tACL
tHR
tLC
tRD
tADL
AD0 to 15
A0 to 15
D0 to 15
tAL
tLA
tCL
ALE
tLL

(2) Write Cycle



4.4 A/D Conversion Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{REF}	Analog reference voltage	V_{CC} - 0.2V	V_{CC}	V_{CC}	V
A_{GND}	Analog reference voltage	V_{SS}	V_{SS}	V_{SS} + 0.2V	
V_{AIN}	Analog input voltage range	V_{SS}		V_{CC}	
I_{REF} (V_{REFL} = 0V)	Analog current for analog reference voltage VCC = 5V±10% <VREFON> = 1		0.5	1.5	mA
	VCC = 5V±10% <VREFON> = 0				
Error	VCC = 5V±10%		±3.0	±6	LSB

4.5 Serial Channel Timing - I/O Interface Mode

(1) SCLK Input Mode

Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	16x		1.6		0.8		μs
t_{OSS}	Output Data→rising edge of SCLK	t_{SCY}/2 - 5x - 50		250		100		ns
t_{OHS}	SCLK rising edge→output data hold	5x - 100		400		150		ns
t_{HSR}	SCLK rising edge→input data hold	0		0		0		ns
t_{SRD}	SCLK rising edge→effective data input		t_{SCY} - 5x - 100		1000		450	ns

(2) SCLK Output Mode

Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (programmable)	16x	8192x	1.6	819.2	0.8	409.6	μs
t_{OSS}	Output Data→rising edge of SCLK	t_{SCY} - 2x - 150		1250		550		ns
t_{OHS}	SCLK rising edge→output data hold	2x - 80		120		20		ns
t_{HSR}	SCLK rising edge→input data hold	0		0		0		ns
t_{SRD}	SCLK rising edge→effective data input		t_{SCY} - 2x - 150		1250		550	ns

4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7)

Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock cycle	8x + 100		900		500		ns
t_{VCKL}	Low level clock pulse width	4x + 40		440		240		ns
t_{VCKH}	High level clock pulse width	4x + 40		440		240		ns

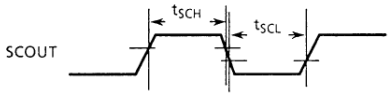
4.7 Interrupt Operation

Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{NMI}$, INT0 Low level pulse width	4x		400		200		ns
t_{INTAH}	$\overline{NMI}$, INT0 High level pulse width	4x		400		200		ns
t_{INTBL}	INT4 ~ INT7 Low level pulse width	8x + 100		900		500		ns
t_{INTBH}	INT4 ~ INT7 High level pulse width	8x + 100		900		500		ns

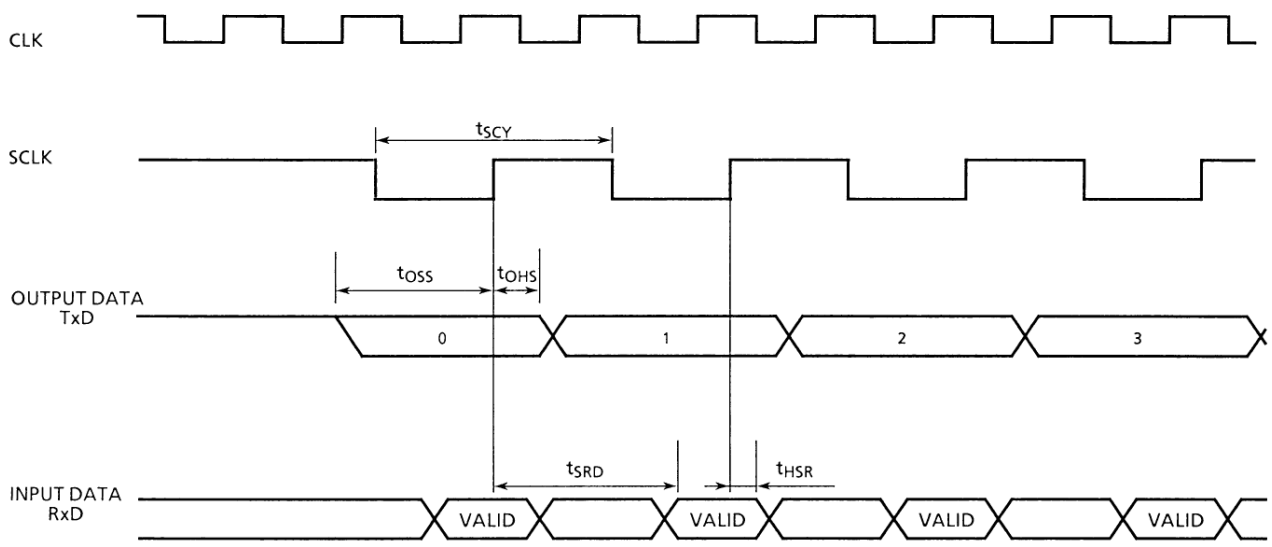
4.8 SCOUT Pin Characteristics

Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
	High level pulse width VCC = 5V±10%	5x - 10		40		15		ns
	↑ VCC = 3V±10%	5x - 20		30		–	–	
	Low level pulse width VCC = 5V±10%	5x - 10		40		15		ns
	↑ VCC = 3V±10%	5x - 20		30		–	–	

