Low Voltage / Low Power CMOS 16-bit MICROCONTROLLERS

TMP93CS44F / TMP93CS45F
TMP93CS44DF

1. OUTLINE AND DEVICE CHARACTERISTICS

The TMP93CS44 / TMP93CS45 are high-speed, advanced 16-bit microcontrollers developed for controlling medium to large-scale equipment. The TMP93CS45 does not have a ROM, the TMP93CS44 has a built-in ROM. Otherwise, the devices function in the same way.
The TMP93CS44F / TMP93CS45F are housed in 80-pin flat package (QFP80-P-1212-0.50A).
The device characteristics are as follows:

(1) Original 16-bit CPU (900 / L 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
- Micro DMA : 4 channels (1.6 μs per 2 bytes at 20 MHz)

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

(3) Internal RAM : 2K bytes

Internal ROM :

TMP93CS44	64K-byte ROM
TMP93CS45	None

(4) External memory expansion

- Can be expanded up to16M-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 bus sizing)

(5) 8-bit timer : 4 channels

(6) 16-bit timer : 2 channel

(7) Serial interface : 2 channels

(8) I2C bus channel : 1 channel

(9) 10-bit A/D converter : 8 channels

(10) High current output : 8 ports

(11) Watchdog timer

(12) Bus width / wait controller : 3 blocks

(13) Interrupt functions : 33

- 9 CPU interrupts
- 17 internal interrupts
- 7 external interrupts

7-level priority can be set.

Purchase of TOSHIBA I^{2}C components conveys a license under the Philips I^{2}C Patent Rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.

(14) I/O ports

62 pins for TMP93CS44 and 44 pins for TMP93CS45

(15) Standby function : 4 halt modes (RUN, IDLE2, IDLE1, STOP)

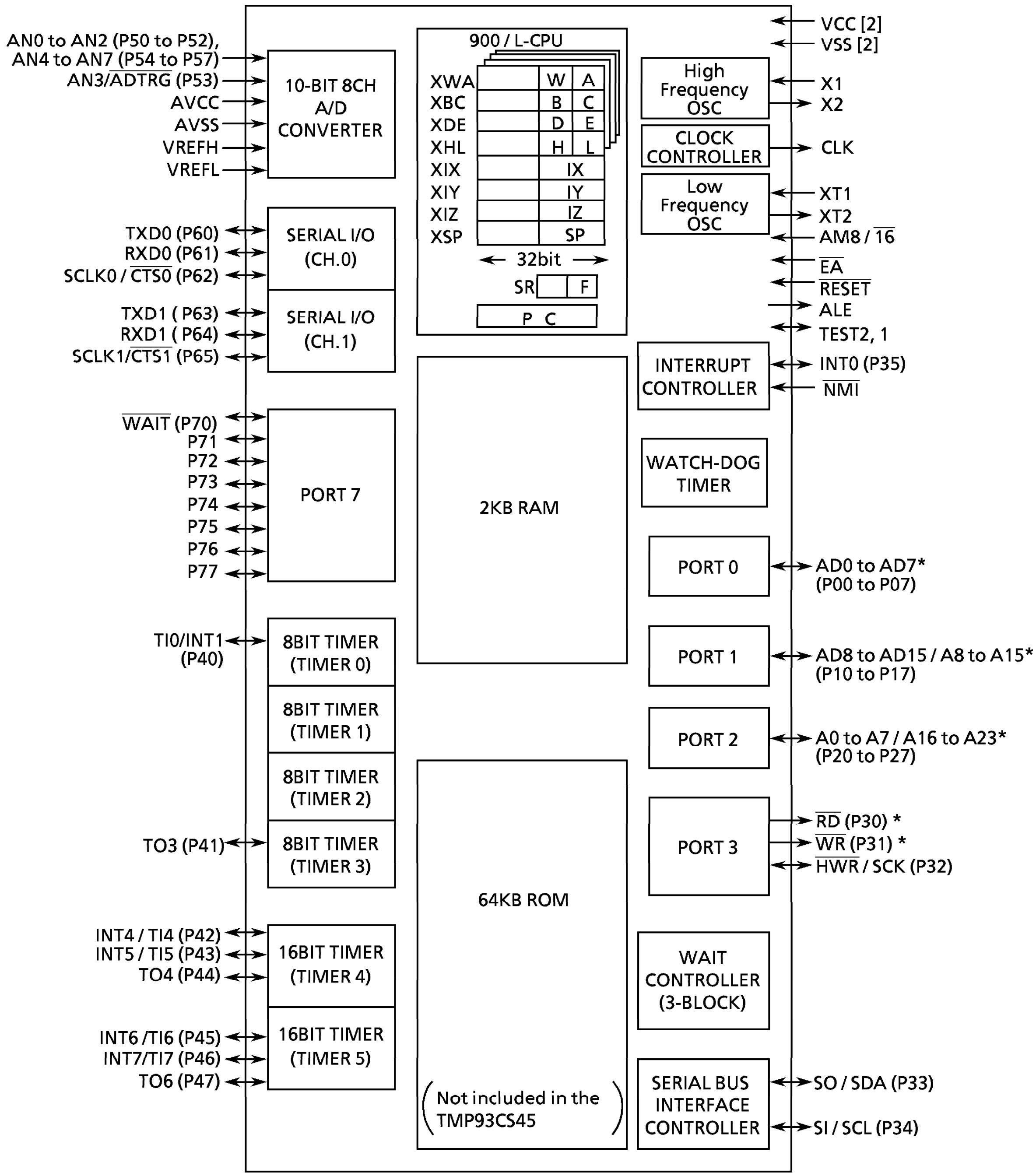
- High-frequency clock can be changed from fc to fc / 16.
- Dual clock Operation (16) Clock gear function

(17) Wide Range of Operating Voltage

- Vcc = 2.7 to 5.5 V

(18) Package

Type No.	Package
TMP93CS44F TMP93CS45F	LQFP80-P-1212-0.50A
TMP93CS44DF	QFP80-P-1420-0.80B



Note) The pin state after reset

Product	AM8/16	Pin function after reset
TMP93CS44	"H" level	Item in parentheses ( ) are the initial setting after reset.
TMP93CS45	"H" level	Except for "*" pins, item in parentheses ( ) are the initial setting after reset.
	"L" level	Except for "*" pins, item in parentheses ( ) are the initial setting after reset. However, port 1 is initialized item of out parentheses.

Figure 1 TMP93CS44/TMP93CS45 Block Diagram

2. PIN ASSIGNMENT AND FUNCTIONS

The assignment of input and output pins for the TMP93CS44 / TMP93CS45, their names and functions are described below.

2.1 Pin Assignment

Figure 2.1 shows pin assignment of the TMP93CS44F / TMP93CS45F.
Figure 2.2 shows pin assignment of the TMP93CS44DF.

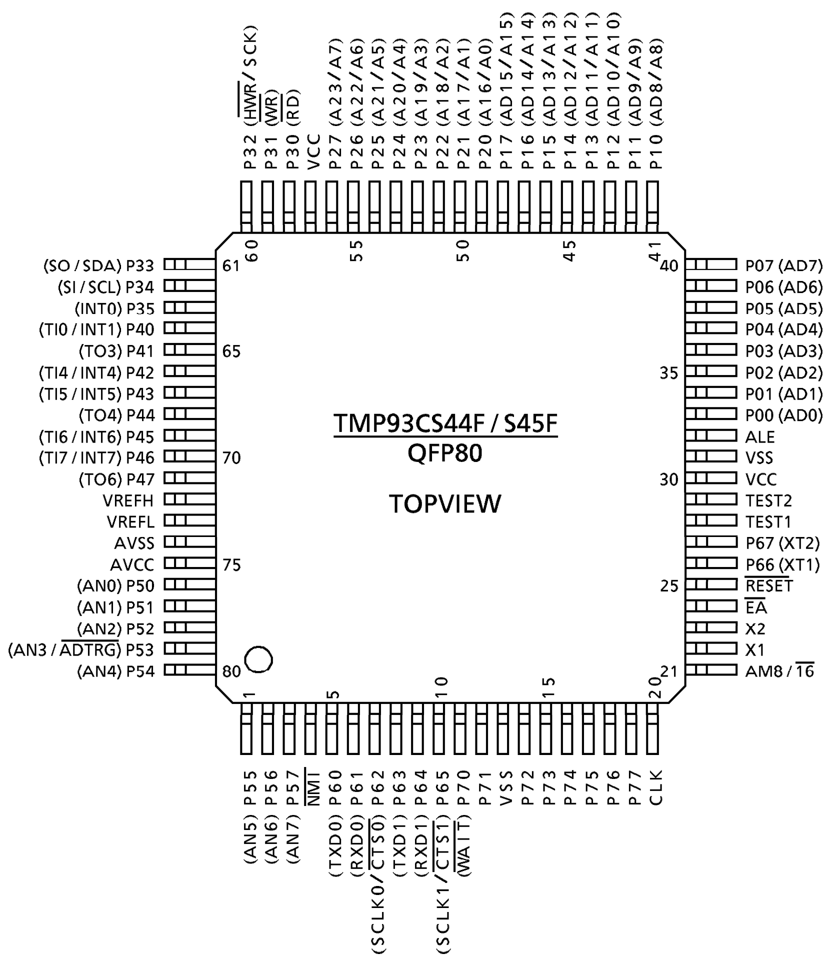


Figure 2.1 Pin Assignment (LQFP80-P-1212-0.50A)

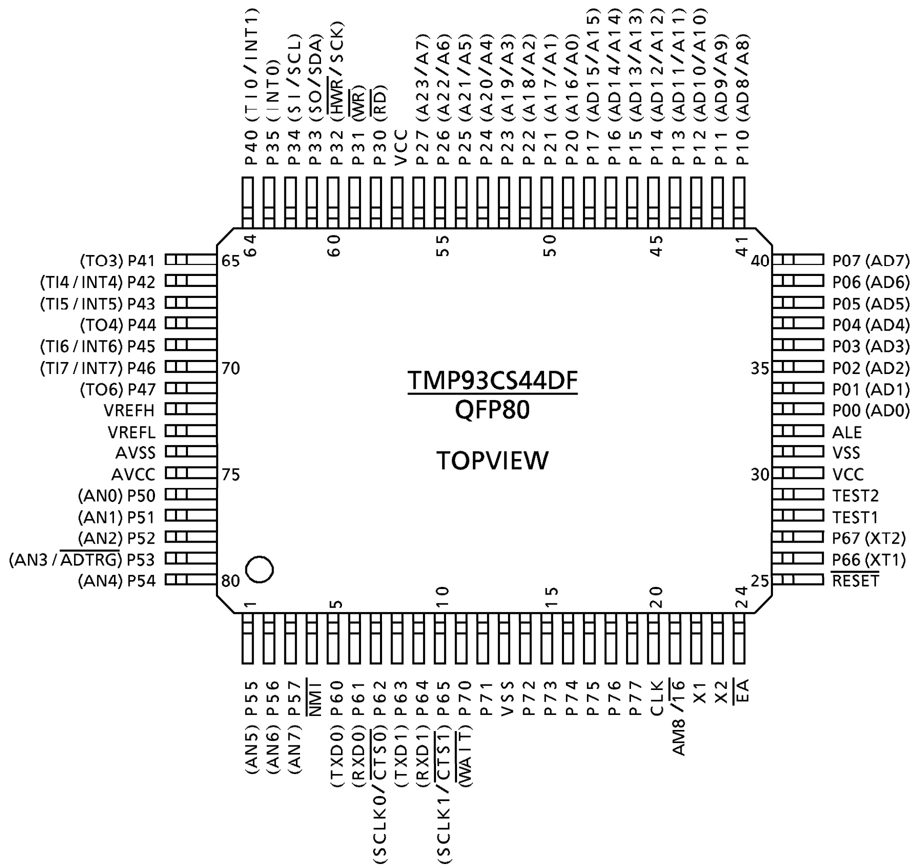


Figure 2.2 Pin Assignment (QFP80-P-1420-0.80B)

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.

Table 2.2 Pin Names and Function (1/3)

Pin name	Number of pins	I/O	Functions
P00 to P07	8	I/O	Port 0: I/O port that allows selection of I/O on a bit basis
/ AD0 to AD7		3-state	Address/data (lower): Bits 0 to 7 for address/data bus
P10 to P17	8	I/O	Port 1: I/O port that allows selection of I/O on a bit basis
/ AD8 to AD15		3-state	Address data (upper): Bits 8 to 15 for address/data bus
/ A8 to A15		Output	Address: Bits 8 to 15 for address bus
P20 to P27	8	I/O	Port 2: I/O port that allows selection of I/O on a bit basis (with pull-up resistor)
/ A0 to A7		Output	Address: Bits 0 to 7 for address bus
/ A16 to A23		Output	Address: Bits 16 to 23 for address bus
P30	1	Output	Port 30: Output port
/ $\overline{RD}$		Output	Read: Strobe signal for reading external memory
P31	1	Output	Port 31: Output port
/ $\overline{WR}$		Output	Write: Strobe signal for writing data on pins AD0 to 7
P32	1	I/O	Port 32: I/O port (with pull-up resistor)
/ $\overline{HWR}$		Output	High write: Strobe signal for writing data on pins AD8 to 15
/ SCK		I/O	Mode clock SBI SIO mode clock
P33	1	I/O	Port 33: I/O port
/ SO		Output	Serial Send Data
/ SDA		I/O	SBI I^2C bus mode channel data
P34	1	I/O	Port 34: I/O port
/ SI		Input	Serial Receive Data
/ SCL		I/O	SBI I^2C bus mode clock
P35	1	I/O	Port 35: I/O port
/ INT0		Input	Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P40	1	I/O	Port 40: I/O port
/ TI0		Input	Timer input 0: Timer 0 input
/ INT1		Input	Interrupt request pin 1: Interrupt request pin with rising edge
P41	1	I/O	Port 41: I/O port
/ TO3		Output	Timer output 3: 8-bit Timer 3 output
P42	1	I/O	Port 42: I/O port
/ TI4		Input	Timer input 4: Timer 4 input
/ INT4		Input	Interrupt request pin 4: Interrupt request pin with programmable rising / falling edge
P43	1	I/O	Port 43: I/O port
/ TI5		Input	Timer input 5: Timer 4 input
/ INT5		Input	Interrupt request pin 5: Interrupt request pin with rising edge
P44	1	I/O	Port 44 : I/O port
/ TO4		Output	Timer output 4: Timer 4 output pin

Table 2.2 Pin Names and Function (2/3)

Pin name	Number of pins	I/O	Functions
P45 / TI6 / INT6	1	I/O Input Input	Port 45: I/O port Timer input 6: Timer 5 input Interrupt request pin 6: Interrupt request pin with programmable rising / falling edge
P46 / TI7 / INT7	1	I/O Input Input	Port 46: I/O port Timer input 7: Timer 5 input Interrupt request pin 7: Interrupt request pin with rising edge
P47 / TO6	1	I/O Output	Port 47: I/O port Timer output 6: Timer 5 output pin
P50 to P52, P54 to P57 / AN0 to AN2, AN4 to AN7	7	Input Input	Port 50 to Port 52, Port 54 to Port 57: Input port Analog input: Analog signal input for A/D converter
P53 / AN3 / $\overline{\text{ADTRG}}$	1	Input Input Input	Port53: Input Port Analog input: Analog signal input for A/D converter A/D converter external start trigger input
P60 / TXD0	1	I/O Output	Port 60: I/O port (with pull-up resistor) Serial send data 0
P61 / RXD0	1	I/O Input	Port 61: I/O port (with pull-up resistor) Serial receive data 0
P62 / $\overline{\text{CTS0}}$ / SCLK0	1	I/O Input I/O	Port 62: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send) Serial Clock I/O 0
P63 / TXD1	1	I/O Output	Port 63: I/O port (with pull-up resistor) Serial send data 1
P64 / RXD1	1	I/O Input	Port 64: I/O port (with pull-up resistor) Serial receive data 1
P65 / $\overline{\text{CTS1}}$ / SCLK1	1	I/O Input I/O	Port 65: I/O port (with pull-up resistor) Serial data send enable 1 (Clear to Send) Serial clock I/O 1
P70 / $\overline{\text{WAIT}}$	1	I/O Input	Port 70: I/O port (High current output available) WAIT: Pin used to request CPU bus wait (It is active in 1 WAIT + N mode. Set by the Bus-width/wait control register.)
P71 to P77	7	I/O	Port 7: I/O port (High current output available)
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at falling and rising edges by program.
CLK	1	Output	Clock output: Outputs "$f_{SYS} \div 2$" Clock. Pulled-up during reset. Can be disabled for reducing noise.
$\overline{\text{EA}}$	1	Input	External access: "0" should be inputted with TMP93CS45. "1" should be inputted with TMP93CS44.

Table 2.2 Pin Names and Function (3/3)

Pin name	Number of pins	I/O	Functions
AM8 / $\overline{16}$	1	Input	Address Mode: Selects external Data Bus width. (the case of TMP93CS44) "1" should be inputted. The Data Bus Width for external access is set by Chip Select / WAIT Control register, Port 1 Control register. (the case of TMP93CS45) "0" should be inputted with fixed 16bit Bus Width or 16bit Bus interlarded with 8bit Bus. "1" should be inputted with fixed 8bit Bus Width.
ALE	1	Output	Address Latch Enable Can be disabled for reducing noise.
$\overline{RESET}$	1	Input	Reset: Initializes TMP93CS44/S45. (With pull-up resistor)
VREFH	1	Input	Pin for high level reference voltage input to A/D converter
VREFL	1	Input	Pin for low level reference voltage input to A/D converter
AVCC	1		Power supply pin for A/D converter
AVSS	1		GND pin for A/D converter (0 V)
X1	1	Input	High Frequency Oscillator connecting pin
X2	1	Output	High Frequency Oscillator connecting pin
P66	1	I/O	Port 66: I/O port (Open Drain Output)
/ XT1		Input	Low Frequency Oscillator connecting pin
P67	1	I/O	Port 67: I/O port (Open Drain Output)
/ XT2		Output	Low Frequency Oscillator connecting pin
TEST1 / TEST2	2	Output / Input	TEST1 Should be connected with TEST2 pin.
VCC	2		Power supply pin (All V_{CC} pins should be connected with GND (0V).)
VSS	2		GND pin (0 V) (All V_{SS} pins should be connected with GND (0V).)

Note : Built-in Pull-up resistors can be released from the pins other than the $\overline{RESET}$ pin by software.

3. OPERATION

This section describes the functions and basic operational blocks of TMP93CS44/ S45 devices.
See the「7. Points of Concern and Restrictions」for the using notice and restrictions for each block.

3.1 CPU

TMP93CS44/S45 devices have a built-in high-performance 16-bit CPU (900 / L CPU). (For CPU operation, see TLCS-900 / L CPU in the previous section).
This section describes CPU functions unique to the TMP93CS44/S45 that are not described in the previous section.

3.1.1 Reset

To reset the TMP93CS44/S45, the $\overline{\text{RESET}}$ input must be kept at 0 for at least 10 system clocks Resetting initializes the clock gear to 1/16. (16 μs at 20 MHz) within the operating voltage range and with a stable oscillation.
When reset is accepted, the CPU sets as follows:

- Program Counter (PC) according to Reset Vector that is stored FFFF00H to FFFF02H.
  PC (7 to 0) ← stored data in location FFFF00H
  PC (15 to 8) ← stored data in location FFFF01H
  PC (23 to 16) ← stored data in location FFFF02H
- 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 1. (Sets to maximum 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.
- Pulls up the CLK pin to 1"H" level.
- Sets the ALE pin to "L" level (the case of TMP93CS45), to High Impeadance (Hi-Z) (the case of TMP93CS44) .

Note 1 : 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.

Note 2 : The CLK pin is pulled up to "H" level during reset. When the voltage is put down externally, there is possible to cause malfunctions.

Figure 3.1 (1) and (2) show the reset timing chart of TMP93CS45 and TMP93CS44.

3.1.2 AM8/16 pin

(1) TMP93CS44

Set this pin to “H”. After reset, the CPU accesses the internal ROM with 16 bit bus width. The bus width when the CPU accesses an external area is set by Chip Select/Wait Control Register and the registers of Port 1, which are described in section 3.6.2 (The value of this pin is ignored and the value set by register is active.)

(2) TMP93CS45

① With fixed 16-bit data bus external 16-bit data bus or 8-bit data bus is selectable

Set this pin to “L”. Port 1, AD8 to15 and A8 to 15 pins are fixed to AD8 to 15 functions. The values set in Port 1 control register and Port 1 function register are invalid.
The external data bus width is set by the chip select/wait control register which is described in section 3.6.2.
It is necessary to set the program memory to be accessed to 16-bit data bus after reset.

② With fixed external 8-bit data bus

Set this pin to “H”. Port 1, AD8 to 15 and A8 to 15 pins are fixed to A8 to 15 functions. The values set in Port 1 control register and Port 1 function register are invalid.
The values of bit 4 <B0BUS>, <B1BUS> and <B2BUS> in the chip select/wait control register described in section 3.6.2 are invalid. The external 8-bit data bus is fixed.