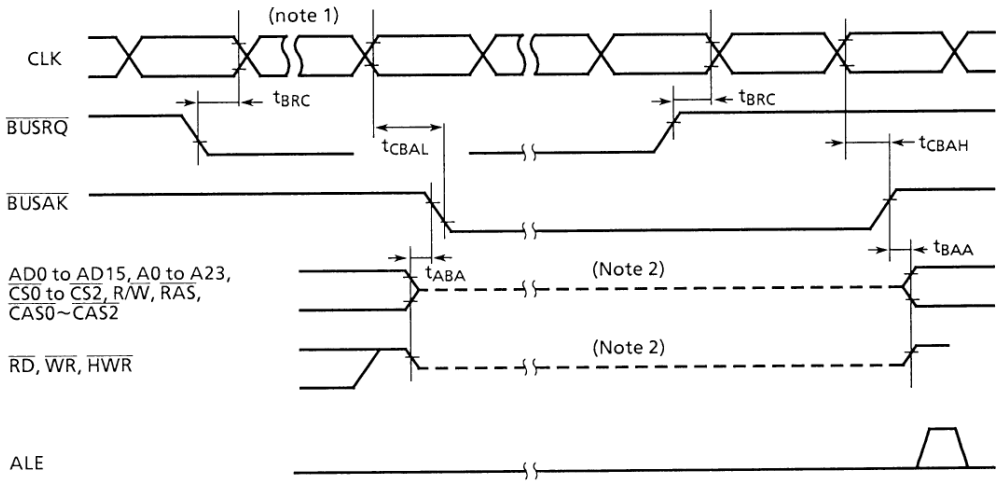
Measurement condition
- Output level : High 2.2 V / Low 0.8 V, CL = 10 pF



4.9 Timing Chart for I/O Interface Mode



4.10 Timing Chart for Bus Request ($\overline{BUSRQ}$)/BUS Acknowledge ($\overline{BUSAK}$)



Symbol	Parameter	Variable		10MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{BRC}	$\overline{BUSRQ}$ setup time for CLK	120		120		120		ns
t_{CBAL}	CLK→$\overline{BUSAK}$ falling edge		1.5x + 120		270		195	ns
t_{CBAH}	CLK→$\overline{BUSAK}$ rising edge		0.5x + 40		90		65	ns
t_{ABA}	Output buffer is off to $\overline{BUSAK}$	0	80	0	80	0	80	ns
t_{BAA}	$\overline{BUSAK}$ output buffer is on.	0	80	0	80	0	80	ns

Note 1: The Bus will be released after the $\overline{WAIT}$ request is inactive, when the $\overline{BUSRQ}$ is set to "0" during "Wait" cycle.

Note 2: This line only shows the output buffer is off-states. They don't indicate the signal levels are fixed. After the bus is released, the signal level is kept dynamically before the bus is released by the external capacitance. Therefore, to fix the signal level by an external resistance under the bus is releasing, the design must be carefully because of the level-fix will be delayed. The internal programmable pull-up/pull-down resistance is switched active by the internal signal.

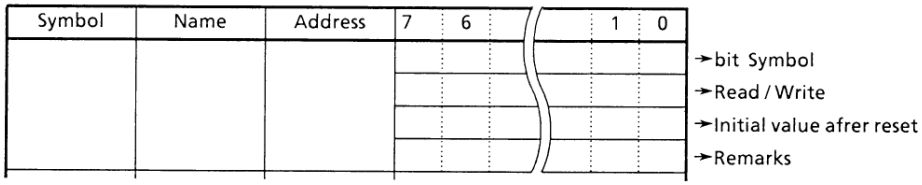
5. Table of Special Function Registers (SFRs)

(SFR; Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 128-byte addresses from 000000H to 00007FH.

(1) I/O port
(2) I/O port control
(3) Timer control
(4) Pattern Generator control
(5) Watch Dog Timer control
(6) Serial Channel control
(7) A/D converter control
(8) Interrupt control
(9) Chip Select/Wait Control
(10) Clock Control

Configuration of the table



Note: "Prohibit "RMW" in table means that you cannot use RMW instructions to these registers.

(Example) Setting only the bit0 of register P0CR, do not use "Set 0, (0002H)".

Table 5 I/O Register Address Map

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
000000H	P0	20H	TRUN	40H	TREG6L	60H	ADREG04L
1H	P1	21H		41H	TREG6H	61H	ADREG04H
2H	P0CR	22H	TREG0	42H	TREG7L	62H	ADREG15L
3H		23H	TREG1	43H	TREG7H	63H	ADREG15H
4H	P1CR	24H	TMOD	44H	CAP3L	64H	ADREG26L
5H	P1FC	25H	TFFCR	45H	CAP3H	65H	ADREG26H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG37L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG37H
8H	P2CR	28H	P0MOD	48H	T5MOD	68H	B0CS
9H	P2FC	29H	P1MOD	49H	T5FFCR	69H	B1CS
AH	P3CR	2AH	PFFCR	4AH		6AH	B2CS
BH	P3FC	2BH		4BH		6BH	
CH	P4	2CH		4CH	PG0REG	6CH	
DH	P5	2DH		4DH	PG1REG	6DH	CKOCR
EH	P4CR	2EH		4EH	PG01CR	6EH	SYSCR0
FH		2FH		4FH		6FH	SYSCR1
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H		31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTEPW10
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	P7FC	37H	CAP2H	57H	BR1CR	77H	INTES0
18H	P8	38H	T4MOD	58H	ODE	78H	INTES1
19H	P9	39H	T4FFCR	59H		79H	
1AH	P8CR	3AH	T45CR	5AH		7AH	
1BH	P9CR	3BH		5BH		7BH	IIMC
1CH	P8FC	3CH		5CH	WDMOD	7CH	DMA0V
1DH	P9FC	3DH		5DH	WDCR	7DH	DMA1V
1EH	PA	3EH		5EH	ADMOD1	7EH	DMA2V
1FH	PACR	3FH		5FH	ADMOD2	7FH	DMA3V

(1) I/O Port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Input mode							
			Underfined							
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode							
			0	0	0	0	0	0	0	0
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
			0	0	0	0	0	0	0	0
P3	PORT3	07H	P37	P36	P35	P34	P33	P32	P31	P30(note1)
			R/W							
			Input mode						Output mode	
			1	1	1	1	1	1	1	1
P4	PORT4	0CH						P42	P41	P40
								R/W		
								Input mode		
								0	1	1
P5	PORT5	0DH	P57	P56	P55	P54	P53	P52	P51	P50
			R							
			Input mode							
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
P7	PORT7	13H					P73	P72	P71	P70
							R/W			
							Input mode			
							1	1	1	1
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
P9	PORT9 (note2)	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W	R/W	R/W					
			Output mode	Output mode	Input mode					
			1	1	1	1	1	1	1	1
PA	PORTA	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			Input mode							
			1							

Note1 : 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.

Note2 : Port96, 97 is also used as XT1, XT2. Therefore these pins are open drain output type.

Read / Write

R/W ; Either read or write is possible

R ; Only read is possible

W ; Only write is possible

Prohibit RMW ; Prohibit Read Modify Write. (Prohibit RES / SET / TSET / CHG / STCF / ANDCF / ORCF / XORCF Instruction)

(2) I/O Port Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	PORT0 Control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0 : IN 1 : OUT (When external access, set as AD7-0 and cleared to "0".)							
P1CR	PORT1 Control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P1FC">>							
P1FC	PORT1 Function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00 : IN, 01 : OUT, 10 : AD15-8, 11 : A15-8							
P2CR	PORT2 Control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P2FC">>							
P2FC	PORT2 Function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00 : IN, 01 : OUT, 10 : A7-0, 11 : A23-16							
P3CR	PORT3 Control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C		
			W							
			0	0	0	0	0	0		
			0 : IN			1 : OUT				
P3FC	PORT3 Function	0BH (Prohibit RMW)	P37F	P36F	P35F	P34F		P32F	P31F	P30F
			W							
			0	0	0	0		0	0	0
			0 : PORT 1 : $\overline{RAS}$	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	PORT4 Control	0EH (Prohibit RMW)						P42C	P41C	P40C
									W	
								0	0	0
								0 : IN	1 : OUT	
P4FC	PORT4 Function	10H (Prohibit RMW)						P42F	P41F	P40F
									W	
								0	0	0
								0 : PORT	1 : $\overline{CS}$/$\overline{CAS}$	

Note : With the TMP93CM41, which requires an external ROM, PORT0 functions as AD0 to AD7; PORT1, AD8 to AD15 or A8 to A15; P30, the $\overline{RD}$ signal; P31, the $\overline{WR}$ signal, regardless of the values set in P0CR, P1CR, P1FC, P30F and P31F.

(2) I/O Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	PORT6 Control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0 : IN				1 : OUT			
P7CR	PORT7 Control	15H (Prohibit RMW)					P73C	P72C	P71C	P70C
							W			
							0	0	0	0
							0 : IN		1 : OUT	
P6FC	PORT6 Function	16H (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0	0	0	0	0	0	0	0
			0 : PORT		1 : PG1-OUT		0 : PORT		1 : PG0-OUT	
P7FC	PORT7 Function	17H (Prohibit RMW)					P73F	P72F	P71F	
								W		
							0	0	0	
							0 : PORT 1 : TO3	0 : PORT 1 : TO2	0 : PORT 1 : TO1	
P8CR	PORT8 Control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0 : IN				1 : OUT			
P9CR	PORT9 Control	1BH (Prohibit RMW)	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W	W			W			
			1	1	0	0	0	0	0	0
			0 : IN				1 : OUT			
P8FC	PORT8 Function	1CH (Prohibit RMW)		P86F			P83F	P82F		
				W			W	W		
				0			0	0		
				0 : PORT 1 : TO6			0 : PORT 1 : TO5	0 : PORT 1 : TO4		
P9FC	PORT9 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	PORTA Control	1FH (Prohibit RMW)	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0							
			0 : IN				1 : OUT			