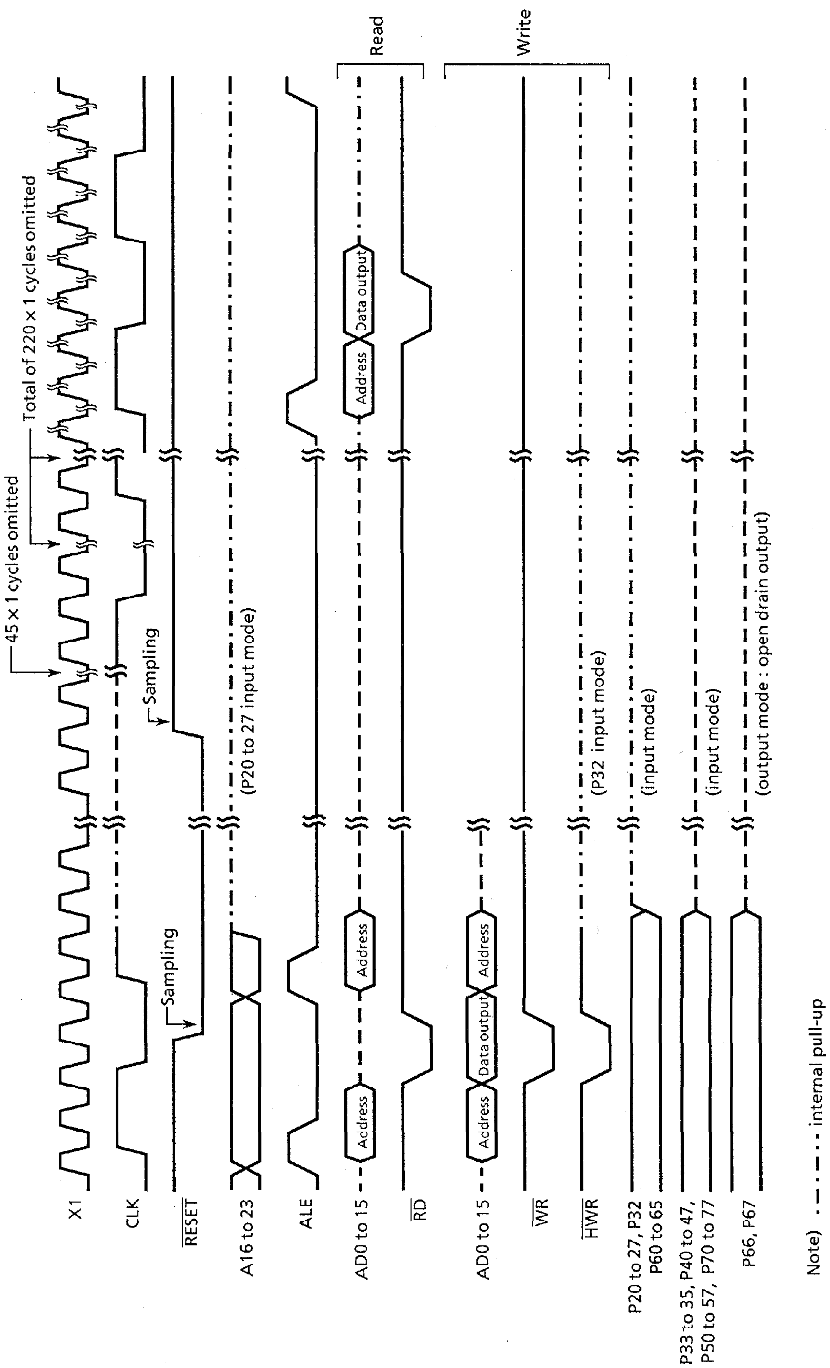


Figure 3.1 (1) TMP93CS45 Reset Timing Chart

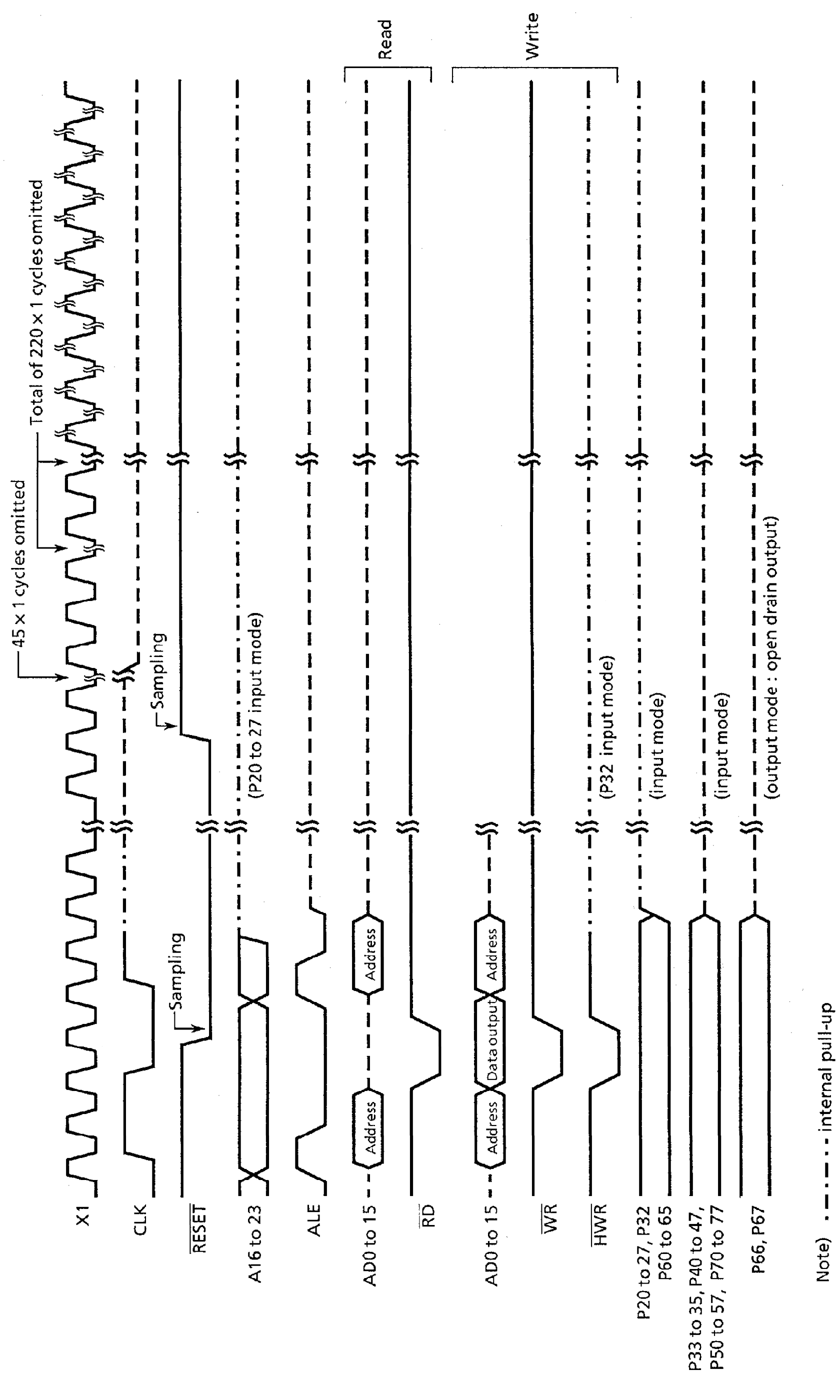


Figure 3.1 (2) TMP93CS44 Reset Timing Chart

3.2 Memory Map

Figure 3.2 is a memory map of the TMP93CS44/S45.

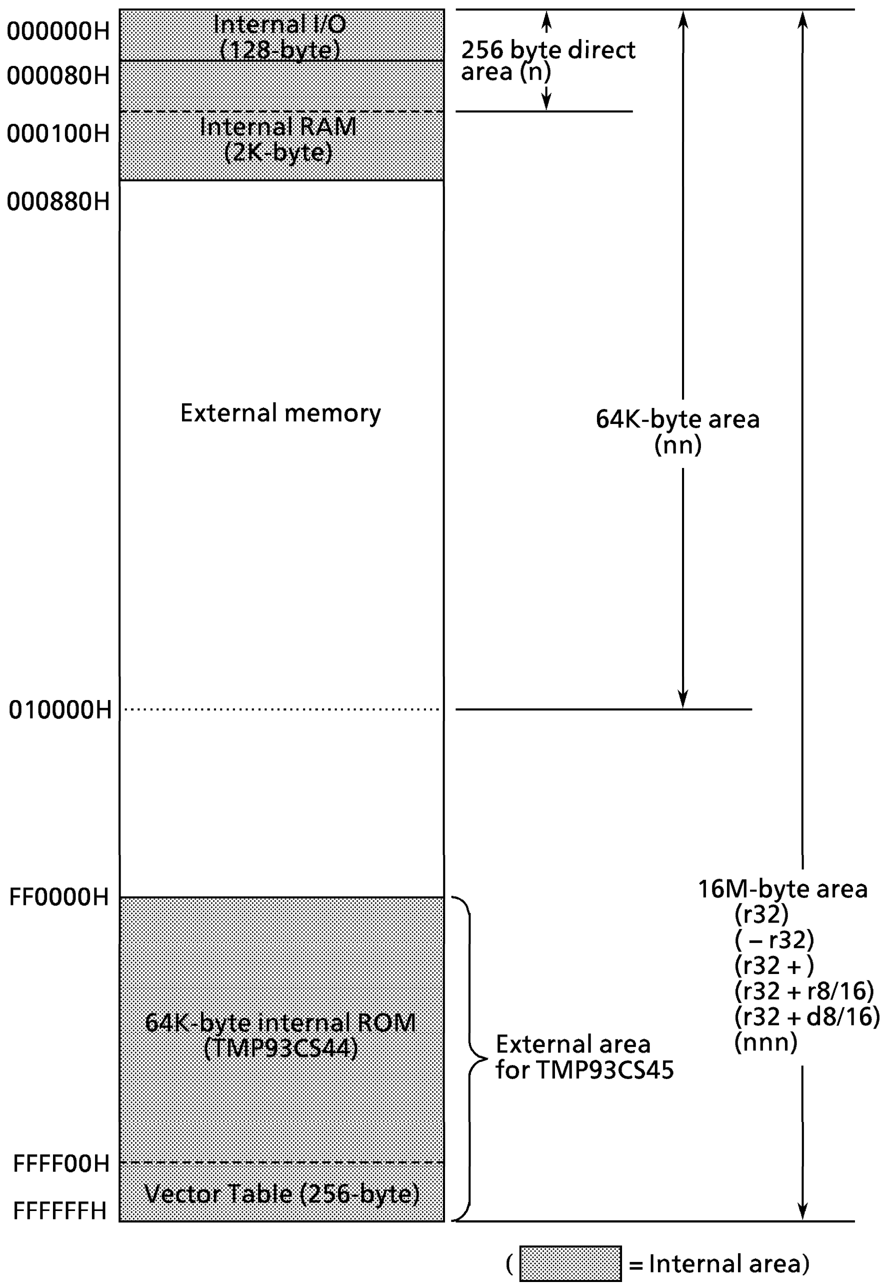
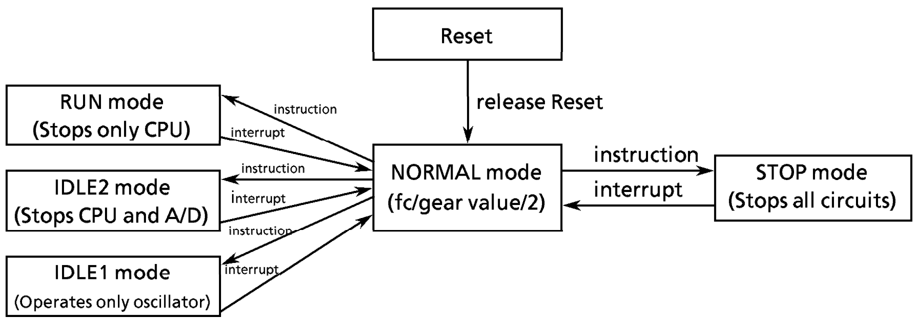


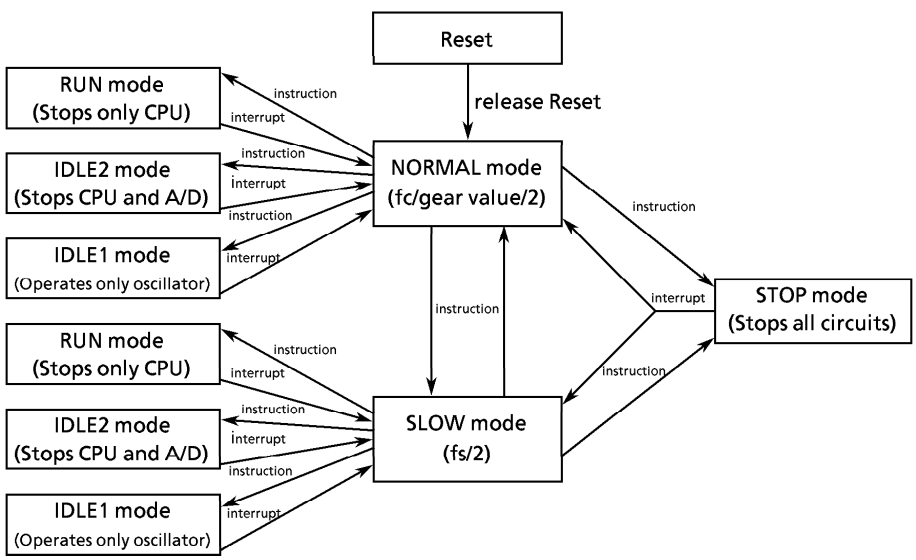
Figure 3.2 Memory map

3.3 Dual Clock, Standby Function

Dual Clock, Stand by Control Circuits consist of (1) System clock Controller, (2) Prescaler Clock Controller and (3) Standby Controller.
The Oscillator operating 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. Table 3.3 (1) shows the internal operation and system clock.



(a) Single Clock mode transition figure



(b) Dual Clock mode transition figure

Figure 3.3 (1) 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 devided clock of f_{FPH} is called system clock f_{SYS}, and the 1 cycle of f_{SYS} is called 1 state.

Table 3.3 (1) Internal operation and system clock

<table>
<tr><th colspan="2" rowspan="2">Operating Mode</th><th colspan="2">Oscillator</th><th rowspan="2">CPU</th><th rowspan="2">internal I/O</th><th rowspan="2">System clock f_{SYS}</th></tr>
<tr><th>High Frequency (fc)</th><th>Low Frequency (fs)</th></tr>
<tr><td rowspan="6">Single Clock</td><td>RESET</td><td rowspan="5">oscillation</td><td rowspan="6">stop</td><td>reset</td><td>reset</td><td>$fc/32$</td></tr>
<tr><td>NORMAL</td><td>operate</td><td rowspan="2">operate</td><td rowspan="4">programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$)</td></tr>
<tr><td>RUN</td><td rowspan="4">stop</td></tr>
<tr><td>IDLE2</td><td>stop only A/D</td></tr>
<tr><td>IDLE1</td><td rowspan="2">stop</td></tr>
<tr><td>STOP</td><td>stop</td><td>stop</td></tr>
<tr><td rowspan="7">Dual Clock</td><td>RESET</td><td rowspan="2">oscillation</td><td>stop</td><td>reset</td><td>reset</td><td>$fc/32$</td></tr>
<tr><td>NORMAL</td><td>programmable</td><td rowspan="2">operate</td><td rowspan="3">operate</td><td>programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$)</td></tr>
<tr><td>SLOW</td><td>programmable</td><td>oscillation</td><td>$fs/2$</td></tr>
<tr><td>RUN</td><td colspan="2" rowspan="3">Oscillator being used as system clock : oscillation
Other oscillator : programmable</td><td rowspan="4">stop</td><td rowspan="3">programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$, $fs/2$)</td></tr>
<tr><td>IDLE2</td><td>stop only A/D</td></tr>
<tr><td>IDLE1</td><td rowspan="2">stop</td></tr>
<tr><td>STOP</td><td colspan="2">stop</td><td>stop</td></tr>
</table>

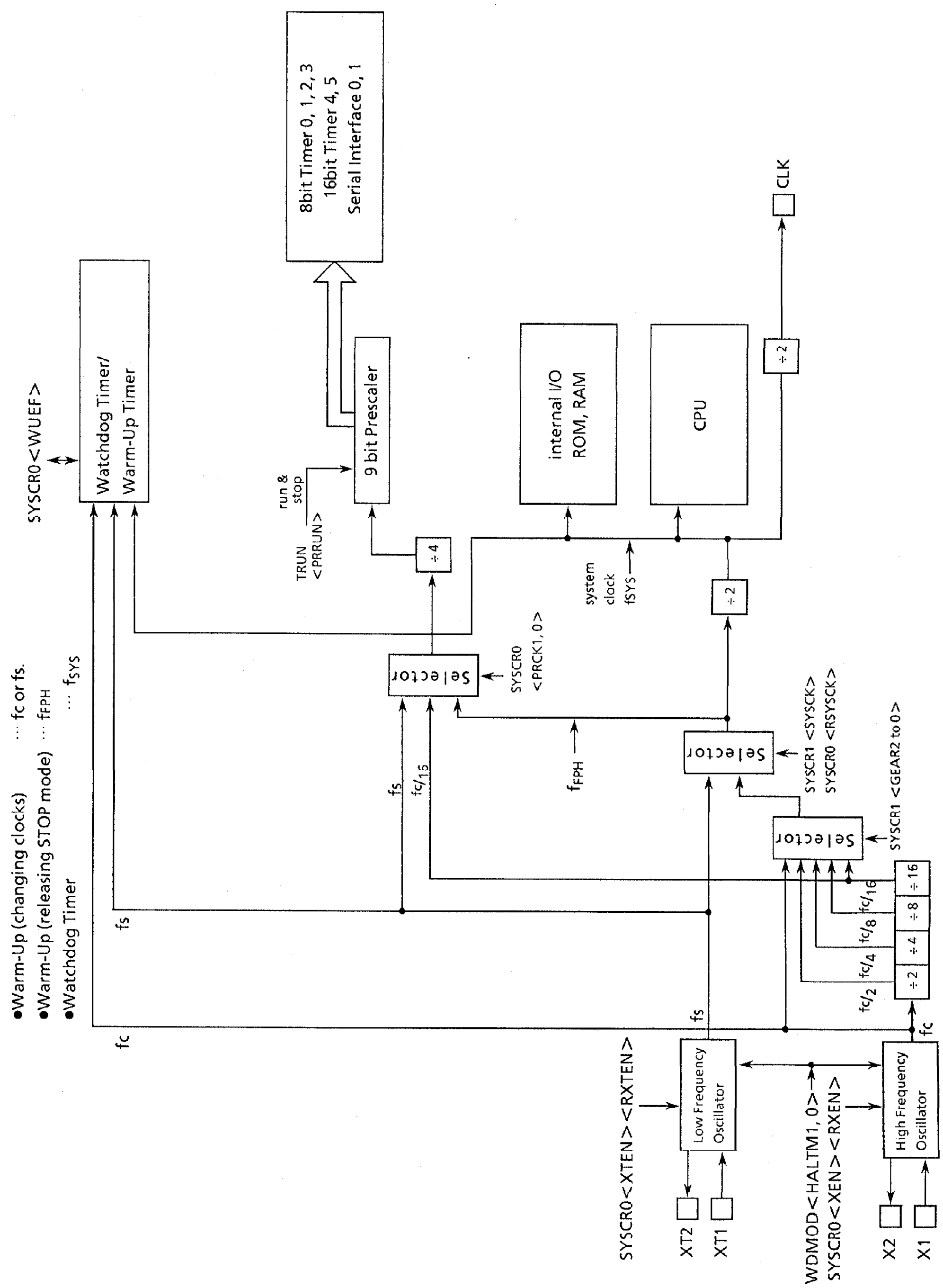


Figure 3.3 (2) Block Diagram of Dual Clock, Standby circuits

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	Warm-Up Timer 0 write : don't care 1 write : start timer 0 read : end warm-up 1 read : not end warm-up	select prescaler clock 00 : f_{FPH} 01 : fs 10 : fc/ (16 × 4) 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)		

CKOCR (006DH)

	7	6	5	4	3	2	1	0
bit Symbol							ALEEN	CLKEN
Read / Write							R/W	
After reset							0/1 (Note2)	0/1 (Note2)
Function							ALE Pin output control 0 : HZ output 1 : ALE output	CLK pin output control 0 : HZ output 1 : CLK output

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read / Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0 : disable 1 : enable	WDT Detection Time 00 : $2^{15}/f_{SYS}$ 01 : $2^{17}/f_{SYS}$ 10 : $2^{19}/f_{SYS}$ 11 : $2^{21}/f_{SYS}$		Warm-Up Timer 0 : 2^{14}/ frequency inputted 1 : 2^{16}/ frequency inputted	HALT mode 00 : RUN mode 01 : STOP mode 10 : IDLE1 mode 11 : IDLE2 mode		0 : Don't care 1 : Connects WDT output to $\overline{RESET}$ pin internally.	Pin state control in STOP mode 0 : I/O off 1 : Remains the state before HALT

Note 1 : SYSCR1<bit 7-4>and CK0CR<bit 7-2>are read as "1".