(3) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
			Prescaler & Timer Run / Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							
TREG0	8 bit Timer Register 0	22H (Prohibit RMW)	–							
			W							
			Undefined							
TREG1	8 bit Timer Register 1	23H (Prohibit RMW)	–							
			W							
			Undefined							
TMOD	8 bit Timer Source CLK & MODE	24H (Prohibit RMW)	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			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 : 2^6-1 10 : 2^7-1 11 : 2^8-1 PWM		00 : TO0TRG 01 : φT1 10 : φT16 11 : φT256		00 : TI0 入力 01 : φT1 10 : φT4 11 : φT16	
TFFCR	8 bit Timer Flip-Flop Control	25H				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS
						R/W	W		R/W	
						0	0	0	0	0
						1 : Double Buffer Enable	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	0: Inverted by Timer 0
TREG2	PWM Timer Register 2	26H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
TREG3	PWM Timer Register 3	27H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
P0MOD	PWM0 Mode	28H (Prohibit RMW)	FF2RD	DB2EN	PWM0INT	PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0
			R		W					
			–	0	0	0	0	0	0	0
			TFF2 output value	1 : Double Buffer Enable	0 : Overflow interrupt 1 : Compare / match interrupt	0 : PWM Mode 1 : Timer Mode	00 : φP1 01 : φP4 10 : φP16 11 : Don't care		00 : 2^6-1 01 : 2^7-1 10 : 2^8-1 11 : Don't care	
P1MOD	PWM1 Mode	29H (Prohibit RMW)	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0
			R		W					
			–	0	0	0	0	0	0	0
			TFF3 output value	1 : Double Buffer Enable	0 : Overflow interrupt 1 : Compare / match interrupt	0 : PWM Mode 1 : Timer Mode	00 : φP1 01 : φP4 10 : φP16 11 : Don't care		00 : 2^6-1 01 : 2^7-1 10 : 2^8-1 11 : Don't care	

(3) Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PFFCR	PWM Flip-Flop Control	2AH	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
			W		R/W		W		R/W	
			0	0	0	0	0	0	0	0
			00 : Don't care 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		00 : Prohibit TFF3 invert 01 : Invert if matched 10 : Set if matched; clear if overflowed 11 : Clear if matched; set if overflowed		00 : Don't care 01 : Set TFF2 10 : Clear TFF2 11 : Don't care		00: Prohibit TFF2 invert 01 : Invert if matched 10 : Set if matched; clear if overflowed 11 : Clear if matched; set if overflowed	
TREG4L	16 bit Timer Register4L	30H (Prohibit RMW)	–							
			W							
			Undefined							
TREG4H	16 bit Timer Register4H	31H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5L	16 bit Timer Register5L	32H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5H	16 bit Timer Register5H	33H (Prohibit RMW)	–							
			W							
			Undefined							
CAP1L	Capture Register1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register2H	37H	–							
			R							
			Undefined							
T4MOD	16 bit Timer 4 Source CLK & MODE	38H	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
			R/W		W			R/W		
			0	0	0	0	0	0	0	0
			TFF5 INV TRG 0 : TRG Disable 1 : TRG Enable		0 : Soft-Capture 1 : Don't care		Capture Timming 00 : Disable 01 : TI4 ↑ TI5 ↑ 10 : TI4 ↑ TI4 ↓ 11 : TFF1 ↑ TFF1 ↓	1 : UC4 Clear Enable	Source Clock 00 : TI4 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T4FFCR	16bit Timer 4 Flip-Flop Control	39H	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
			W		R/W				W	
			0	0	0	0	0	0	0	0
			00 : Invert TFF5 01 : Set TFF5 10 : Clear TFF5 11 : Don't care		TFF4 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF4 01 : Set TFF4 10 : Clear TFF4 11 : Don't care	

(3) Timer Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T45CR	T4, T5 Control	3AH	–				PG1T	PG0T	DB6EN	DB4EN
			R/W				R/W			
			0				0	0	0	0
			Fix at "0"				PG1 shift trigger 0 : Timer 0, 1 1 : Timer 5	PG0 shift trigger 0 : Timer 0, 1 1 : Timer 4	1 : Double Buffer Enable	
TREG6L	16 bit Timer Register6L	40H (Prohibit RMW)	–							
			W							
			Undefined							
TREG6H	16 bit Timer Register6H	41H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7L	16 bit Timer Register7L	42H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7H	16 bit Timer Register7H	43H (Prohibit RMW)	–							
			W							
			Undefined							
CAP3L	Capture Register3L	44H	–							
			R							
			Undefined							
CAP3H	Capture Register3H	45H	–							
			R							
			Undefined							
CAP4L	Capture Register4L	46H	–							
			R							
			Undefined							
CAP4H	Capture Register4H	47H	–							
			R							
			Undefined							
T5MOD	16 bit Timer 5 Source CLK & MODE	48H			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
					W	R/W				
					0	0	0	0	0	0
					0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI6 ↑ TI7 ↑ 10 : TI6 ↑ TI6 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC5 Clear Enable	Source Clock 00 : TI6 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T5FFCR	16 bit Timer 5 Flip-Flop Control	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	0	0
					TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(4) Pattern Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG0REG	PG0 Register	4CH (Prohibit RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0	0	0	0	Undefined			
PG1REG	PG1 Register	4DH (Prohibit RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			W				R/W			
			0	0	0	0	Undefined			
PG01CR	PG0, 1 Contorol	4EH	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0	0	0	0	0	0	0	0
			0: 8 bit write 1: 4 bit write	0: Normal Rotation 1: Reverse Rotation	0: 4 bit Step 1: 8 bit Step	PG1 trigger input enable 1: Enable	0: 8 bit write 1: 4 bit write	0: Normal Rotation 1: Reverse Rotation	0: 4 bit Step 1: 8 bit Step	PG0 trigger input enable 1: Enable

(5) Watch Dog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode	5CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT Enable	00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warming up Time 0: 2^{14}/inputted frequency 1: 2^{16}/inputted frequency	Standby Mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Connect internally WDT out to Reset Pin	1: Drive the pin in STOP mode
WDCR	Watch Dog Timer Control Register	5DH	–							
			W							
			–							
			B1H: WDT Disable Code 4EH: WDT Clear Code							