Note 2 : In the TMP93CS44, resetting sets <ALEEN>, <CLKEN>bit to "0". (High impedance ALE and CLK) In the TMP93CS45, resetting sets <ALEEN>, <CLKEN>bit to "1".(output ALE and CLK) The CLK pin is internally pulled up during reset, regardless of the product types.

Note 3 : Writing "0" to SYSCR1<SYSCK>enables the high-frequency oscillator regardless of the value of SYSCR0<XEN>. Additionally, writing "1" to <SYSCK> register enebles the low-frequency oscillator regardless of th value of SYSCR0<XTEN>.

Figure 3.3 (3) I/O registers about Dual Clock, Standby

3.3.1 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 to 0> changes high frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16), and these functions can reduce the power consumption of the equipment in which the device is installed.

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

For example, f_{SYS} is set to 0.625 MHz by resetting the case of 20 MHz 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 66, 67 function. Therefore the case of single clock mode, the XT1, XT2 pins can be used as I/O port pins.

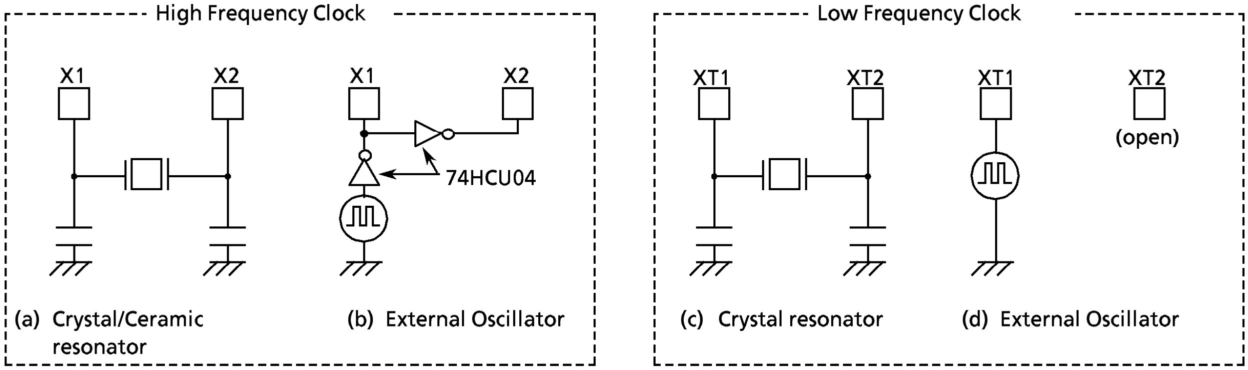


Figure 3.3 (4) Examples of Resonator Connection

Note 1 : Note on using the low frequency oscillation circuit.
In connecting the low frequency resonator to ports 66 and 67, it is necessary to make the following settings to reduce the power consumption.
(connecting with resonators) P6CR<P66C, P67C> = "11", P6<P66, P67> = "00"
(connecting with oscillators) P6CR<P66C, P67C> = "11", P6<P66, P67> = "10"

Note 2 : Accurate adjustment of the oscillation frequency
The CLK pin outputs at 1/2 the system clock frequency (FSYS/2) is used to monitor the oscillation clock. With a system requiring adjustment of the oscillation frequency, an adjusting program must be wvitten.

(1) Switching from NORMAL to SLOW mode

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

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

Note 2 : The case of using oscillator (not resonator) with stabilized oscillation, a warm-up timer is not need.

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

Table 3.3 (2) Warm-up Time

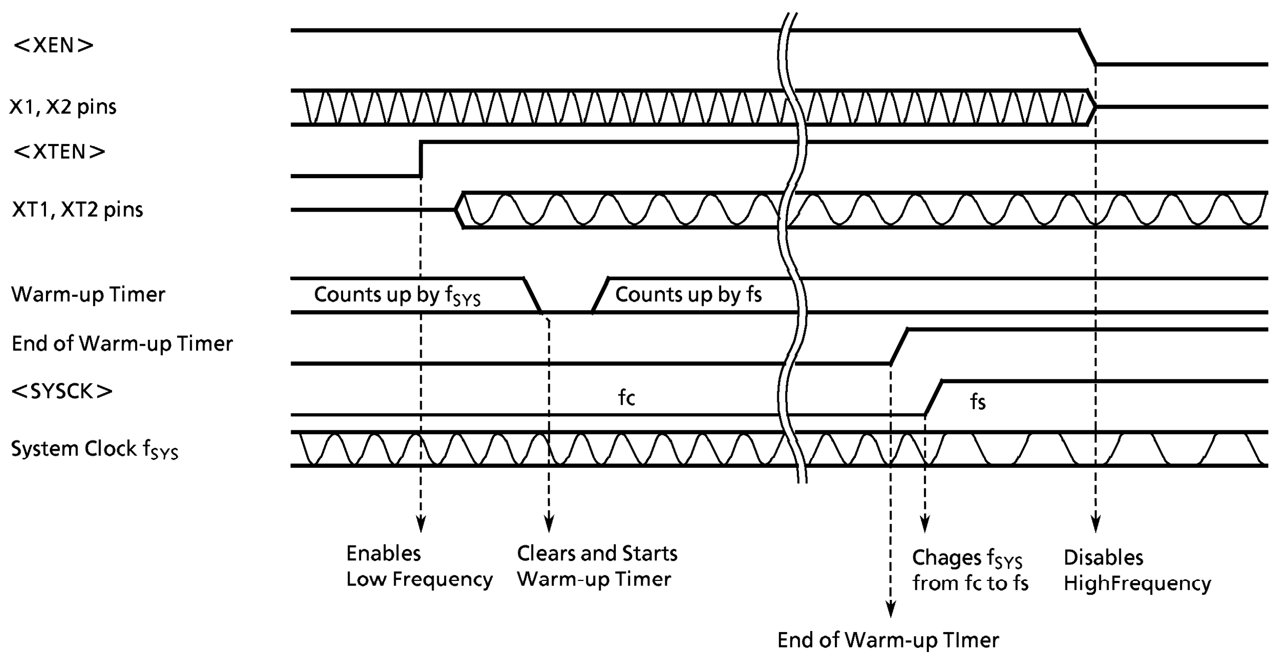
Warm-up Time WDMOD<WARM>	Change to NORMAL	Change toSLOW
0 (2^{14}/frequency)	0.8192 (ms)	500 (ms)
1 (2^{16}/frequency)	3.2768 (ms)	2000 (ms)

at fc = 20 MHz,
fs = 32.768 kHz

Clock Setting Example 1 :

Changing from the high frequency (fc) to the low frequency (fs).

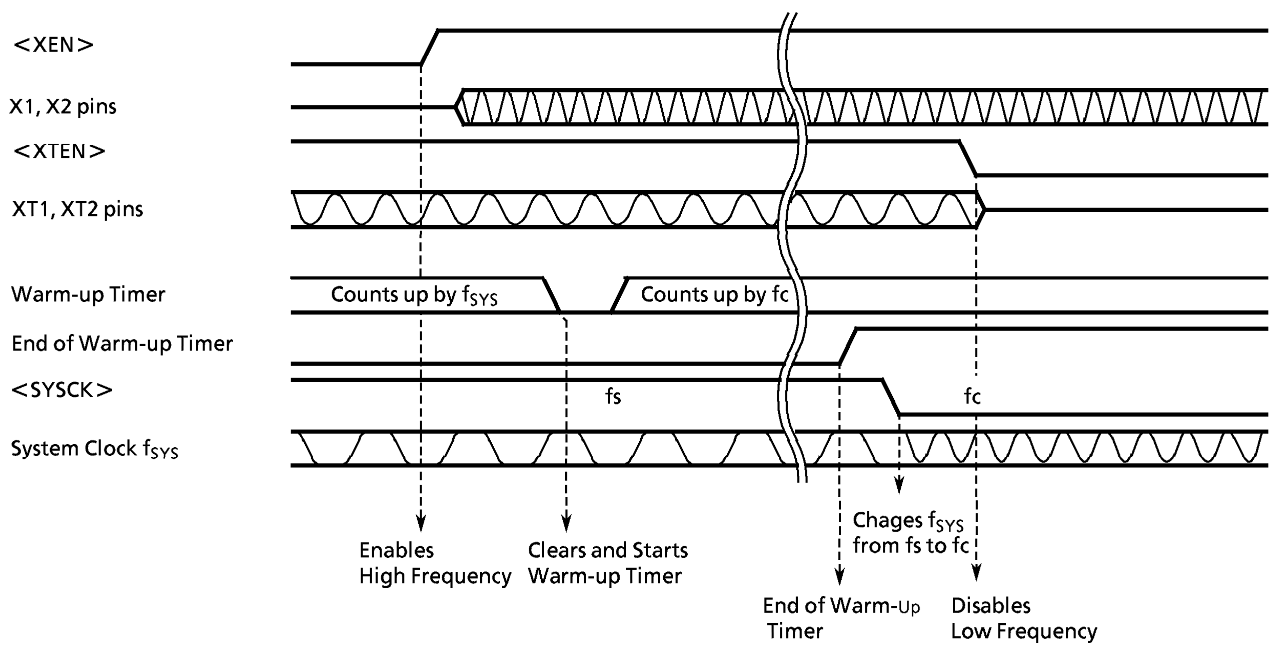
```
SYSCR0   EQU   006EH
SYSCR1   EQU   006FH
WDCR     EQU   005DH
WDMOD    EQU   005CH
         RES   7, (WDMOD)      ; } Disables Watchdog Timer.
         LD    (WDCR), B1H     ; }
         SET   4, (WDMOD)      ; Sets Warm-up Time to 2^16/fs.
         SET   6, (SYSCR0)     ; Enables Low Frequency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



Clock Setting Example 2 :

changing from the low frequency (fs) to the high frequency (fc).

```
SYSCR0    EQU    006EH
SYSCR1    EQU    006FH
WDCR      EQU    005DH
WDMOD     EQU    005CH
          RES    7, (WDMOD)       ; } Disables Watchdog Timer.
          LD     (WDCR), B1H      ; }
          RES    4, (WDMOD)       ; Sets Warm-up Time to 2^14/fc.
          SET    7, (SYSCR0)      ; Enables High Frequency (fc).
          SET    2, (SYSCR0)      ; Clears and Starts Warm-up Timer.
WUP :     BIT    2, (SYSCR0)      ; } Detects End of Warm-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
```



(2) Clock Gear Controller

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

Clock Setting Example 3 :

Changing gear value of the high-frequency clock

```
SYSCR1   EQU   006FH
         LD    (SYSCR1), XXXX0000B    ; Changes fSYS to fc/2
         LD    (SYSCR1), XXXX0100B    ; Changes fSYS to fc/32
```

X : Don't care

(High-frequency clock gear changing)

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

(Example)

```
SYSCR1   EQU   006FH
         LD    (SYSCR1), XXXX0001B    ; Changes fSYS to fc/4.
         LD    (DUMMY), 00H           ; Dummy instruction
               Instruction to be executed by the clock gear after changing
```

X : Don't care

3.3.2 Prescaler Clock Controller

The 9-bit prescaler provides a clock to 8-bit Timer 0, 1, 2, 3, 16-bit Timer 4, 5, and Serial Interface 0, 1.
The clock input to the 9-bit prescaler is selected either f_{FPH}, fc/16, or fs by SYSCR0 < PRCK1, 0> register.
<PRCK1, 0> register is initialized to "00" by resetting.
When the IDLE 1 mode (operates only oscillator) is used, set TRUN<PRRUN> to '0' to stop 9 bit prescaler before 'HALT' instruction is executed.

3.3.3 Internal Clock Pin Output Function

CLK pin outputs f_{SYS} divided by 2 internal clock.
Outputs are specified by the clock output control register CKOCR<CLKEN>. Writing "1" sets clock output, and writing "0" sets high impedance. After reset, CKOCR<CLKEN> is depended on each product types. It is necessary to set for each usage. Table 3.3 (3) shows the value and operation after reset.
During reset, CLK pin is internally pulled up regardless of the value of <CLKEN> register. See TMP93CS44/S45 Reset Timing Chart in figure 3.1 (1), (2).

Table 3.3 (3) <CLKEN> and CLK pin operation after reset

Type No.	CKOCR<CLKEN>	CLK pin operation
TMP93CS44	0	High impedance
TMP93CS45	1	f_{SYS}/2 clock output

Note : To set C<CLKEN> = "0" and set CLK pin to high impedance, pull up externally to prevent through current which follows to the input buffer of CLK pin.

3.4 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask flip-flop <IFF2 to 0> and the built-in interrupt controller.
Altogether the TMP93CS44 / S45 have the following 33 interrupt sources:

Internal interrupts ············ 26

- Software interrupts : 8
- Illegal instruction execution : 1
- Interrupts from built-in I/Os : 17

External interrupts ············ 7

- External pins ($\overline{\text{NMI}}$, INT0, INT1, INT4, to INT7)

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority 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 interupt 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 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 (Executing EI n changes the contents of <IFF2 to 0> to n). 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 (<IFF2 to 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 acceptance of maskable interrupts. 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 Micro DMA processing mode . Micro DMA 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.

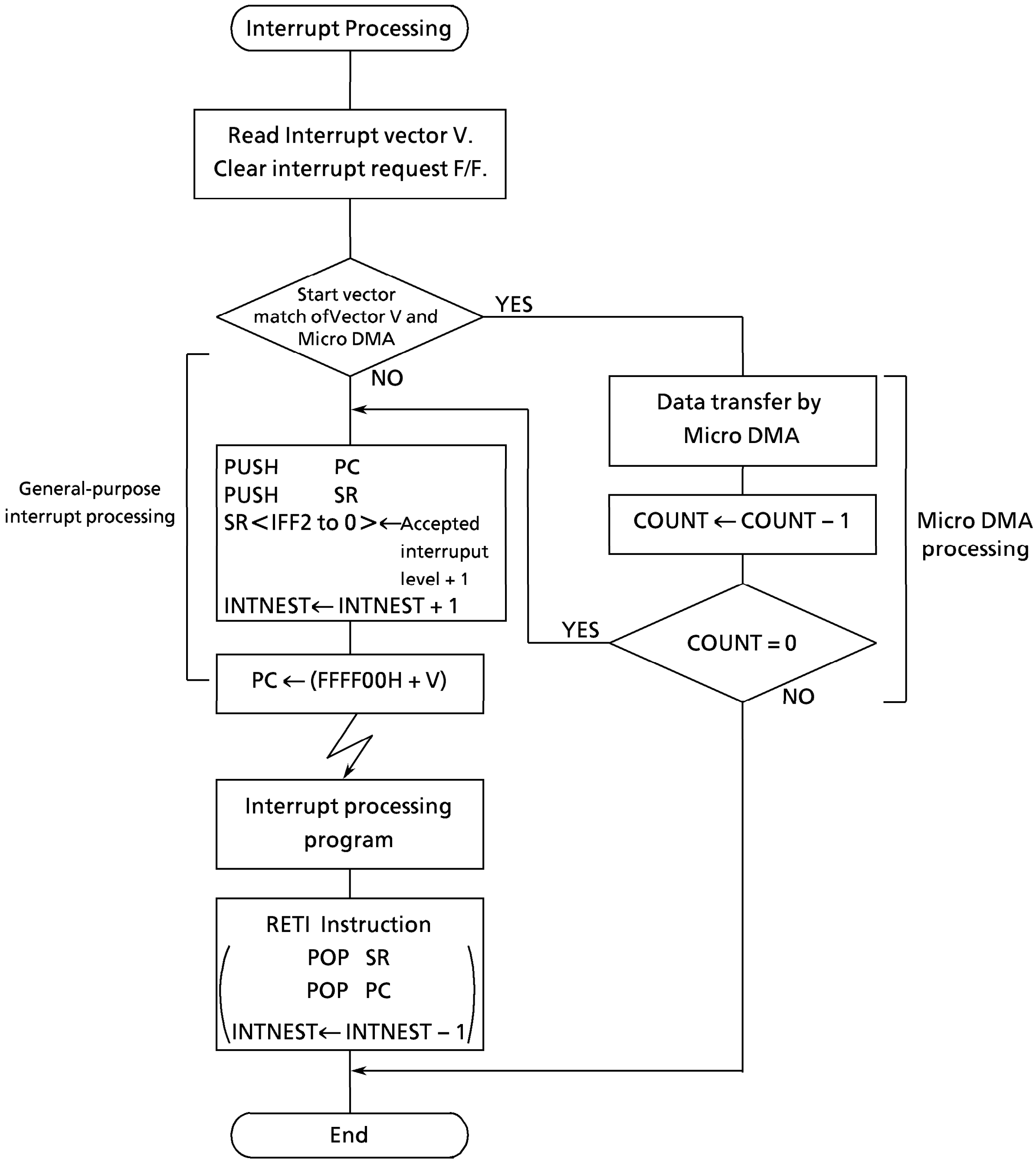


Figure 3.4 (1) Interrupt Processing Flowchart

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).

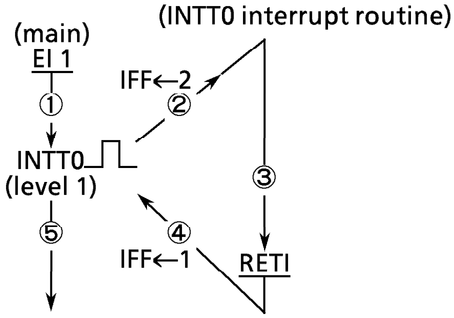
(5) The CPU jumps to address stored at FFFF00H + interrupt vector, then starts the interrupt processing routine.

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
8 bit	8 bit	35
	16 bit	31
16 bit	8 bit	29
	16 bit	25

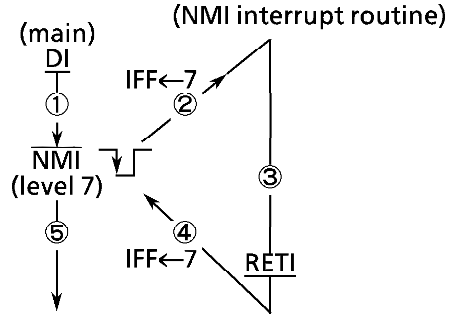
To return to the main routine after completion of the interrupt processing, the RETI instruction is usually used. Executing this instruction restores the contents of the program counter and the status registers and decrements INTNEST (Interrupt Nesting Counter).
Though acceptance of non-maskable interrupts 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 1'st instruction in the interrupt processing routine, causing interrupt processing to nest. (This is the same case of over lapped each Non-Maskable interrupt (level "7").) The CPU does not accept an interrupt request 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 following (1) to (5) show a flowchart of interrupt processing.

(1) Maskable interrupt



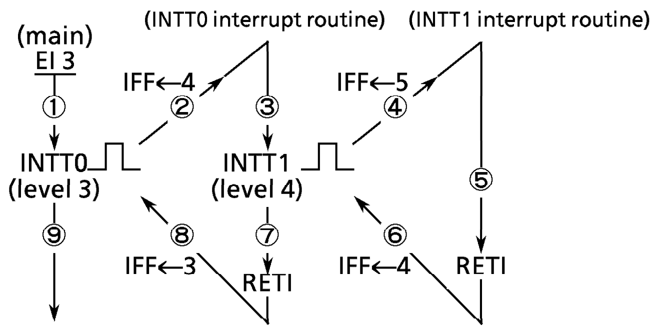
During execution of the main program, the CPU accepts an interrupt request. The CPU increments the IFF so that the interrupts of level 1 are not accepted during processing the interrupt routine.

(2) Non-maskable interrupt



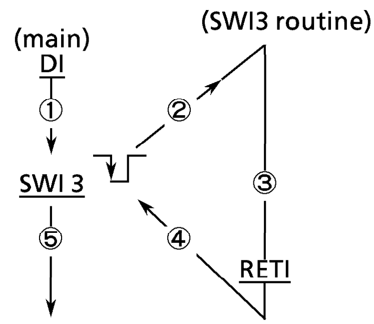
DI instruction is executed in the main program, so that the interrupts of only level 7 are accepted. The CPU does not increment the IFF even if the CPU accepts an interrupt request of level 7.

(3) Interrupt nesting



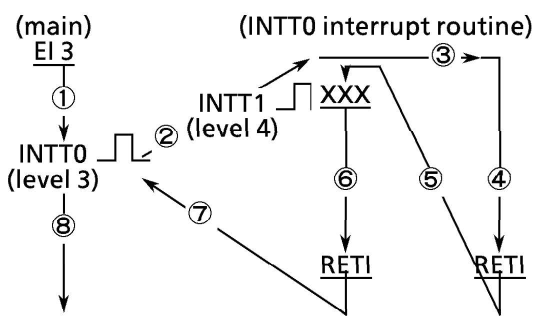
During processing the interrupts of level 3, the IFF is set to 4. When an interrupt with a level higher than level 4 is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

(4) Software interrupt



The CPU accepts the software interrupt request during DI status(IFF = 7) because of the level 7. The IFF is not changed by the software interrupts.

(5) Interrupt sampling timing



If an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level. The program counter which returns at ⑤ is the start address of INTT0 interrupt routine.

(example) ____ (underline) : Instruction
①, ②, ··· : Execution flow

The addresses FFFF00H to FFFFFFH (256 bytes) of the TMP93CS44/S45 are assigned for interrupt vector area.

Table3.4 (1) TMP93CS44 / S45 Interrupt Table

Default priority	Type	Interrupt source	Vecter value "V"	Address refer to vector	HDMA start vector
1	Non-maskable	Reset or SWI0 instruction	0000H	FFF00H	–
2		SWI 1 instruction	0004H	FFF04H	–
3		Illegal instruction, or SWI2	0008H	FFF08H	–
4		SWI 3 instruction	000CH	FFF0CH	–
5		SWI 4 instruction	0010H	FFF10H	–
6		SWI 5 instruction	0014H	FFF14H	–
7		SWI 6 instruction	0018H	FFF18H	–
8		SWI 7 instruction	001CH	FFF1CH	–
9		NMI : $\overline{\text{NMI}}$ Pin input	0020H	FFF20H	08H
10		INTWD : Watch-dog timer	0024H	FFF24H	09H
11	Maskable	INT0 : INT0 pin input	0028H	FFFF28H	0AH
12		INT1 : INT1 pin input	002CH	FFFF2CH	0BH
13		INT4 : INT4 pin input	0030H	FFFF30H	0CH
14		INT5 : INT5 pin input	0034H	FFFF34H	0DH
15		INT6 : INT6 pin input	0038H	FFFF38H	0EH
16		INT7 : INT7 pin input	003CH	FFFF3CH	0FH
17		INTT0 : 8-bit timer0	0040H	FFFF40H	10H
18		INTT1 : 8-bit timer1	0044H	FFFF44H	11H
19		INTT2 : 8-bit timer2	0048H	FFFF48H	12H
20		INTT3 : 8-bit timer3	004CH	FFFF4CH	13H
21		INTTR4 : 16-bit timer4 (TREG4)	0050H	FFFF50H	14H
22		INTTR5 : 16-bit timer4 (TREG5)	0054H	FFFF54H	15H
23		INTTR6 : 16-bit timer5 (TREG6)	0058H	FFFF58H	16H
24		INTTR7 : 16-bit timer5 (TREG7)	005CH	FFFF5CH	17H
25		INTT04 : 16-bit timer4 (Over flow)	0060H	FFFF60H	18H
26		INTT05 : 16-bit timer5 (Over flow)	0064H	FFFF64H	19H
27		INTRX0 : Serial receive (Channel.0)	0068H	FFFF68H	1AH
28		INTTX0 : Serial send (Channel.0)	006CH	FFFF6CH	1BH
29		INTRX1 : Serial receive (Channel.1)	0070H	FFFF70H	1CH
30		INTTX1 : Serial send (Channel.1)	0074H	FFFF74H	1DH
31		INTAD : A/D conversion completion	0078H	FFFF78H	1EH
32		INTS2 : Serial bus send and receive	007CH	FFFF7CH	1FH
–		(Reserved)	0080H	FFFF80H	20H
to		to	to	to	to
–		(Reserved)	00FCH	FFFFFCH	3FH

Setting to Reset / Interrupt Vector

① Reset Vector

FFFF00H	PC (7 to 0)
FFFF01H	PC (15 to 8)
FFFF02H	PC (23 to 16)
FFFF03H	XX

The vector base addresses are depended on the products.

Type No.	Vector base address	PC setting sequence after reset	Notes
TMP93CS44 TMP93CS45 TMP93PS44	FFFF00H	PC (7 to 0) ← address FFFF00H PC (15 to 8) ← address FFFF01H PC (23 to 16) ← address FFFF02H	P27 to 20/A23 to 16 pins input ports with pull-up due to reset. The logic data is "FFH". When Port 2 is used as A23 to 16 pins to access the program ROM, set PC (23 to 16) to "FFH" and the reset vector to "FF0000H to FFFFFFH". (for mainly products without ROM)

② Interrupt Vector (except Reset Vector)

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

XX : don't care

(Setting Example)

Sets the Reset Vector : FF0000H, $\overline{\text{NMI}}$ Vector : FF9ABCH, INTAD Vector : 123456H.

```
ORG    FFFF00H
DL     FF0000H      ; Reset = FF0000H

ORG    FFFF20H
DL     FF9ABCH      ; NMI = FF9ABCH

ORG    FFFF78H
DL     123456H      ; INTAD = 123456H

ORG    FF0000H
LD     A, B
  :
ORG    FF9ABCH
LD     B, C
  :
ORG    123456H
LD     C, A
  :
```

Note :

ORG, DL are Assembler Directives.

ORG : control location counter
DL : defines long word (32-bits) data

3.4.2 Micro DMA

In addition to the conventional interrupt processing, the TLCS-900 also has a Micro DMA function. When an interrupt is accepted, in addition to an interrupt vector, the CPU receives data indicating whether processing is Micro DMA mode or general-purpose interrupt. If Micro DMA mode is requested, the CPU performs Micro DMA processing.
The TLCS-900 can process at very high speed 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) Micro DMA operation

Micro DMA operation starts when the accepted interrupt vector value matches the Micro DMA start vector value. The Micro DMA has four channels so that it can be set for up to four types of interrupt source.
When a Micro DMA 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, Micro DMA processing is completed; if the value in the counter after decrementing is 0, general-purpose interrupt processing is performed.
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 Micro DMA.
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 Micro DMA processing.
When the transfer counter is decremented to “0” after data is transferred with micro DMA, general-purpose interrupt processing is performed. After processing the general-purpose interrupt, starting the interrupts of the same channel restarts the transfer counter from 65536. If necessary, reset the transfer counter.
Interrupt sources processed by Micro DMA processing are those with the Micro DMA start vectors listed in Table 3.4 (1).
The following timing chart is a Micro DMA cycle of the Transfer Address INC rement mode (Condition : MAX mode, 16 bit Bus width for 16M Byte, 0 wait).

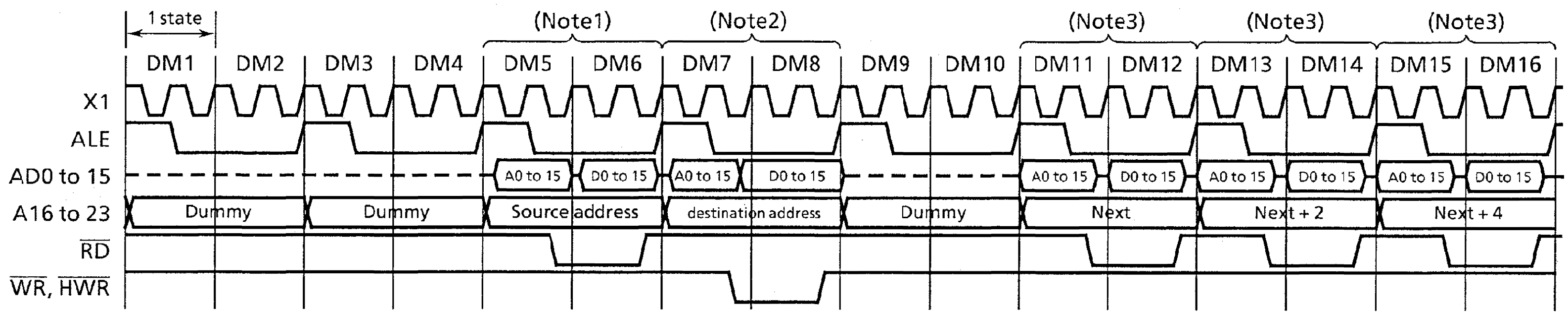


Figure 3.4 (2) Micro DMA cycle (COUNT ≠ 0)

(Note 1) These 2 states are added in the case that the bus width of the source address area is 8 bits.

(Note 2) These 2 states are added in the case that the bus width of the destination address area is 8 bits.

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

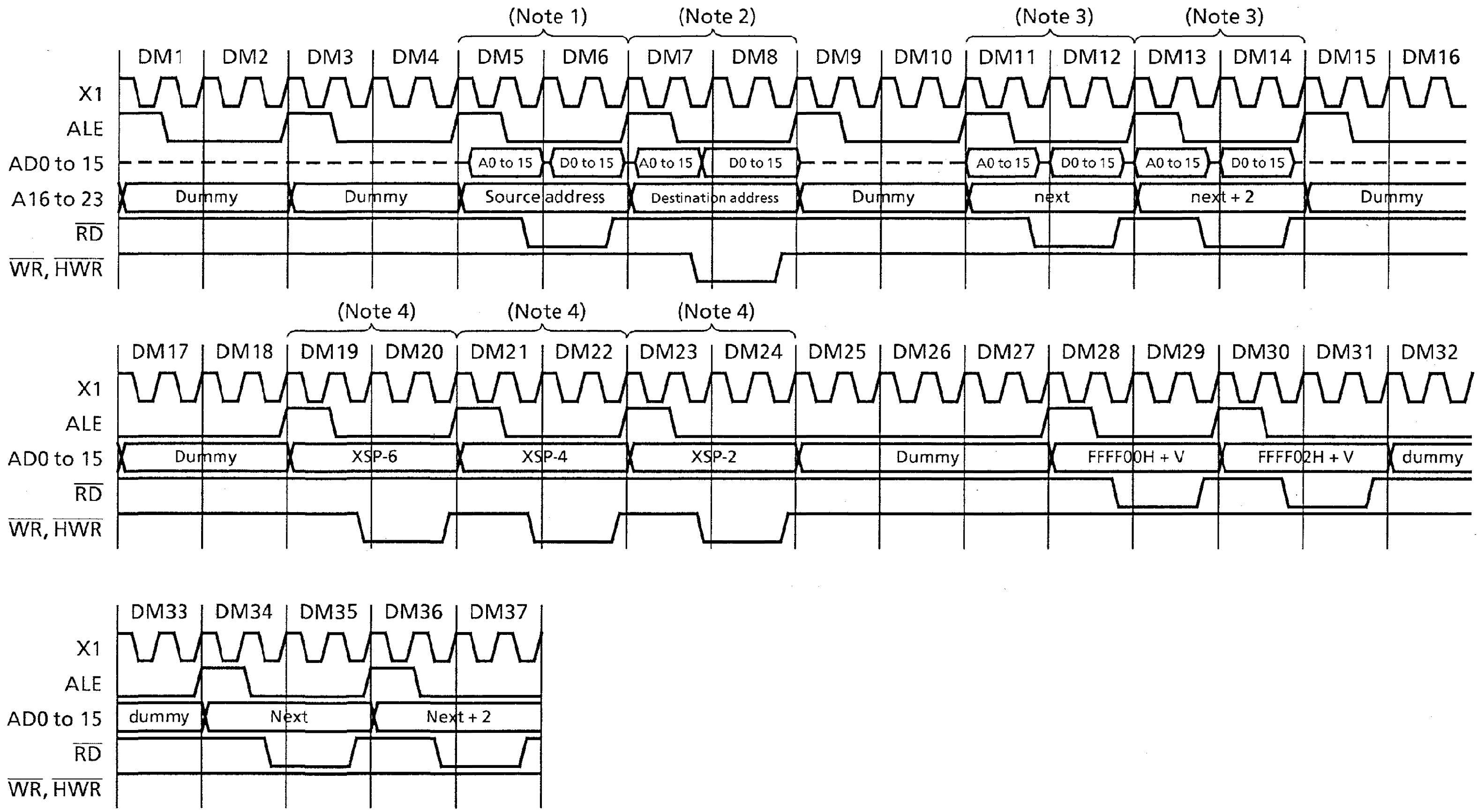


Figure 3.4 (3) Micro DMA cycle (COUNT = 0)

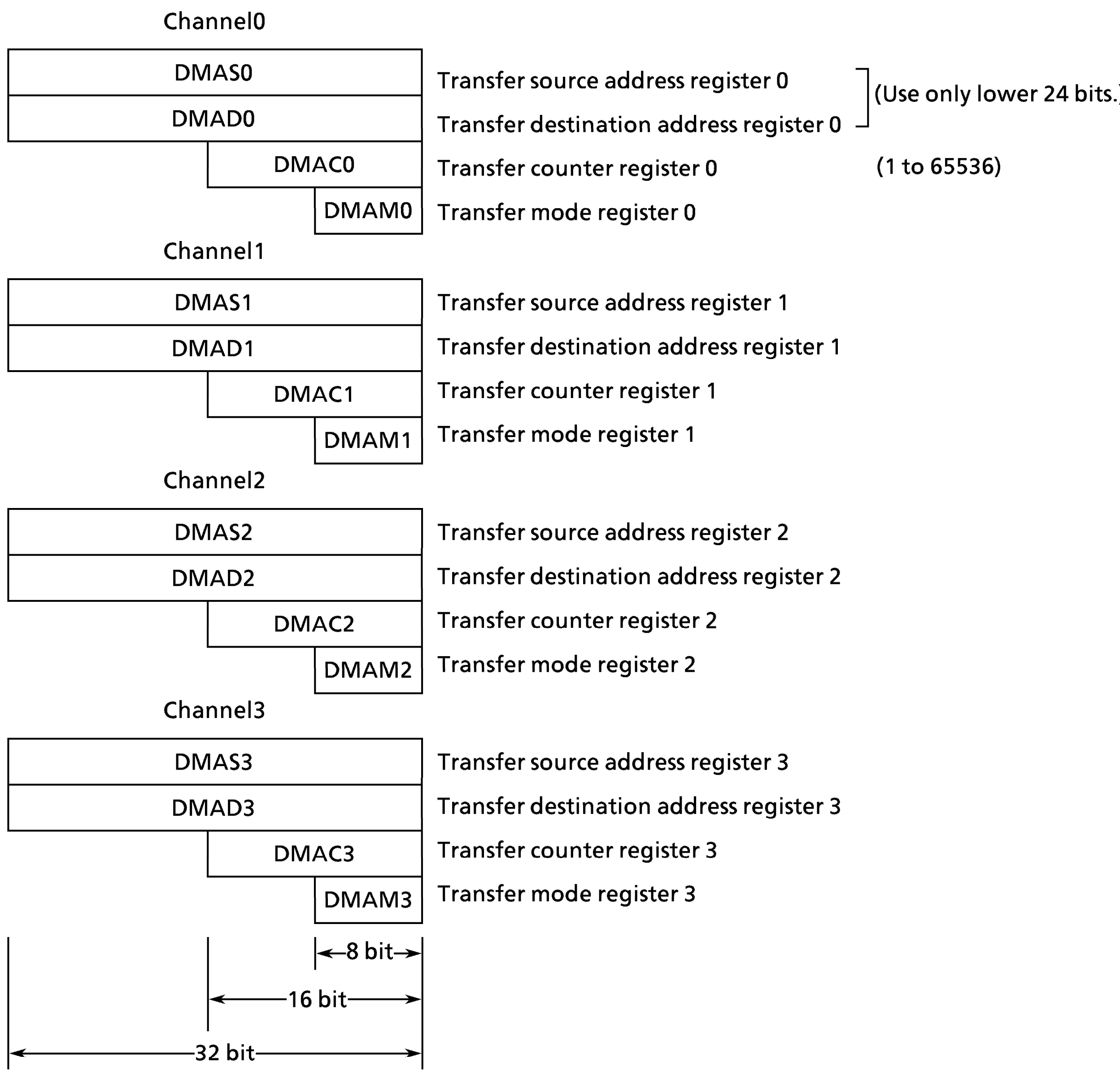
(Note 1) These 2 states are added in the case that the bus width of the source address area is 8 bits.

(Note 2) These 2 states are added in the case that the bus width of the destination address area is 8 bits.

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

(Note 4) These 2 states are added in the case of the bus width of stack address area is 8 bits.

(2) Register configuration (CPU control register)



These Control Register can not be set only “LDC cr, r” instruction.

Example :

```
LD      XWA, 100H
LDC     DMAS0, XWA
LD      XWA, 50H
LDC     DMAD0, XWA
LD      WA, 40H
LDC     DMAC0, WA
LD      A, 05H
LDC     DMAM0, A
```

(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 memory to I/O (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 memory to I/O (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	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)

Note1 : n : corresponds to micro DMA channels 0 to 3.
DMADn + / DMASn + : Post-increment (Increments register value after transfer.)
DMADn – / DMASn – : Post-decrement (Decrement register value after transfer.)

Note2 : Execution time : When setting source address/destination address area to 16-bit bus, 0WAIT.
Clock condition : fc = 20 MHz, Clock gear : 1 (fc)

Note3 : Do not use the codes other than the above mentioned codes for transfer mode register.

3.4.3 Interrupt Controller

Figure 3.4 (2) 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 24 channels) in the interrupt controller has an interrupt request flip-flop, interrupt priority setting register, and a register for storing the Micro DMA start vector. The interrupt request fip-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 - - - B

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 (eg, 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 Micro DMA start vector. These are I/O registers; unlike other Micro DMA registers (DMAS, DMAD, DMAM, and DMAC). Writing the start vector of the interrupt source for the Micro DMA processing (see Table 3.4.(1)), enables the corresponding interrupt to be processed by Micro DMA processing. The values must be set in the Micro DMA parameter registers (eg, DMAS and DMAD) prior to the Micro DMA processing.

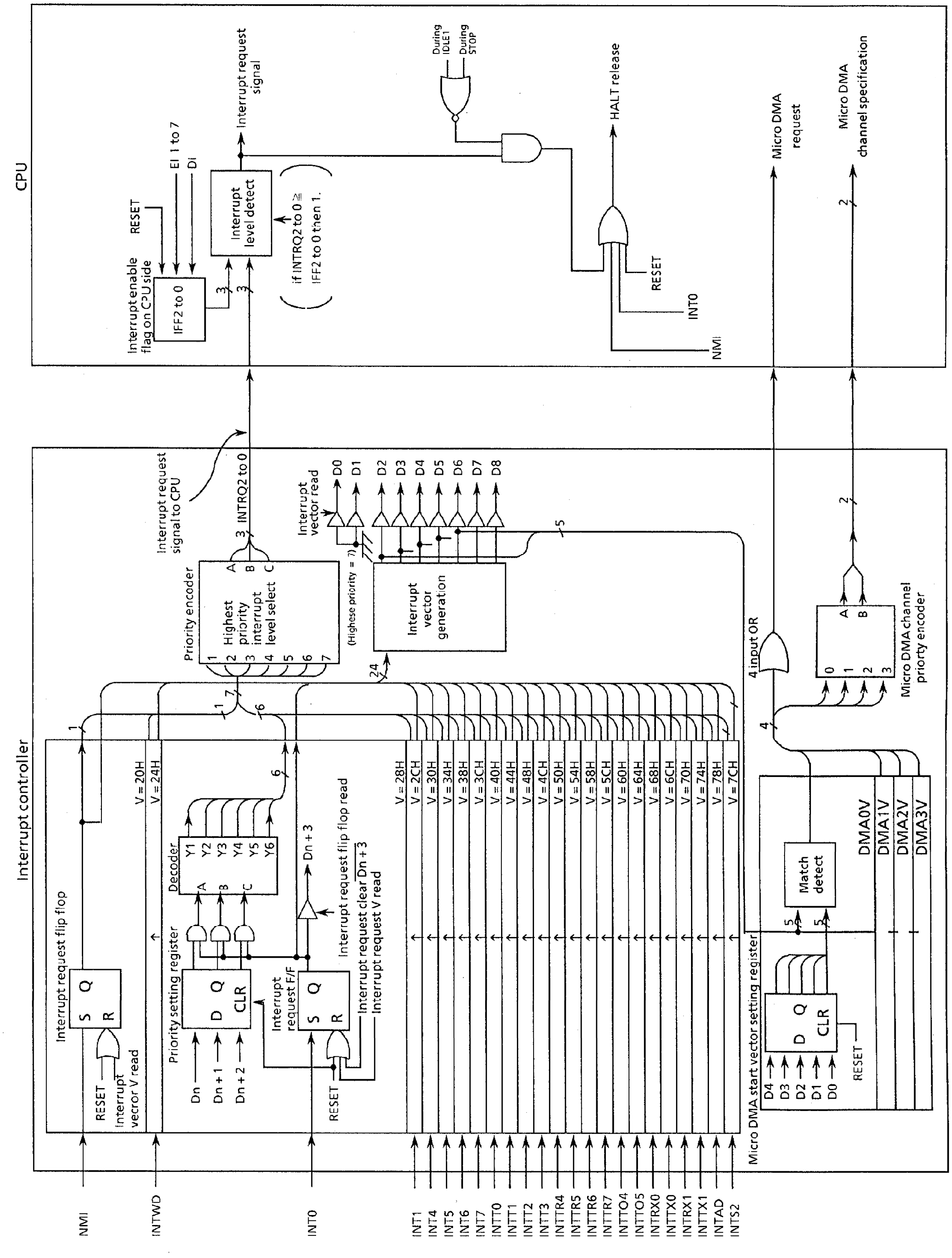


Figure3.4 (2) Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

(Prohibit read-modify-write)

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	
INTET32	0074H	INTT3 (Timer3)				INTT2 (Timer2)				
		IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0	
		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	
INTEO54	0077H	INTTO5				INTTO4				
		ITO5C	ITO5M2	ITO5M1	ITO5M0	ITO4C	ITO4M2	ITO4M1	ITO4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES0	0078H	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	0079H	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE1S2	007AH	INT1				INTS2				
		I1C	I1M2	I1M1	I1M0	IS2C	IS2M2	IS2M1	IS2M0	
		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 1: Read-modify-write is prohibited.
Note 2: Note about clearing interrupt request flag
The interrupt request flag of INTRX0, INTRX1 are not cleared by writing "00" to IXXC because of they are level interrupts. They can be cleared only by resetting or reading SCBUFn.

* Note about clearing interrupt request flag

The interrupt request flag of INTRX0, INTRX1 are not cleared by writing "00" to IXXC because of they are level interrupt s.
They can be cleared only by resetting or reading SCBUFn.

(2) External interrupt control

Interrupt Input Mode Control Register

IIMC (007BH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			–			I0IE	I0LE	NMIREE
Read / Write						W		
After reset						0	0	0
Function			(Note) Always write "0".			1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Can be accepted in $\overline{\text{NMI}}$ rising edge.

INT0 input enable (Note1)

0	INT0 disable (P35 function only)
1	Input enable

$\overline{\text{NMI}}$ rising edge enable

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

INT0 level enable (Note2)

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 disable, clean the request flag
LD (IIMC)   , xxxxx10xB   ; Change from level to edge
LD (INTE0AD), xxxx0nnnB   ; Set  interrupt level "n" for INT0, clear
                            the request flag
```

Note3 : IIMC<bit 7 to 3> is always read as "1".