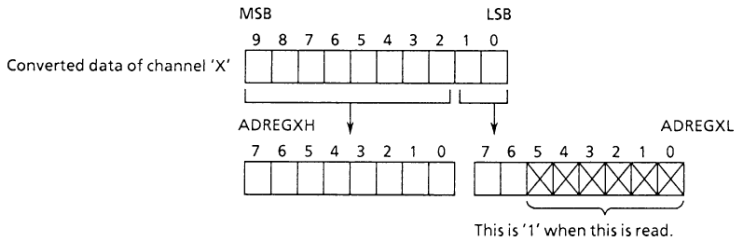
(6) Serial Channel

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	50H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) / W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	51H	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 Overrun	Parity	Framing	0: SCLK0 1: SCLK0	1: Input SCLK0 pin
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			undefined	0	0	0	0	0	0	0
			Transmisson data bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO0 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	
BR0CR	Baud Rate Control	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W			R/W				
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F ("1" prohibited)			
SC1BUF	Serial Channel 1 Buffen	54H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) / W (Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control	55H	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 Overrun	Parity	Framing	0: SCLK1 1: SCLK1	1: Input SCLK1 pin
SC1-MOD	Serial Channel 1 Mode	56H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmisson data bit 8	Fix at "0"	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO0 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	
BR1CR	Baud Rate Control	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W			R/W				
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F ("1" prohibited)			
ODE	Serial Open Drain Enable	58H							ODE1	ODE0
									R/W	
									0	0
									1:P93 Open-drain	1:P90 Open-drain

(7) A/D Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD 1	A/D Mode Reg 1	5EH	EOCF	ADBF	RPT	SCAN		ADS		
			R		R/W			R/W		
			0	0	0	0		0		
			1: End	1: Busy	1: Repeat	1: Scan		1: Start		
ADMOD 2	A/D Mode Reg 2	5FH	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
			R/W		R/W			R/W		
			1		0	0		0	0	0
			Ladder Resistance Switch ON/OFF		SPEED			Analog Input Channel Select		
AD REG04L	*1) AD Result Reg 0/4 low	60H	ADR01	ADR00						
			R							
			Undefined		1	1	1	1	1	1
AD REG04H	AD Result Reg 0 / 4 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG15L	*1) AD Result Reg 1 / 5 low	62H	ADR11	ADR10						
			R							
			Undefined		1	1	1	1	1	1
AD REG15H	AD Result Reg 1 / 5 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG26L	*1) AD Result Reg 2 / 6 low	64H	ADR21	ADR20						
			R							
			Undefined		1	1	1	1	1	1
AD REG26H	AD Result Reg 2 / 6 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG37L	*1) AD Result Reg 3 / 7 low	66H	ADR31	ADR30						
			R							
			Undefined		1	1	1	1	1	1
AD REG37H	AD Result Reg 3 / 7 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

*1 : Data to be stored in A/D Conversion Result Reg Low are the lower 2 bits of the conversion result. The contents of the lower 6 bits of this register are always read as "1".



(8) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	INTerrupt Enable 0 & A/D	70H (Prohibit RMW)	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4 / 5	71H (Prohibit RMW)	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE67	INTerrupt Enable 6 / 7	72H (Prohibit RMW)	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET10	INTerrupt Enable Timer 1 / 0	73H (Prohibit RMW)	INTT1 (Timer 1)				INTT0 (Timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE-PW10	INTerrupt Enable PWm 1 / 0	74H (Prohibit RMW)	INTT3 (Timer 3/PWM1)				INTT2 (Timer 2/PWM0)			
			IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET54	INTerrupt Enable Treg 5 / 4	75H (Prohibit RMW)	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET76	INTerrupt Enable Treg 7 / 6	76H (Prohibit RMW)	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	
INTES0	INTerrupt Enable Serial 0	77H (Prohibit RMW)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES1	INTerrupt Enable Serial 1	78H (Prohibit RMW)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Prohibit interrupt request.
0	0	1	Set interrupt request level to "1".
0	1	0	Set interrupt request level to "2".
0	1	1	Set interrupt request level to "3".
1	0	0	Set interrupt request level to "4".
1	0	1	Set interrupt request level to "5".
1	1	0	Set interrupt request level to "6".
1	1	1	Prohibit interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicate no interrupt request.	Clear interrupt request flag.
1	Indicate interrupt request.	----- Don't care -----

(8) Interrupt Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request Vector	7CH (Prohibit RMW)				μDMA0 start vector				
						DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
						W				
						0	0	0	0	0
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)				μDMA1 start vector				
						DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
						W				
						0	0	0	0	0
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)				μDMA2 start vector				
						DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
						W				
						0	0	0	0	0
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)				μDMA3 start vector				
						DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
						W				
						0	0	0	0	0
IIMC	Interrupt Input Mode Contorol	7BH (Prohibit RMW)						I0IE	I0LE	NMIREE
								W	W	W
								0	0	0
								1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Operate even at $\overline{\text{NMI}}$ rise edge

(9) Chip Select/Controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS / WAIT control register	68H (Prohibit RMW)	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B0CS Master bit		0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16 bit Bus 1: 8 bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 7F00H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B1CS	Block 1 CS / WAIT control register	69H (Prohibit RMW)	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B1CS Master bit		0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16 bit Bus 1: 8 bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 880H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B2CS	Block 2 CS / WAIT control register	6AH (Prohibit RMW)	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1: B2CS Master bit		0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16 bit Bus 1: 8 bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 8000H to 01: 400000H to 10: 800000H to 11: C00000H to	

Note 1 : After reset, only "Block 2" is set to enable.