Note4 : See Electrical characteristics in section 4 for external 4 for external interrupt input pulse.

Setting of External Interrupt Pin Functions

Interrupt	Shared pin	Mode	Setting method
NMI	$\overline{\text{NMI}}$ (Dedicated pin)	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	P35	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 1
INT1	P40	Rising edge	———
INT4	P42	Rising edge	T4MOD<CAP12M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T4MOD<CAP12M1, 0> = 1, 0
INT5	P43	Rising edge	———
INT6	P45	Rising edge	T5MOD<CAP34M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T5MOD<CAP34M1, 0> = 1, 0
INT7	P46	Rising edge	———

(3) Micro DMA start vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector with each channel's Micro DMA start vector (bits 2 to 6 of the interrupt vector). When the two match, the interrupt from the channel whose value matched is processed in Micro DMA mode. If the interrupt vector matches more than one channel, the channel with the lower channel number has a higher priority.

Micro DMA0 Start Vector

DMA0V (007CH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol				DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
Read / Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 0 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA1 Start Vector

DMA1V (007DH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol				DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
Read / Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 1 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA2 Start Vector

DMA2V (007EH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol				DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
Read / Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 2 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA3 Start Vector

DMA3V (007FH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol				DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
Read / Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 3 processed by matching bits 2 to 6 of the interrupt vector.							

(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 "FFFF28H" and start the interrupt processing from the address "FFFF28H".
To avoid the above occurring, clear the interrupt request flag by entering the instruction to clear the flag after the DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.
In the case of changing the value of the interrupt mask register<IFF2:0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

3.5 Functions of Ports

The TMP93CS44 has 62 bits for I/O ports. The TMP93CS45 has 44 bits for I/O ports because Port0, Port1, P30, and P31 are dedicated pins for AD0 to 7, AD8 to 15 (or A8 to 15), $\overline{\text{RD}}$, and $\overline{\text{WR}}$.
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 (1) lists the function of each port pin. Table 3.5 (2) lists I/O registers and specification.

Table 3.5 (1) Functions of Ports

(PU = With programmable pull-up resistor)

Port Name	Pin Name	Pin No.	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	PU	Bit	A0 to A7 / A16 to A23
Port3	P30	1	Output	–	(fixed)	$\overline{\text{RD}}$
	P31	1	Output	–	(fixed)	$\overline{\text{WR}}$
	P32	1	I/O	PU	Bit	$\overline{\text{HWR}}$ / SCK
	P33	1	I/O	–	Bit	SO /SDA
	P34	1	I/O	–	Bit	SI / SCL
	P35	1	I/O	–	Bit	INT0
Port4	P40	1	I/O	–	Bit	TI0 / INT1
	P41	1	I/O	–	Bit	TO3
	P42	1	I/O	–	Bit	TI4 / INT4
	P43	1	I/O	–	Bit	TI5 / INT5
	P44	1	I/O	–	Bit	TO4
	P45	1	I/O	–	Bit	TI6 / INT6
	P46	1	I/O	–	Bit	TI7 / INT7
	P47	1	I/O	–	Bit	TO6
Port5	P50 to P52	3	Input	–	(fixed)	AN0 to AN2,
	P53	1	Input	–	(fixed)	AN3 / $\overline{\text{ADTRG}}$
	P54 to P57	4	Input	–	(fixed)	AN4 to AN7
Port6	P60	1	I/O	PU	Bit	TXD0
	P61	1	I/O	PU	Bit	RXD0
	P62	1	I/O	PU	Bit	SCLK0 / $\overline{\text{CTS0}}$
	P63	1	I/O	PU	Bit	TXD1
	P64	1	I/O	PU	Bit	RXD1
	P65	1	I/O	PU	Bit	SCLK1 / $\overline{\text{CTS1}}$
	P66	1	I/O	–	Bit	XT1
	P67	1	I/O	–	Bit	XT2
Port7	P70	1	I/O	–	Bit	$\overline{\text{WAIT}}$ / (High current output)
	P71 to P77	7	I/O	–	Bit	(High current output)

Table 3.5 (2) I/O registers and specification (1/2)

X : don’t care

Port	Name	Specification	I/O register		
			Pn	PnCR	PnFC
Port 0	P00 to P07	Input port (note 1)	×	0	None
		Output port (note 1)	×	1	
		AD0 to AD7) bus	×	×	
Port 1	P10 to P17	Input port (note m1)	×	0	0
		Output port (Note1)	×	1	0
		AD8 to AD15 bus (note 2)	×	0	1
		AD8 to AD15 output (note 2)	×	1	1
Port 2	P20 to P27	Input port (without pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	×	1	0
		A0 to A7 output (Note1)	1	0	1
		A16 to A23 output	1	1	1
Port 3	P30	Output port (Note1)	×	None	0
		Outputs $\overline{RD}$ only when accessing external space	1		1
		always outputs $\overline{RD}$	0		1
	P31	Output port (Note1)	×	None	0
		Outputs $\overline{WR}$ only when accessing external space	×		1
	P32	Input port SCK Input(without pull-up)	0	0	0
		Input port SCK Input(with pull-up)	1	0	0
		Output port	×	1	0
		$\overline{HWR}$ Output (<P32M> = 0)	×	1	1
		SCK Output (<P32M> = 1)	×	1	1
	P33	Input port	×	0	0
		Output port	×	1	0
		SDA I/O, SO Input	×	1	1
	P34	Input port	×	0	0
		SI Input	×	0	×
		Output port	×	1	0
		SCL I/O	×	1	1
	P35	Input port/INT0 Input (note 3)	×	0	None
		Output port	×	1	
Port 4	P40	Input port/TI0/INT	×	0	None
		Output port	×	1	
	P41	Input port	×	0	0
		Output port	×	1	0
		TO3 Output	×	1	1
	P42	Input port / T14 / INT4 Input	×	0	None
		Output port	×	1	
	P43	Input port / T15 / INT5 Input	×	0	
		Output port	×	1	

Note 1 : In the TMP93CS45, these functions are not available.

Note 2 : In the TMP93CS45, these functions are fixed depending on the value of the AM8/$\overline{16}$ pin.

Note 3 : Using P35 pin as INT0, IIMC register has to be set enable interrupt.

Port	Name	Specification	I/O register		
			Pn	PnCR	PnFC
Port 4	P44	Input port	×	0	0
		Output port	×	1	0
		TO4 Output	×	1	1
	P45	Input port / T16 / INT6 Input	×	0	None
		Output port	×	1	
	P46	Input port / T17 / INT7 Input	×	0	
		Output port	×	1	
	P47	Input port	×	0	0
		Output port	×	1	0
		TO6 Output	×	1	1
Port 5	P50 to P57	Input port	×	None	
		AN0 to AN7 Input (note 4)	×		
Port 6	P60	Input port (without Pull-up)	×	0	0
		Input port (with Pull-up)	×	0	0
		Output port	0	1	0
		TxD0 Output	1	1	1
	P61	Input port / RxD0 Input (without Pull-up)	0	0	None
		Input port / RxD0 Input (with Pull-up)	1	0	
		Output port	×	1	
	P62	Input port / SCLK0 / $\overline{\text{CTS0}}$ Input (without Pull-up)	0	0	0
		Input port / SCLK0 / $\overline{\text{CTS0}}$ Input (with Pull-up)	1	0	0
		Output port	×	1	0
		SCLK0 Output	×	1	1
	P63	Input port (without Pull-up)	0	0	0
		Input port (with Pull-up)	1	0	0
		Output port	×	1	0
		TxD1 Output (note 3)	×	1	1
	P64	Input port / RxD1 input (without Pull-up)	0	0	None
		Input port / RxD1 input (with Pull-up)	1	0	
		Output port	×	1	
	P65	Input port / SCLK1 / $\overline{\text{CTS1}}$ Input (without Pull-up)	0	0	0
		Input port / SCLK1 / $\overline{\text{CTS1}}$ Input (with Pull-up)	1	0	0
		Output port	×	1	0
		SCLK1 Output	×	1	1
	P66, p67	Input port	×	0	None
		Output port (note 5)	×	1	
		XT1/2 (note 6)	×	0	
Port 7	P70	Input port / $\overline{\text{WAIT}}$ Input	×	0	None
		Output port	×	1	
	P71 to P77	Input port	×	0	
		Output port	×	1	

Note 4 : Using P50 to P57 pins as input channels for the A/D converter, the channels are selected by ADMOD1<ADCH2 to 0>.

Note 5 : Using P66 and P67 pins as the output ports, output is through the open-drain buffer.

Note 6 : Using P66 and P67 pins as the XT1 to XT2, oscillation is enabled by the SYSCR0 register.

Resetting makes the port pins listed below function as general-purpose I/O ports.
I/O pins programmable for input or output are set to input ports except P66/XT1, P67/XT2.
To set port pins for built-in functions, a program is required.

Since the TMP93CS45 needs external ROMs, some ports are permanently assigned for memory-interface.

- P00 to P07 → AD0 to AD7
- P10 to P17 → AD8 to AD15 (or A8 to A15)
- P30 → $\overline{RD}$
- P31 → $\overline{WR}$

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using the control register P0CR. Resetting sets all bits of P0CR to 0 and sets Port 0 to input mode. Figure 3.5 (3) shows the registers for Port 0.

In addition to functioning as a general-purpose I/O port, Port 0 also shares functions as an address data bus (AD0 to 7). To access external memory, Port 0 functions as an address data bus (AD0 to 7) and all bits of the control register P0CR are set to 0.

With the TMP93CS45, 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.

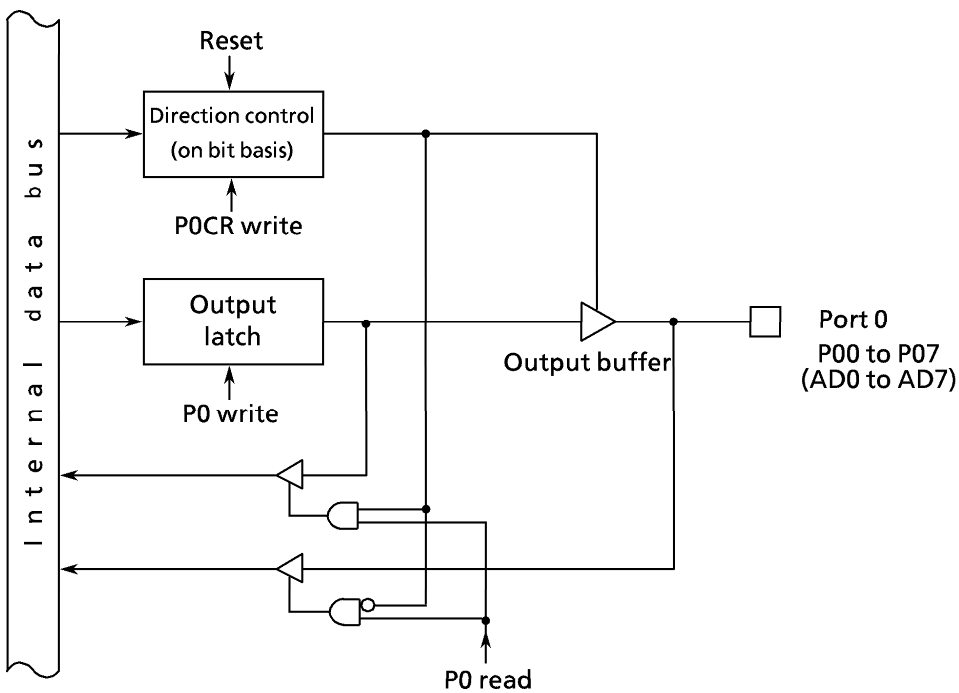


Figure 3.5 (1) Port 0

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 sets all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets Port 1 to input mode.
Figure 3.5 (3) shows the registers for Port 1.
In addition to functioning as a general-purpose I/O port, Port 1 also shares functions as an address data bus (AD8 to 15) or an address bus (A8 to 15).
With the TMP93CS45, which needs external ROMs, Port 1 always functions as an address data bus (AD8 to 15) (the case of AM8 / $\overline{16}$ = "0"), as an address bus (A8 to 15) (the case of AM8 / $\overline{16}$ = "1") regardless of the value set in control register P1CR.

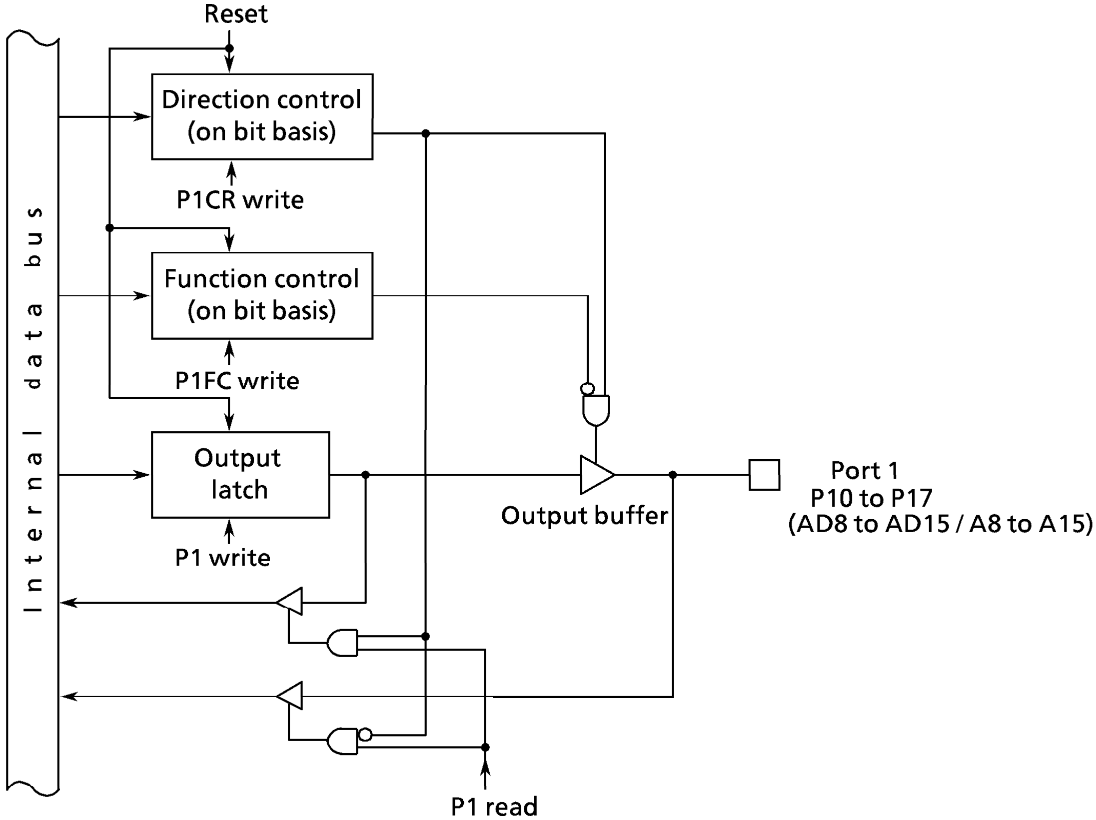


Figure 3.5 (2) Port 1

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)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0:IN 1:OUT (At external access, Port 0 becomes AD7 to 0 and P0CR is set 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 set to "0".)							

Port 1 Control Register

P1CR
(0004H)
Prohibit read-modify-write

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

Port 1 Function Register

P1FC
(0005H)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00 : 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)

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

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. All bits of the output latch P2 is set to "1" by reset, and all bits of P2CR and P2FC are cleared to "0". Port 2 becomes the input mode with the pull-up resistor.

In addition to functioning as a general-purpose I/O port, Port 2 also shares functions as an address data bus (A0 to 7) and an address bus (A16 to 23). Using Port 2 as address bus (A0 to 7 or A16 to 23), write "0" to output latches and be off the programmable pull-up resistor.

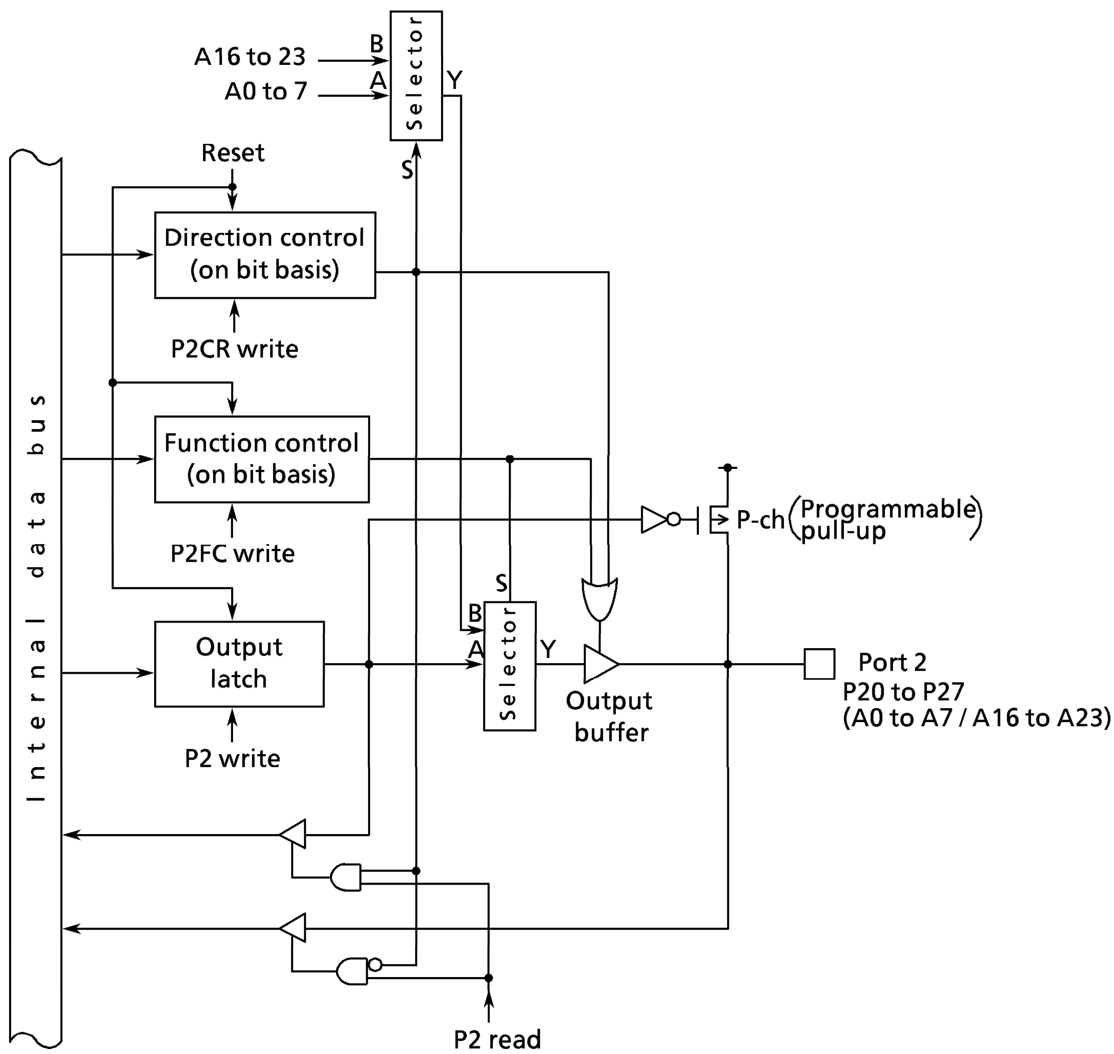


Figure 3.5 (4) Port 2

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 set to "1".)							