(10) Clock Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CKOCR	Clock Output Control Register	006DH					SCOSEL	SCOEN	ALEEN	CLKEN
							R/W			
							0	0	0/1 note1)	0/1 note1)
							SCOUT select 0 : $f_{SYS} \times 2$ clock 1 : f_{SYS} clock	SCOUT Output control 0 : I/O port 1 : SCOUT output	ALE pin control 0 : HZ output 1 : ALE output	CLK pin control 0 : HZ output 1 : CLK output
SYSCR0	System Clock Control Register 0	006EH	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	Warming Up Timer 0 write : don't care 1 write : start timer 0 read : end warming up 1 read : not end warming up	select prescaler clock 00 : f_{FPH} 01 : fs 10 : fc/16 11 : (reserved)	
SYSCR1	System Clock Control Register 1	006FH					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							select system clock 0 : fc 1 : fs note2)	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)		

(note 1) The value after reset of <CLKEN>, <ALEEN> is following :

TMP93CM40 : "0" (High impedance output)

TMP93CM41 : "1" (CLK or ALE output)

But during reset, CLK pin is pulled up internally regardless of the products.

(note2) The high frequency oscillator will be enabled regardless the value of SYSCR0<XEN> when SYSCR1<SYSCK> is set to "0".

On the other hand, the low frequency oscillator will be enabled regardless the value of SYSCR0<XTEN> when SYSCR1<SYSCK> is set to "1".

6. Port Section Equivalent Circuit Diagram

- Reading The Circuit Diagram

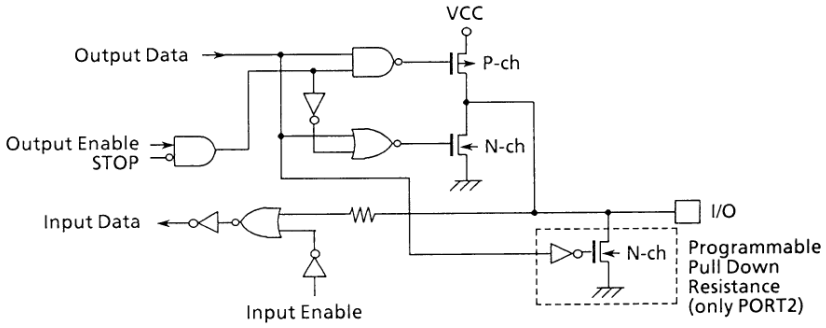
  Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

  The dedicated signal is described below.

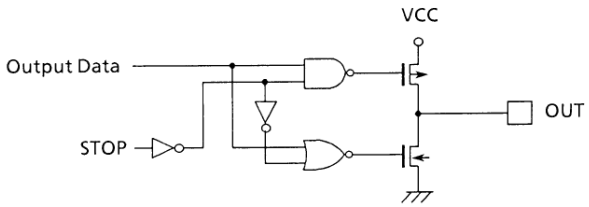
STOP: This signal becomes active “1” when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to “1”, however, STP remains at “0”.

- The input protection resistor ranges from several tens of ohms to several hundreds of ohms.

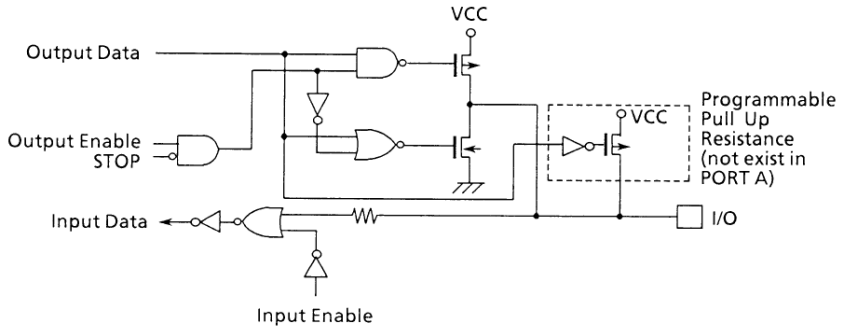
- PO (AD0 to AD7), P1 (AD8 to 15, A8 to 15), P2 (A16 to 23, A0 to 7)



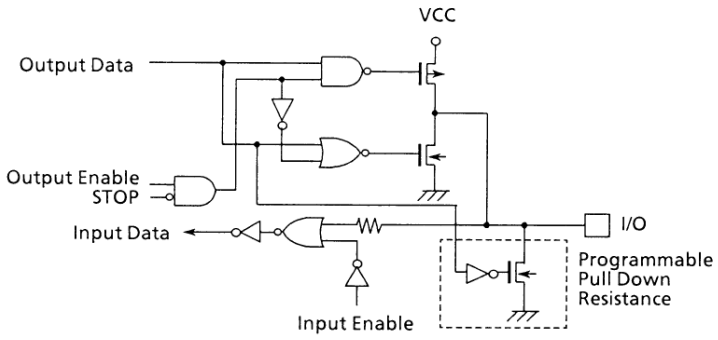
- P30 ($\overline{RD}$), P31 ($\overline{WR}$)



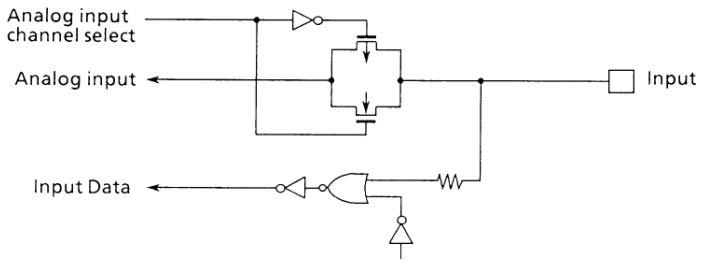
- P32 to 37, P40 to 41, P6, P7, P80 to 86, P91 to 92, P94 to 95, PA



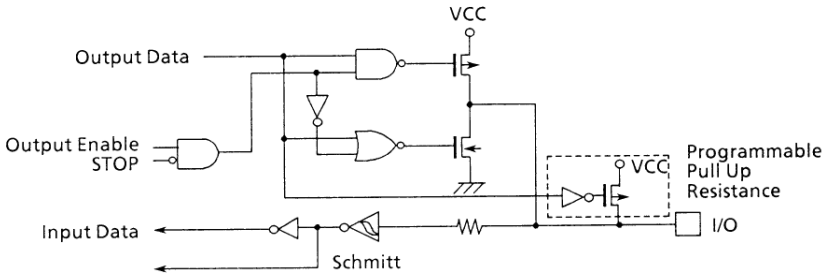
- P42 ($\overline{CS2}$, $\overline{CAS2}$)



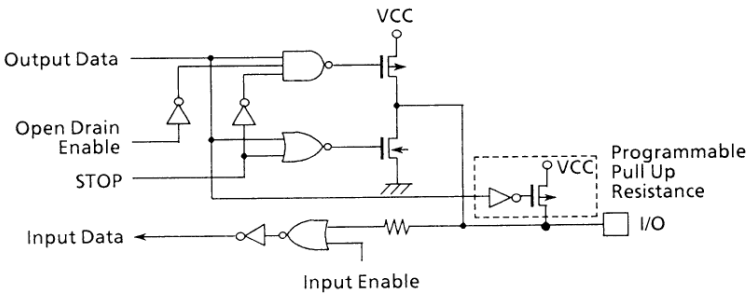
- P5 (AN0 to 7)



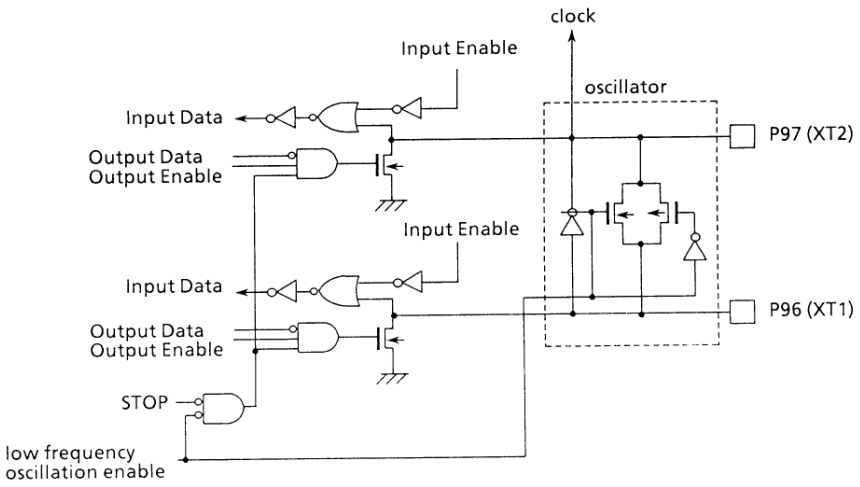
- P87 (INT0)



- P90 (TXD0), P93 (TXD1)

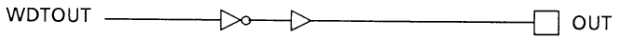


- P96 (XT1), P97 (XT2)