Note 1 : When port 2 is used in the input mode, P2 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode.
Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 2 Control Register

P2CR
(0008H)
Prohibit read-modify-write

	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)
Prohibit read-modify-write

	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

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

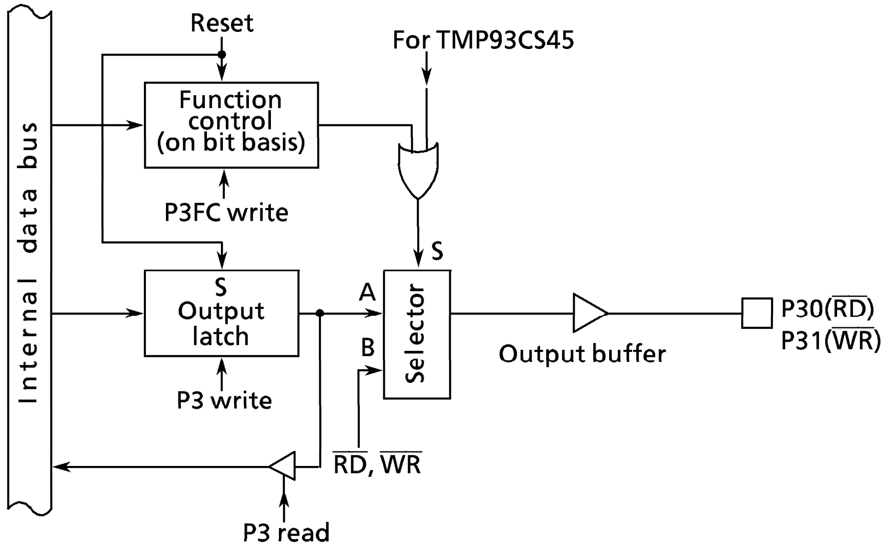
Note 2 : <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR.
To set as an address bus A23 to 16, set P2FC after setting P2CR.

Figure 3.5 (5) Registers for Port 2

3.5.4 Port 3 (P30 to P35)

Port 3 is an 6-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 sets all bits of output latch P3 to 1. All bits of control register P3CR (bits 0 and 1 are unused), and function register P3FC are set to 0. Resetting also outputs 1 from P30 and P31.
In addition to functioning as a general-purpose I/O port, Port 3 also shares functions as an I/O for the CPU's control / status signal and serial bus interface.
With the TMP93CS44, when P30 pin is defined as $\overline{RD}$ signal output mode (<P30F> = 1), setting 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.
When P33 and P34 are used as the serial bus interface I/O pins in I^{2}C bus mode (P3FC<P34F, P33F> = "11"), set open drain outputs (ODE<ODE34, 33> = "11").
With the TMP93CS45, which needs external ROMs, P30 outputs the $\overline{RD}$ signal; P31, the $\overline{WR}$ signal, regardless of the values set in function registers <P30F> and <P31F>.
The $\overline{RD}$ signal is output not only when the external address area is accessed at <P30> = "1" but also the internal address area is accessed at <P30> = "0".

(1) P30 ($\overline{RD}$), P31 ($\overline{WR}$)



(2) P32 ($\overline{HWR}$ / SCK)

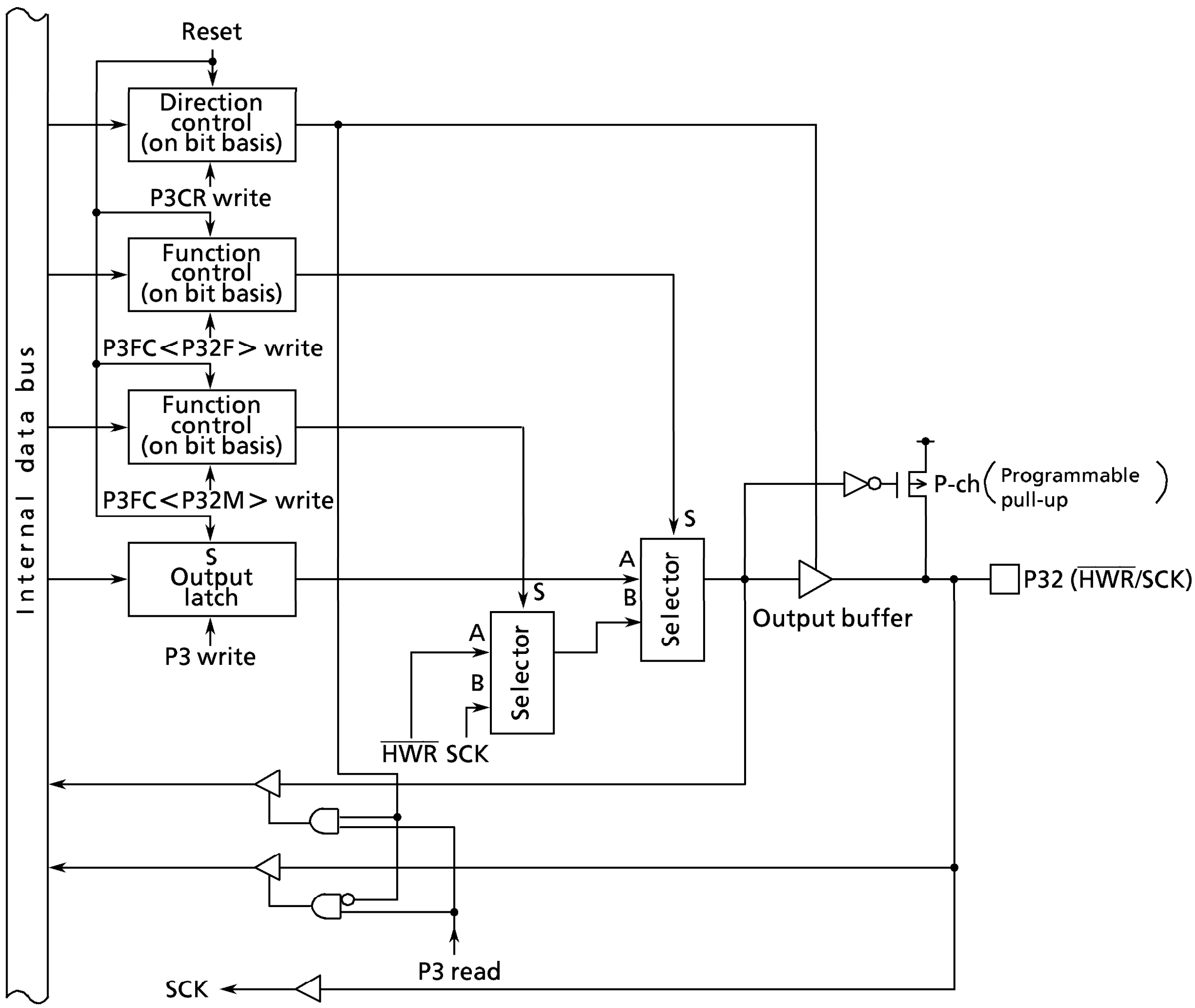


Figure 3.5 (6) Port 3 (P30, P31, P32)

(3) P33 (SDA / SO), P34 (SCL / SI)

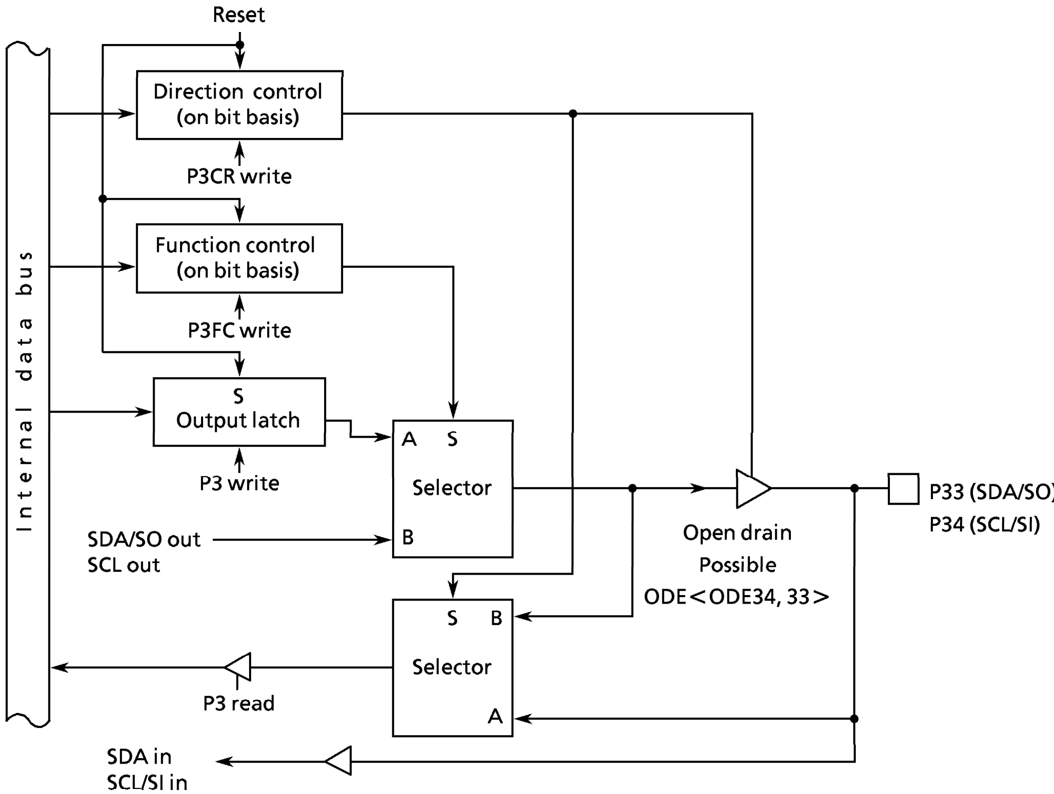


Figure 3.5 (7) Port3 (P33, P34)

(4) P35 (INT0)

Port 35 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

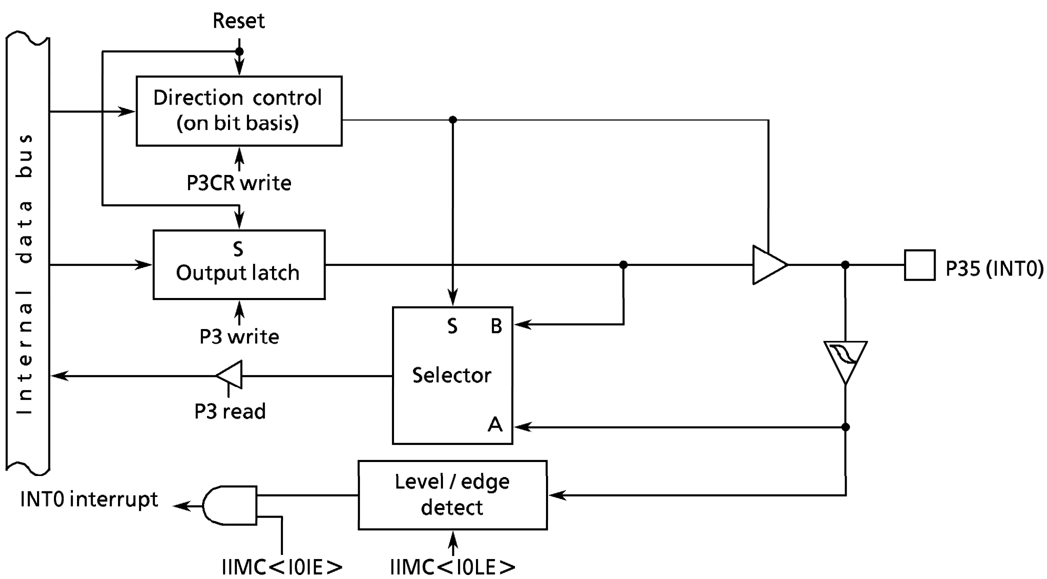


Figure 3.5 (8) Port 35

Port 3 Register

P3 (0007H)

	7	6	5	4	3	2	1	0
bit Symbol			P35	P34	P33	P32	P31	P30
Read / Write			R/W					
After reset			Input mode			Input mode (pulled-up)	Output mode	
			1	1	1	1	1	1

Note 1 : When port 32 is used in the input mode, P3 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode.
Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 3 Control Register

P3CR (000AH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			P35C	P34C	P33C	P32C		
Read / Write			W					
After reset			0	0	0	0		
Function			0 : IN 1 : OUT					

I/O setting

0	Input
1	Output

Port 3 Function Register

P3FC (000BH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol		P32M		P34F	P33F	P32F	P31F	P30F
Read / Write		W		W				
After reset		0		0	0	0	0	0
Function		0 : $\overline{HWR}$ 1 : SCK		0 : PORT 1 : SCL/SI	0 : PORT 1 : SDA/SO	0 : PORT 1 : P32M	0 : PORT 1 : $\overline{WR}$	0 : PORT 1 : $\overline{RD}$

P34 function setting

<P34F> \ <P34C>	0	1
0	Input port	Output port
1	Input SI	I/O SCL

P33 function setting

<P33F> \ <P33C>	0	1
0	Input port	Output port
1	Don't set	I/O SDA /Output SO

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

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

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

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

P32 function setting

<P32F> \ <P32M>	0	1
0	Port	
1	Output "$\overline{HWR}$"	input / Output "SCK"

Note 2 : When P33 and P34 are used as the serial bus interface I/O pins in I²C bus mode (P3FC<P34F, P33F> = "11"), set open drain outputs (ODE<ODE34, 33> = "11").

Figure 3.5 (8) Registers for Port 3

3.5.5 Port 4 (P40 to P47)

Port 4 is a 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 4 to the input port. In addition to functioning as a general-purpose I/O port, Port 4 also shares functions as an input for 8-bit timer 0 clock, 16-bit timer 4 and 5 clocks, an output for 8-bit timer F/F 3, 16-bit timer F/F4 and 6 output. Writing 1 in the corresponding bit of the Port 4 function register (P4FC) enables output of the timer.

(1) P40, P41

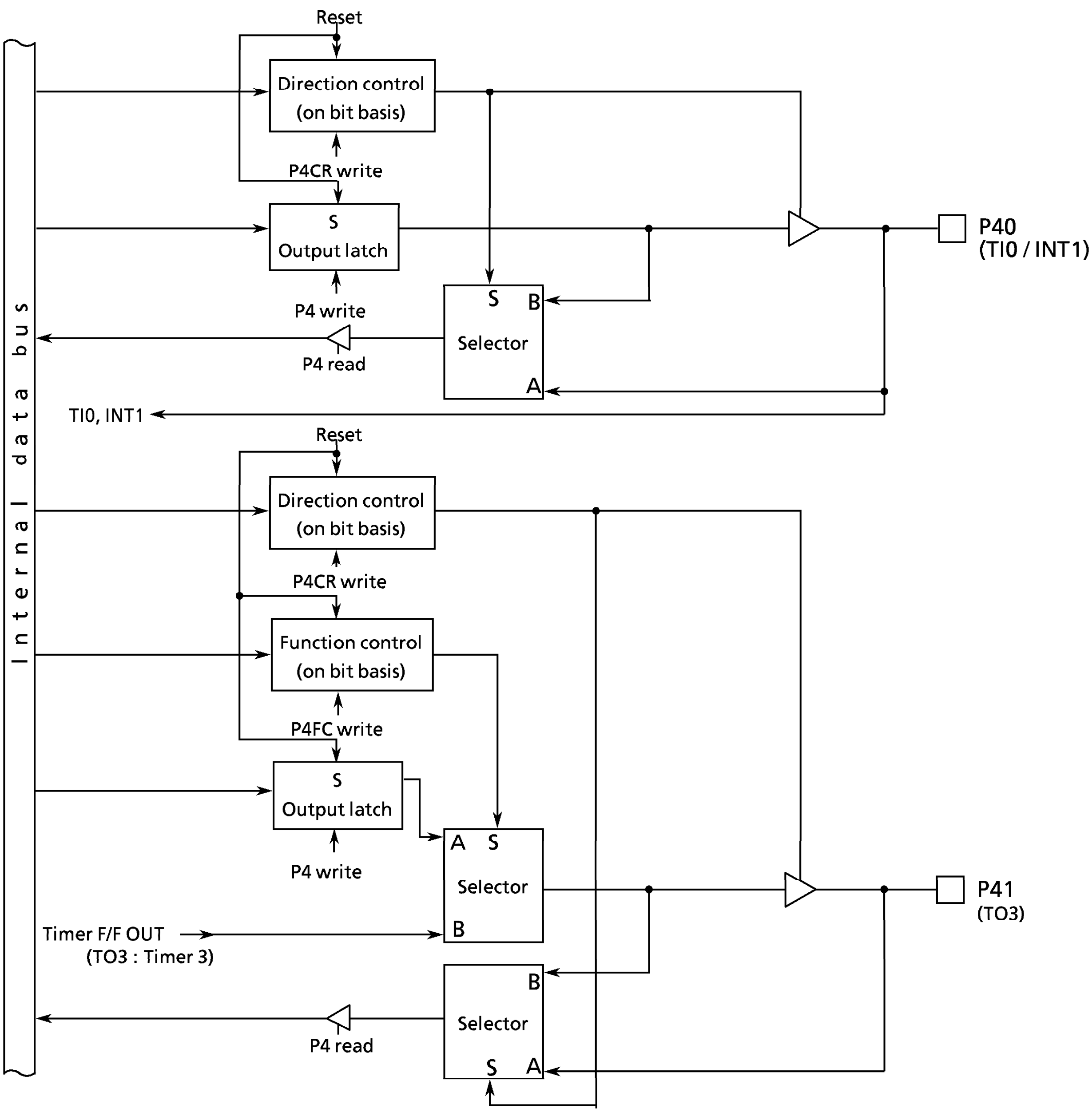


Figure 3.5 (9) Port 4 (P40, P41)

(2) P42 to P47

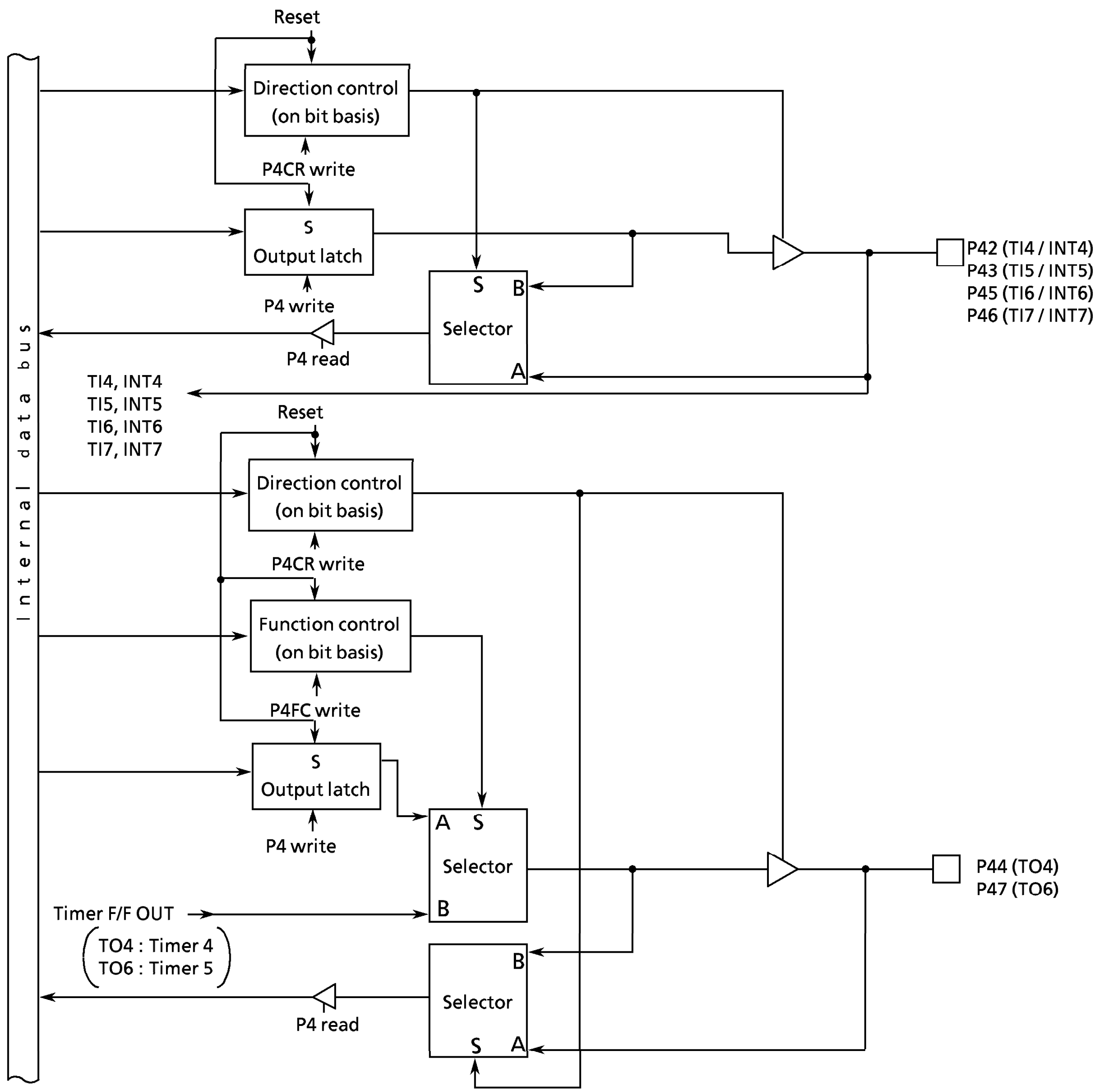


Figure 3.5 (10) Port 4 (P42 to P47)

Port 4 Register

P4
(000CH)

	7	6	5	4	3	2	1	0
bit Symbol	P47	P46	P45	P44	P43	462	P41	P40
Read / Write	R/W							
After reset	Input mode							
	1	1	1	1	1	1	1	1

Port 4 Control Register

P4CR
(000EH)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port 4 I/O setting

0	Input
1	Output

Port 4 Function Register

P4FC
(0010H)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	P47F			P44F			P41F	
Read / Write	W			W			W	
After reset	0			0			0	
Function	0 : PORT 1 : TO6			0 : PORT 1 : TO4			0 : PORT 1 : TO3	

Setting P41 as TO3

P4FC <P41F>	1
P4CR <P41C>	1

Setting P44 as TO4

P4FC <P44F>	1
P4CR <P44C>	1

Setting P47 as TO6

P4FC <P47F>	1
P4CR <P47C>	1

Note) P40/TI0, P42/TI4, P43/TI5, P45/TI6, P46/TI7 pin does not have a register changing PORT / FUNCTION.
For example, when it is used as an input port, the input signal for port is inputted to 8/16 bit Timer as a timer input.

Figure 3.5 (11) Register 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. Additionally, P53 is also used as an analog conversion external trigger input pin ($\overline{\text{ADTRG}}$).

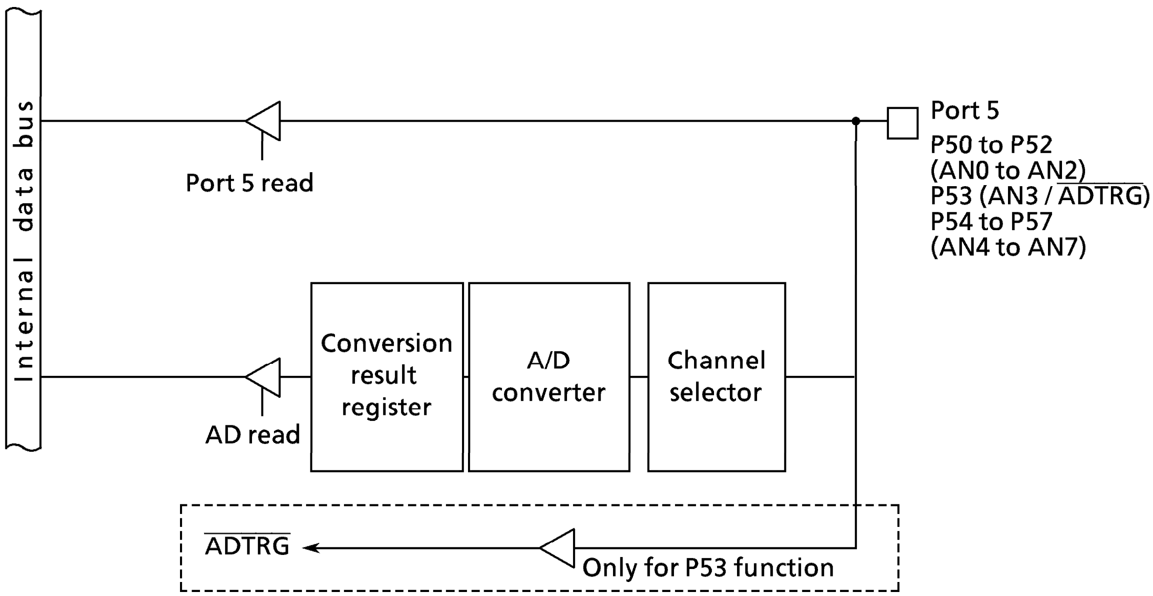


Figure 3.5 (12) Port 5

Port 5 Register

P5
(000DH)

	7	6	5	4	3	2	1	0
bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
Read / Write	R							
After reset	Input mode							

Note) The input channel selection of A/D Converter is set by A/D Converter mode register ADMOD1.

Figure 3.5 (13) Registers for Port 5

3.5.7 Port 6 (P60 to P67)

Port 60 to 65 is a 6-bit general-purpose I/O port. I/Os can be set on a bit basis.
Resetting sets P60 to 65 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, P60 to 65 can also share function as an I/O for serial channels 0 and 1. Writing "1" in the corresponding bit of the Port 6 function register (P6FC) enables this function.
Resetting sets the function register value to '0' and sets all bits to input ports.

Port 66, 67 is a 2-bit general-purpose I/O port. I/Os can be set on a bit basis.
The output buffer for P66, 67 is an open drain type buffer.
Resetting outputs high-impedance (Hi-Z) because output latch and control register are set to "1".
In addition to functioning as a general-purpose I/O port, P66, 67 can also function as a low-frequency oscillator connecting pin (XT1, XT2) for dual clock mode. The dual clock function can be set by programming system clock control registers SYSCR0, 1.

(1) Port 60 (TXD0), 63 (TXD1)

Ports 60 and 63 also function as serial channel TXD output pins in addition to I/O ports.
They have a programmable open drain function.

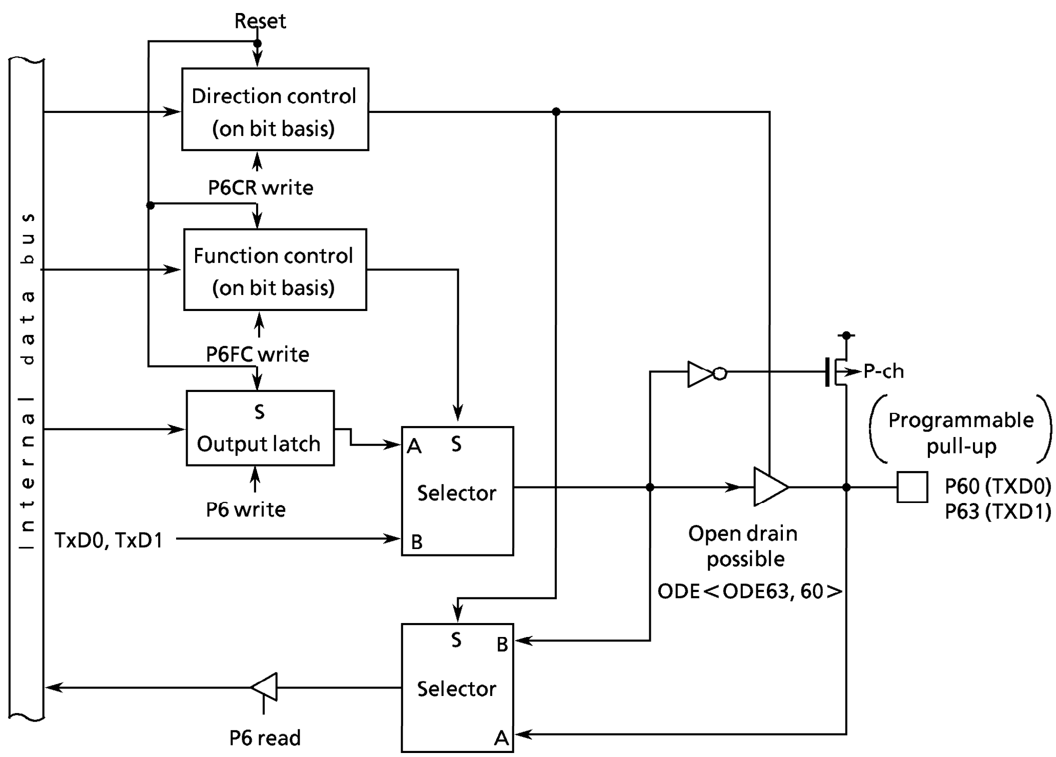


Figure 3.5 (14) Ports 60 and 63

(2) Port 61 (RXD0), 64 (RXD1)

Port 61 and 64 are I/O ports, and also used as RXD input pins for serial channels.

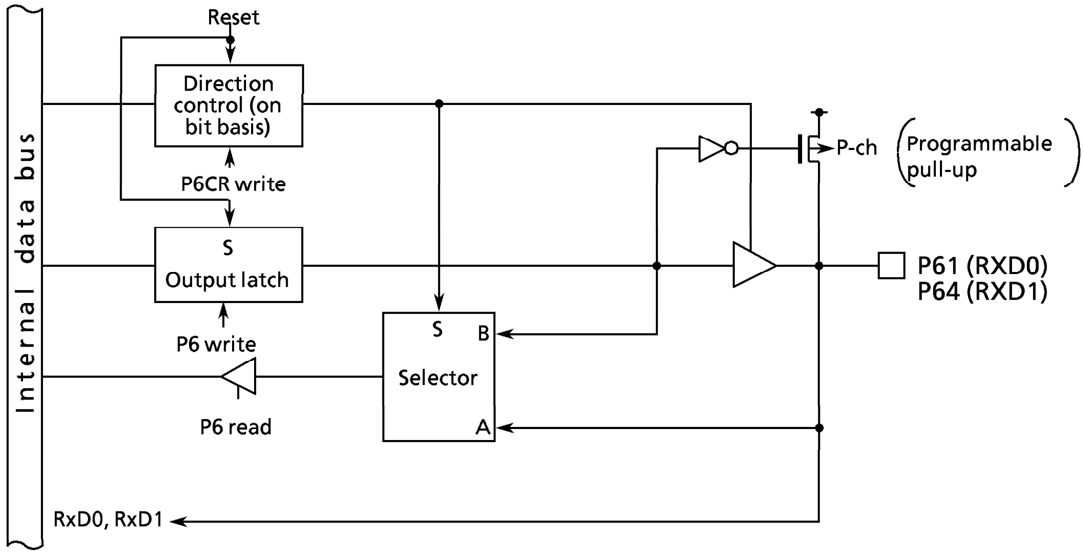


Figure 3.5 (15) Ports 61 and 64

(3) Port 62 ($\overline{\text{CTS0}}$ / SCLK0), 65 ($\overline{\text{CTS1}}$ / SCLK1)

Port 62, 65 is an I/O port, and also used as a $\overline{\text{CTS}}$ input pin and as a SCLK I/O pin for serial channels.

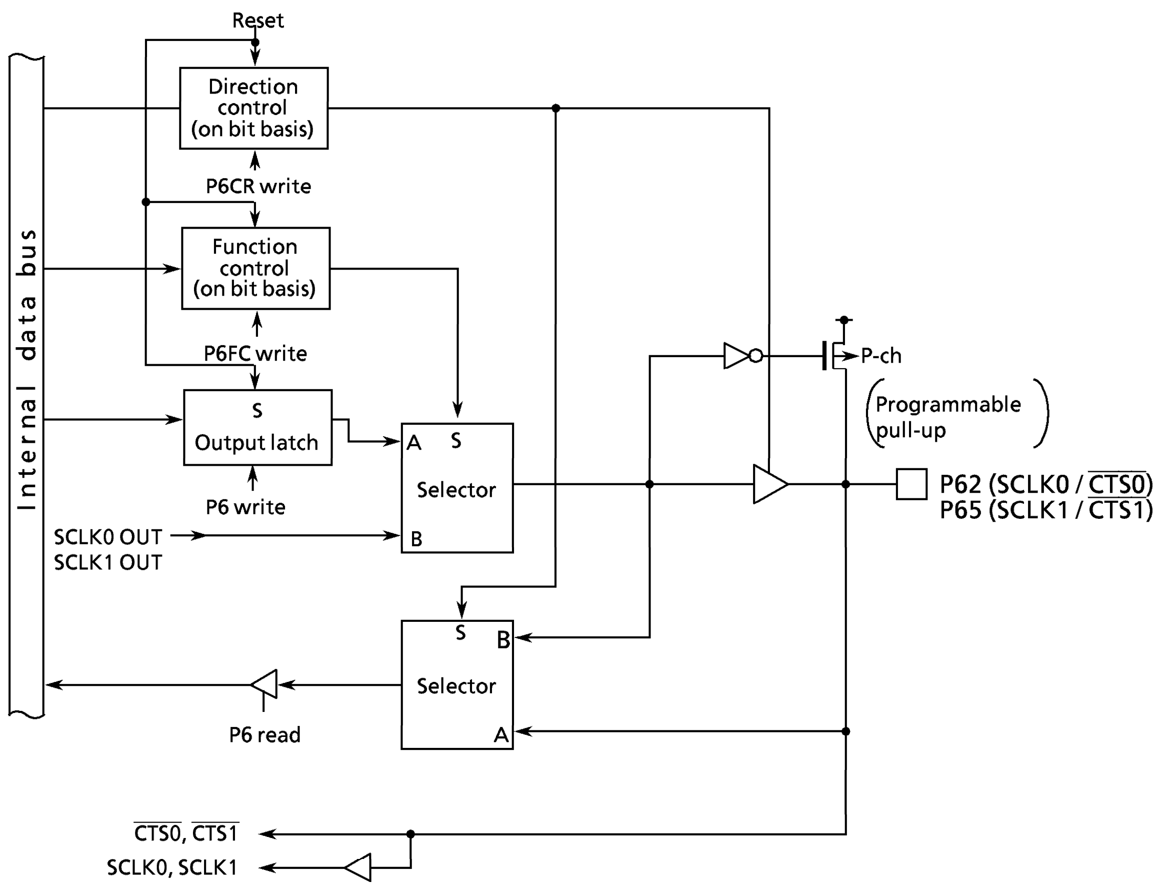


Figure 3.5 (16) Port 62, 65

(4) Port 66 (XT1), 67 (XT2)

Port 66, 67 is general purpose I/O ports. It is also used as a low-frequency oscillator connecting pin.

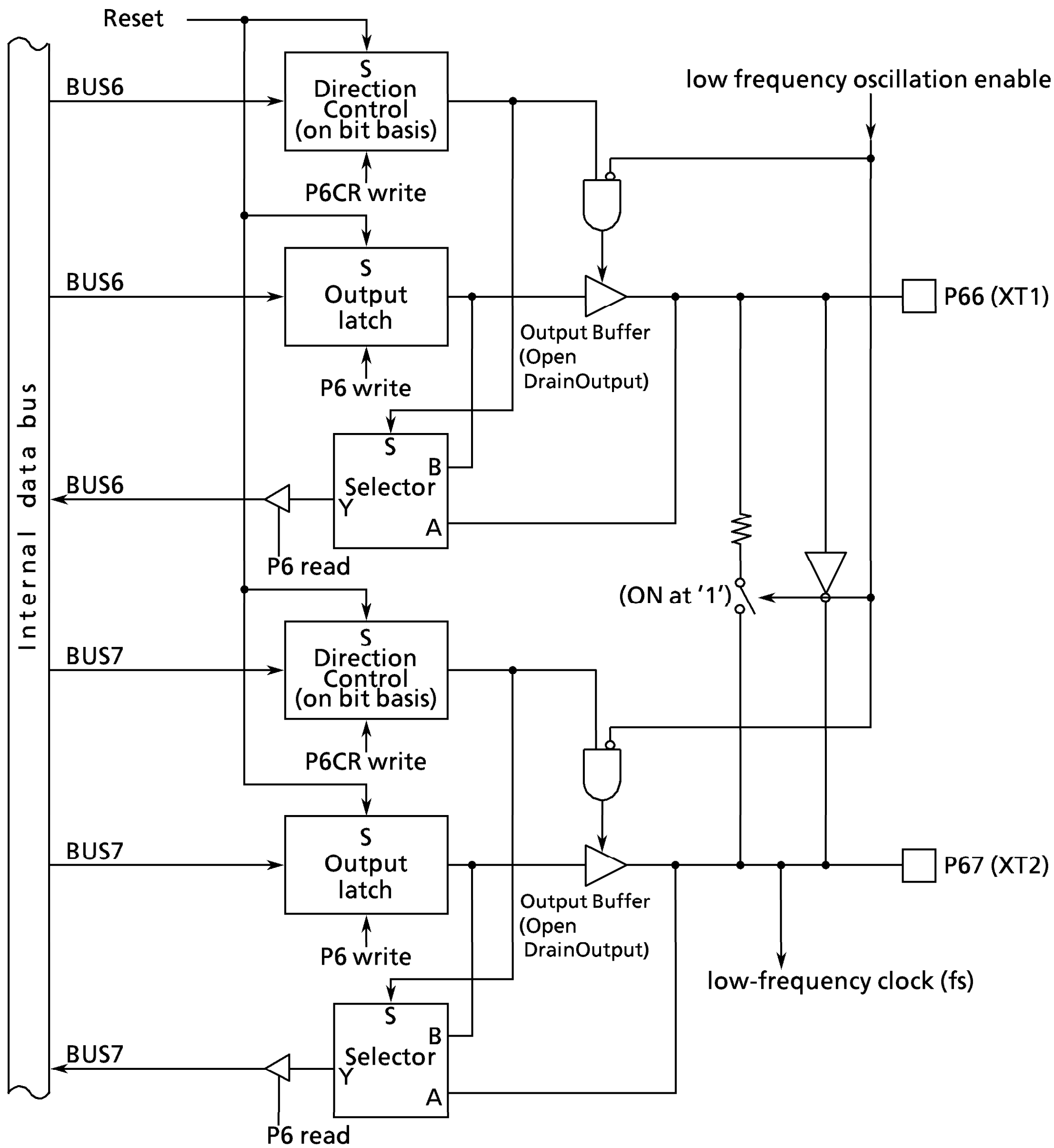


Figure 3.5 (17) Port 66 to 67

Port 6 Register

P6
(0012H)

	7	6	5	4	3	2	1	0
bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
Read / Write	R/W							
After reset	Output mode		Input mode					
	1	1	1	1	1	1	1	1

Note 1 : When port P6 is used in the input mode, P6 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 6 Control Register

P6CR
(0014H)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
Read / Write	W							
After reset	1	1	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port 6 I/O setting

0	Input
1	Output

Note) Port 66, 67's output buffer is an open drain output type.

Port 6 Function Register

P6FC
(0016H)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			P65F		P63F	P62F		P60F
Read / Write			W		W			W
After reset			0		0	0		0
Function			0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0		0 : PORT 1 : TxD0

P60 TxD0 output setting (Note)

P6FC <P60F>	1
P6CR <P60C>	1

P62 SCLK0 output setting

P6FC <P62F>	1
P6CR <P62C>	1

P63 TxD1 output setting (Note)

P6FC <P63F>	1
P6CR <P63C>	1

P65 SCLK1 output setting

P6FC <P65F>	1
P6CR <P65C>	1

Note 2 : To set the TxD pin to open drain, write "1" in bit 0 (for TxD0 pin) or bit 1 (for TxD1 pin) of the ODE register.
P61 / RXD0, P64 / RXD1 pins do not have a register changing PORT / FUNCTION.
When using as input ports, the serial receive data is input to SIO.

Note 3 : Notes on using low-frequency oscillation circuit. To connect a low frequency resonator to port 66, 67, it is necessary to set the following procedures to reduce the consumption power supply.

(connecting to a resonator)
Set P6CR<P66C, P67C> = "11", P6<P66, P67> = "00"

(connecting to an oscillator)
Set P6CR<P66C, P67C> = "11", P6<P66, P67> = "10"

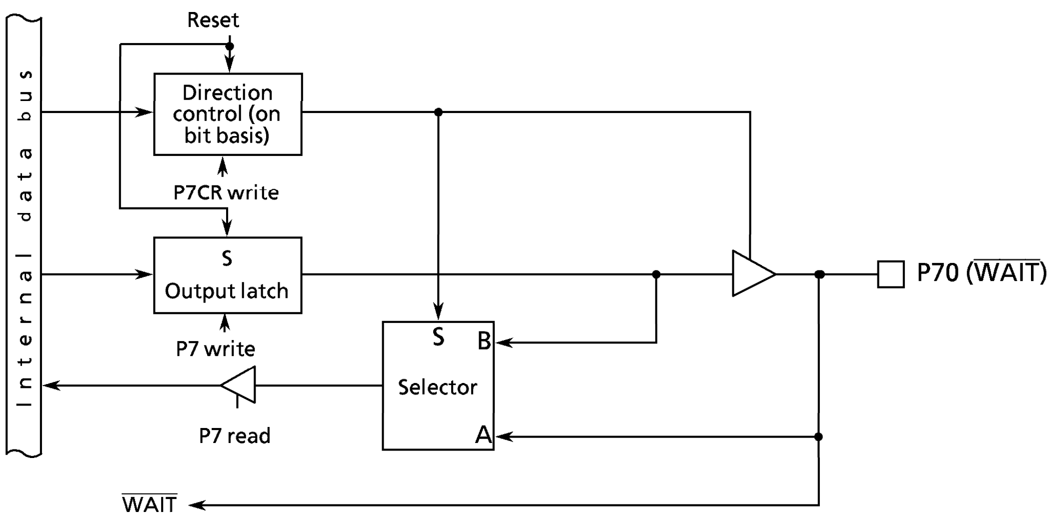
Figure 3.5 (18) Register for Port 6

3.5.8 Port 7 (P70 to P77)

Port 7 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Port 7 can output large current and drive LED directly. In addition to I/O port, Port 70 also shares functions as $\overline{\text{WAIT}}$ input pin. Resetting sets the function register P7CR to 0, and all bits to input ports. Port 7 as an input port. It also sets all bits of the output latch register P7 to 1.