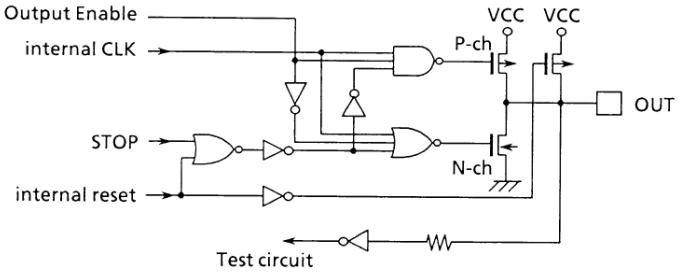


- $\overline{\text{NMI}}$

- $\overline{\text{WDTOUT}}$

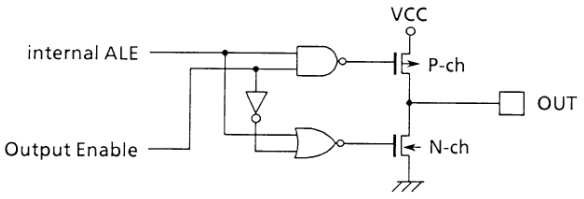


- CLK

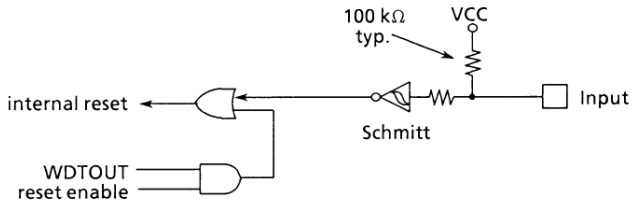


- $\overline{\text{EA}}$, AM8/$\overline{16}$

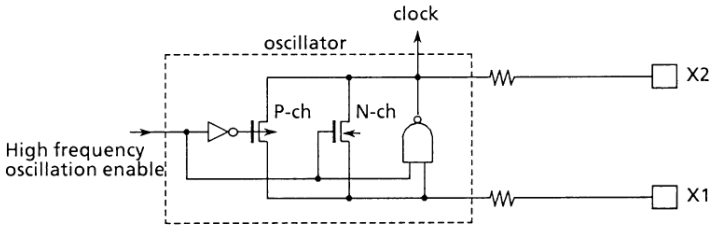
• ALE



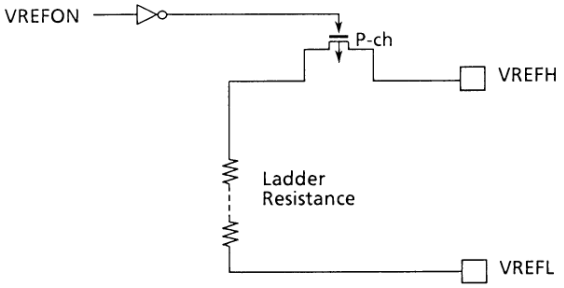
• $\overline{\text{RESET}}$



• X1, X2



• VREF, AGND



7. Care Points and Restriction

(1) Special Expression

① Explanation of a built-in I/O register: Register

Symbol <Bit Symbol>
ex) TRUN <T0RUN> · · · Bit T0RUN of Register TRUN

② Read, Modify and Write Instruction

An instruction which CPU executes following by one instruction.

1. CPU reads data of the memory.

2. CPU modifies the data.

3. CPU writes the data to the same memory.

ex1) SET 3, (TRUN) · · · set bit3 of TRUN
ex2) INC1, (100H) increment the data of 100H

- The representative Read, Modify and Write Instruction in the TLCS-900

SET	imm, mem,	RES	imm, mem
CHG	imm, mem,	TSET	imm, mem
INC	imm, mem,	DEC	imm, mem
RLD	A, mem,	ADD	imm, reg

③ 1 state

1 cycle clock divided by 2 oscillation frequency is called 1 state

ex) The case of oscillation frequency is 20MHz.

(2) Care Points

① $\overline{EA}$, AM8/$\overline{16}$ pin

Fix these pins VCC or GND unless changing voltage.

② TEST2, TEST2 pin

Connect TEST1 pin with TEST2 pin.

③ Reserved Area in Memory Space

The 256 byte memory area for FFFF00H to FFFFFFH cannot be used because it is a reserved area.

④ Standby Mode (IDLE1)

When IDEL1 mode (operates only as an oscillator) is used, set TRUN <PRRUN> to "0" to stop prescaler before "HALT" instruction is executed.

⑤ Warming-up Counter

The warming-up counter operates when the STOP mode. is released even the system which is used an external oscillator. As a result, it takes warming up time from inputting the releasing request to outputting the system clock.

⑥ High Speed μDMA (DRAM refresh mode)

When the bus is released ($\overline{BUSAK}$ = "0") for waiting to accept the interrupt, DRAM refresh is not performed because of the high speed μDMA is generated by an interrupt.

⑦ Programmable Pull Up/Down Resistance

The programmable pull up/down resistors can be selected ON/OFF by program when they are used as the input ports. The case of they are used as the output ports, they cannot be selected ON/OFF by program.

⑧ 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.

⑨ Watch Dog Timer

The watch dog timer starts operation immediately after the reset is released. When the watch dog timer is not used, set watch dog timer to disable.

⑩ Watch Dog Timer

When the bus is released, both internal memory and internal I/O cannot be accessed. But internal I/O continues to operate. So, the watch dog timer continues to run. Therefore, be careful with the bus releasing time and set the detection timer of watch dog timer.

⑪ A/D Converter

The ladder resistor between CREFH and VREFL pins can be cut by program to reduce the power consumption. When the standby mode is used, cut by program before “HALT” instruction is used.

⑫ CPU (High SpeedμDMA)

Only the “LDC cr, r”, “LDC r, cr” instruction can be used to access the control register like transfer source address register (DMASn) in the CPU.

8. TMP93XX40/41 Different Points

	93CM40F	93CM41F	93CM40AF	93CM41AF	93CS40F	93PS40F
Built-in ROM	32 K byte Mask ROM (8000H-FFFFH)	None	32 K byte Mask ROM (8000H-FFFFH)	None	64 K byte Mask ROM (8000H-17FFFH)	64 K byte OTP (8000H-17FFFH)
Built-in RAM	2 K byte (0080H - 087FH)					
Operation frequency fc at 3V ± 10%	4 to 10 MHz		4 to 12.5 MHz			
ADC operation voltage range	5V ± 10% (4 to 20 MHz)		5V ± 10% (4 to 20 MHz) 3V ± 10% (4 to 12.5 MHz)			
CS2 Mapping area in case of <B2C1, 0> = 00	10000H to	08000H to	10000H to	08000H to	18000H to	
CS1 Mapping area in case of <B1C1, 0> = 00	880H to 7FFFH					
Port 5 Input level (V_{IL})	0.2 V_{CC}		0.3 V_{CC}			