(1) P70 ($\overline{\text{WAIT}}$)

Port 70 is a general-purpose I/O port, and also used as an $\overline{\text{WAIT}}$ pin for external wait input.



(2) P71 to P77

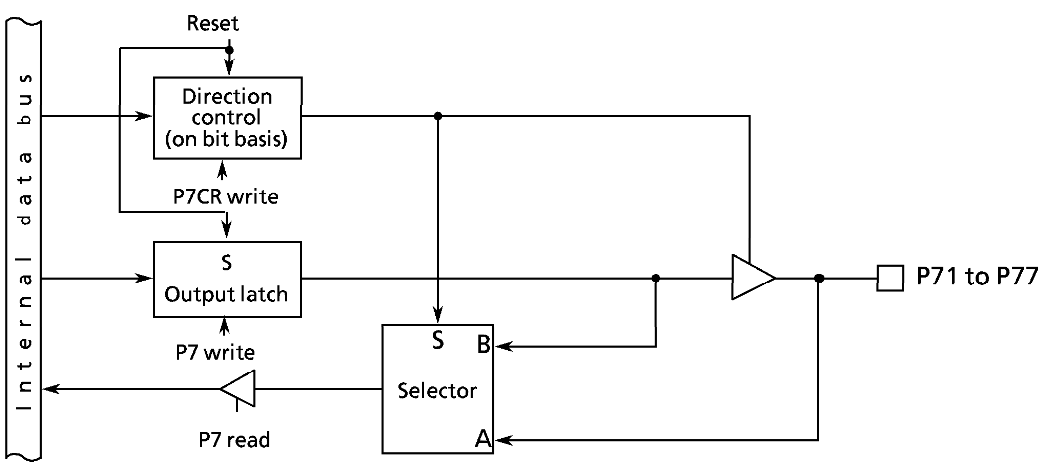


Figure 3.5 (19) Port 7

Port 7 Register

P7
(0013H)

	7	6	5	4	3	2	1	0
bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read / Write	R/W							
After reset	Input mode							
	1	1	1	1	1	1	1	1

Port 7 Control Register

P7CR
(0015H)

Prohibit read-modify-write.

	7	6	5	4	3	2	1	0
bit Symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port 7 I/O setting

0	Input
1	Output

Note : P70/$\overline{\text{WAIT}}$ pin does not have a register changing PORT/FUNCTION.
For example, when it is used as and input port, the input signal is inputted as $\overline{\text{WAIT}}$ input.
When it is used as $\overline{\text{WAIT}}$ input pin, bit <BmWn> of Bus Width WAIT control register must be specified.

Figure 3.5 (20) Registers for Port 7

3.6 Bus Width / Wait Controller, AM8 / $\overline{16}$ pin

TMP93CS44 / S45 have a built-in controller used to control wait ($\overline{\text{WAIT}}$ 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 TMP93CS45.

3.6.1 AM8 / $\overline{16}$ pin

(1) TMP93CS44

Set this pin to “H”. After reset, the CPU accesses the internal ROM with 16 bit bus width. The bus width when the CPU accesses an external area is set by Chip Select / Wait Control Register (described at 3.6.2) and the registers of Port 1. (The value “1” of this pin is ignored and the value set by register is active)

(2) TMP93CS45

① With fixed external 16-bit data bus and external 16-bit data bus or 8-bit data bus is selectable

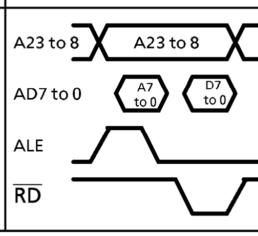
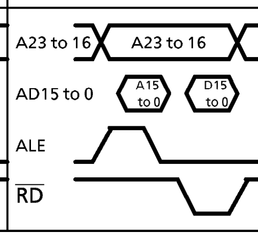
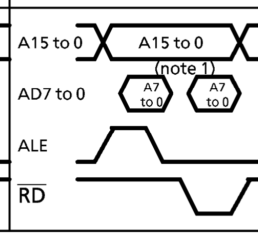
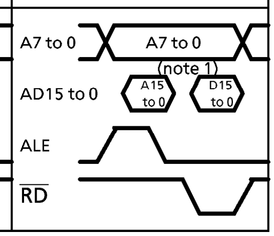
Set this pin to “L”. Port1, AD8 to 15 and A8 to 15 pins are fixed to AD8 to 15 functions. The values set in Port 1 control register and Port 1 function register are invalid.
The external data bus width is set by the chip select / wait control register which is described in section 3.6.2.
It is necessary to set the program memory to be accessed to 16-bit data bus after reset.

② With fixed external 8-bit data bus

Set this pin to “H”. Port1, AD8 to 15 and A8 to 15 pins are fixed to A8 to 15 functions. The values set in Port 1control register and Port 1 function register are invalid.
The values of bit 4 : <B0BUS>, <B1BUS>, and <B2BUS>, in the chip select / wait control register described in section 3.6.2 are invalid. The external 8-bit data bus is fixed.

3.6.2 Address / Data bus pins

Port 0/AD0 to 7, Port 1/AD8 to 15 and Port 2/AD16 to 23/A0 to 7 function as address / data bus for connecting the external memories and so on.

		①	②	③	④
Products		TMP93CS45F (note4)		TMP93CS44F (note2), (note3)	
Number of address bus pins		max24 (to 16MB)	max24 (to 16MB)	max16 (to 64KB)	max8 (to 256B)
Number of data bus pins		8	16	8	16
Number of multiplexed pins		8	16	0	0
Mode pins	$\overline{EA}$	V_{IL}		V_{IH}	
	AM8/$\overline{16}$	V_{IH}	V_{IL}	V_{IH}	
Port function	Port 0	AD0 to 7	AD0 to 7	AD0 to 7	AD0 to 7
	Port 1	A8 to 15	AD8 to 15	A8 to 15	AD8 to 15
	Port 2	A16 to 23	A16 to 23	A0 to 7	A0 to 7
Timing chart					

Note 1 : In case of ③ and ④, the data bus signals output the addresses since the signals are also used as the address bus. Writing "0"to bit CKOCR<ALEEN>, ALE signal can be stopped outputting.

Note 2 : After reset operation, POrt 0, Port 1 and Port 2 of TMP93CS44F function as Input ports.

Note 3 : In case of TMP93CS44F, All ① to ④ can be available using P1CR, P1FC, P2CR and P2FC registers.

Note 4 : In case of TMP93CS45F, Case ③ and ④ cannot be available.

3.6.3 Bus Width / Wait Control Registers

Table 3.6.(1) shows control registers.
One block address areas are controlled by 1-byte Bus-width / WAIT control registers (WAITC0, WAITC1, WAITC2).

(1) Data bus size select

Bit 4 (<B0BUS>, <B1BUS>, <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 (1) shows the details of the bus operation.
This bit is changed by the state of AM8 / $\overline{16}$ pin.

(2) Wait control

Control register bits 3 and 2 (<B0W1,0>, <B1W1,0>, <B2W1,0>) are used to specify the number of waits.
These bits execute the following operation by setting.
"00" A 2-state wait is inserted regardless of the WAIT pin status.
"01" A 1-state wait is inserted regardless of the WAIT pin status.
"10" A 1-state wait is inserted and the WAIT pin starus is sampled. If the pin is low, inserting the wait maintains the bus cycle until the pin goes high.
"11" The bus cycle is completed without a wait (0 WAIT) regardless of the WAIT pin status.

(3) 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 (Wait state, Bus size, etc.) as follows:

* WAITC0 setting enabled when 7F00H to 7FFFH is accessed.
* WAITC1 setting enabled when 880H to 7FFFH is accessed.
* WAITC2 setting enabled when 8000H to 3FFFFFH is accessed.

Setting bits to 01 enables setting for each block 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.

Register		7	6	5	4	3	2	1	0
WAITC0 (0068H) Prohibit read-modify-write.	bit Symbol				B0BUS	B0W1	B0W0	B0C1	B0C0
	Read/Write				W				
	After reset				0	0	0	0	0
	Function				0:16bit Bus 1:8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 7F00H to 7FFFH 01: From 400000H 10: From 800000H 11: From C00000H	
WAITC1 (0069H) Prohibit read-modify-write.	bit Symbol				B1BUS	B1W1	B1W0	B1C1	B1C0
	Read/Write				W				
	After reset				0	0	0	0	0
	Function				0:16bit Bus 1:8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 880H to 7FFFH 01: From 400000H 10: From 800000H 11: From C00000H	
WAITC2 (006AH) Prohibit read-modify-write.	bit Symbol				B2BUS	B2W1	B2W0	B2C1	B2C0
	Read/Write				W				
	After reset				0	0	0	1	1
	Function				0:16bit Bus 1:8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: From 8000H 01: From 400000H 10: From 800000H 11: From C00000H	

Table 3.6 (1) Bus- width / wait control register

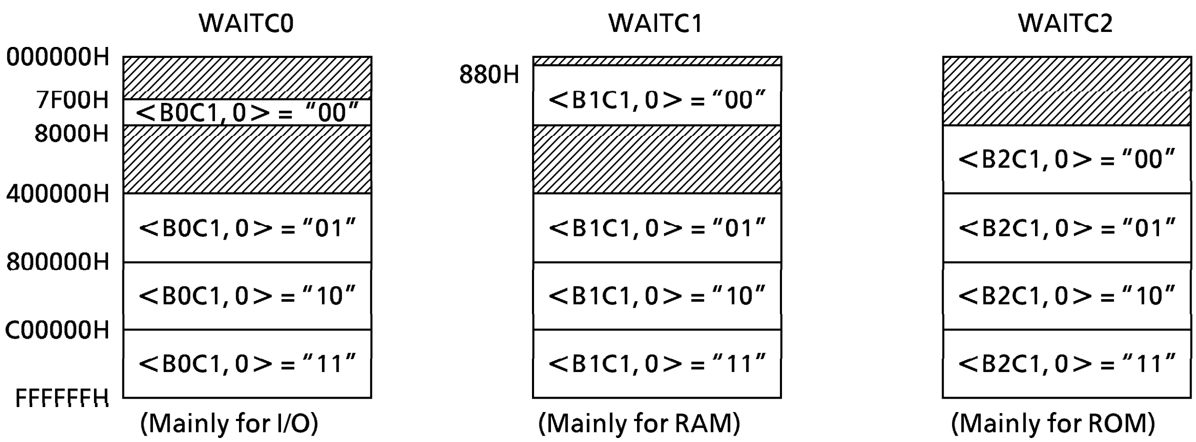
Table 3.6 (1) Dynamic bus sizing

Operand data size	Operand start address	Memory data size	CPU address	CPU data D15 to D8	CPU data D7 to D0
8 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

xxxxx : 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.

3.6.4 Bus-width / Wait controll

An image of the actual Bus-width / Wait control 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 WAITC0; 880H to 7FFFH, for WAITC1; and 8000H to 3FFFFFH, for WAITC2. The reason is that a device other than ROM (ie, RAM or I/O) might be connected externally.
7F00 to 7FFFH (256 bytes) for WAITC0 are mapped mainly for possible expansions to external I/O.
880H to 7FFFH (approx. 31K bytes) for WAITC1 are mapped there mainly for possible extensions to external RAM.
8000H to 3FFFFFH (approx. 4M bytes) for WAITC2 are mapped mainly for possible extensions to external ROM. With the TMP93CS45, which does not have a built-in ROM, the program is externally read at address FF0000H in this setting (16-bit bus, 2-wait). With the TMP93CS44 which has a built-in ROM, addresses from FF0000H to FFFFFFH are used as the internal ROM area; WAITC2 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 WAITC0 to 2 are accessed in 0 wait mode. In the TMP93CS45, when the $AM8/\overline{16}$ pin is set to "L", the data bus width is fixed to 16-bit. When the $AM8/\overline{16}$ pin is set to "H", it is fixed to 8-bit. In the TMP93CS44, the data bus width is always fixed to 16-bit.
When using the chip select/wait controller, do not specify the same address area more than once. (However, when addresses 7F00H to 7FFFH for WAITC0 and 880H to 7FFFH for WAITC1 are specified, in other words, specifications overlap, only the WAITC0 setting is active.)

3.6.5 Example of Usage

(1) Example of Usage -1

Figure 3.6 (2) is an example in which an external memory is connected to the TMP93CS45. In this example, a ROM is connected using 16 bit Bus; a RAM is connected using 8 bit Bus.

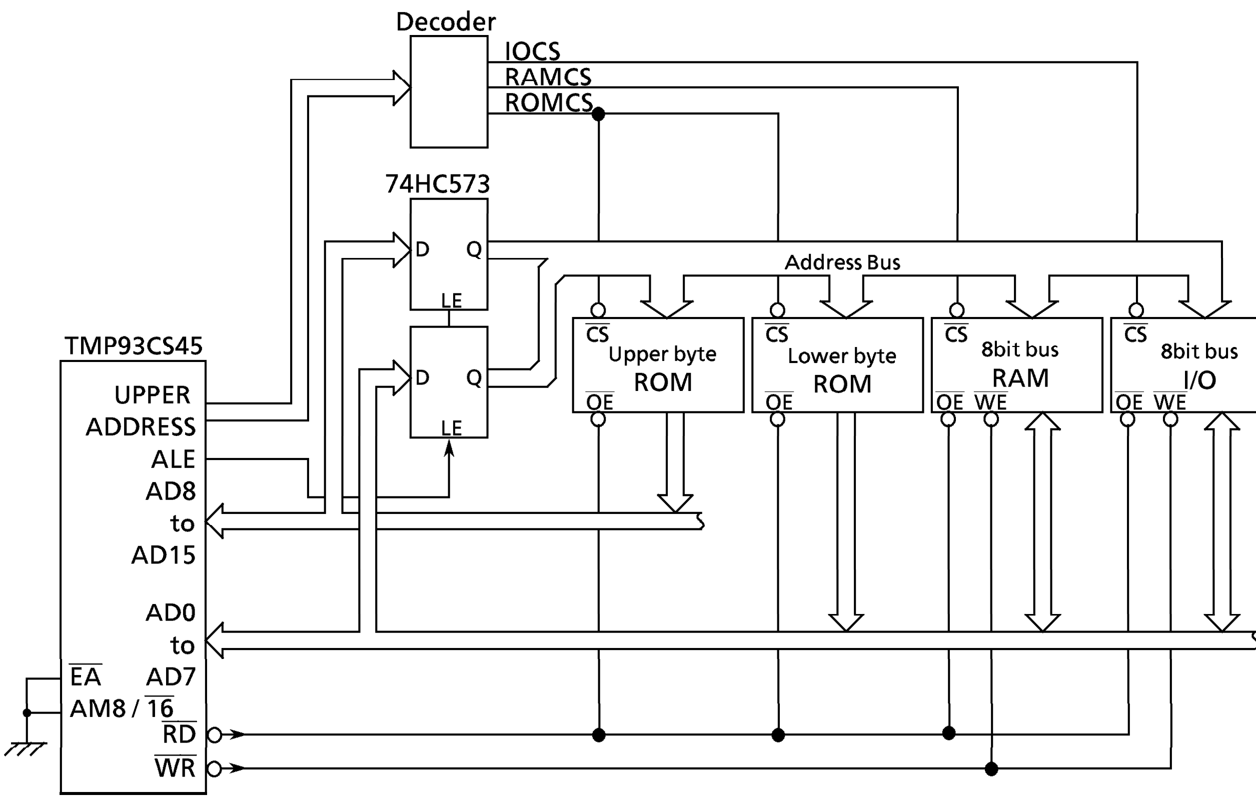


Figure 3.6 (2) Example of external Memory Connection (ROM = 16 bits, RAM & I/O = 8 bits)

```
WAITC0   EQU          68H
WAITC1   EQU          69H
WAITC2   EQU          6AH
LD       (WAITC0),    XXX10000B   ; WAITC0 = 8 bit, 2WAIT, 7F00H to 7FFFH
LD       (WAITC1),    XXX11100B   ; WAITC1 = 8 bit, 0WAIT, 880H to 7EFFH
LD       (WAITC2),    XXX00111B   ; WAITC2 = 16 bit, 1WAIT, C00000H to FFFFFFH
```

(Note) X : Don't care

(2) Example of Usage-2

Figure 3.6 (3) is an example in which an external memory is connected to the TMP93CS45. In this example, a ROM, RAM, and I/O are connected using 8 bit bus.

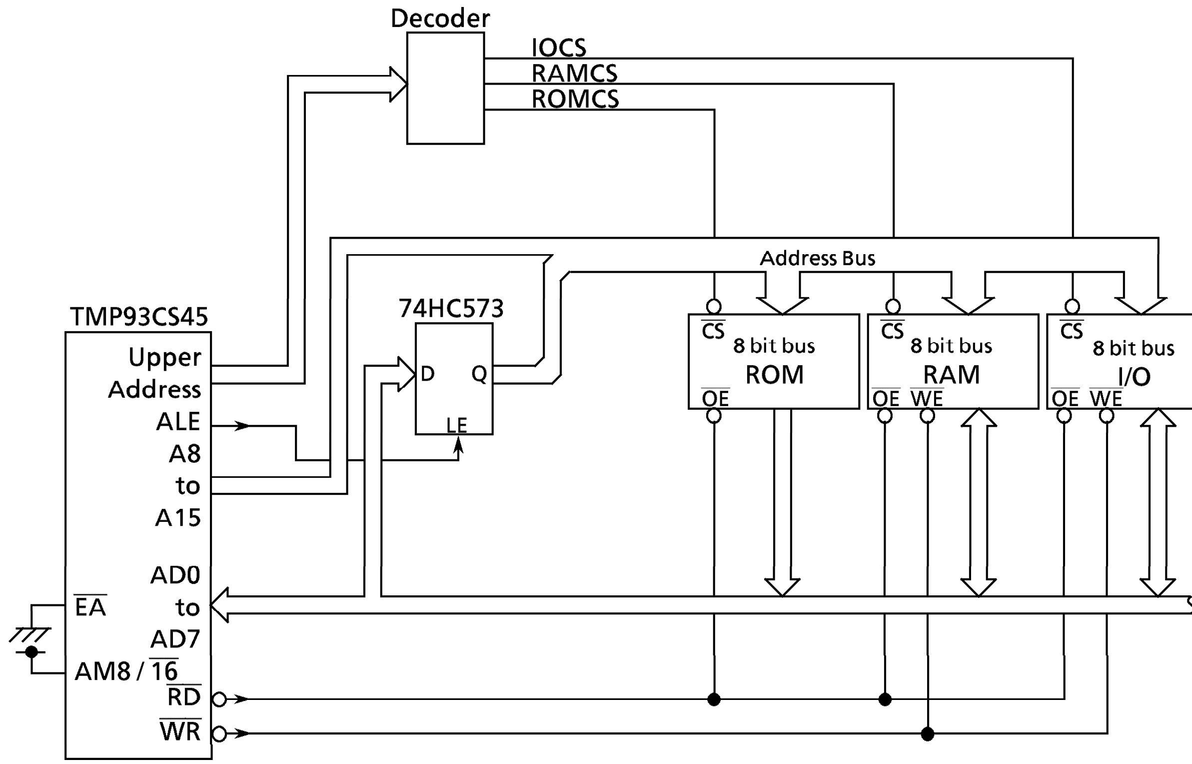


Figure 3.6 (3) Example of External Memory Connection (ROM & RAM & I/O = 8 bit)

```
WAITC0    EQU         68H
WAITC1    EQU         69H
WAITC2    EQU         6AH
LD        (WAITC0),   XXX10000B    ; WAITC0 = 8 bit, 2WAIT, 7F00H to 7FFFH
LD        (WAITC1),   XXX11100B    ; WAITC1 = 8 bit, 0WAIT, 880H to 7EFFH
LD        (WAITC2),   XXX10111B    ; WAITC2 = 8 bit, 1WAIT, C00000H to FFFFFFH
```

(Note) X : Don't care

(3) Example of Usage-3

Figure 3.6.(4) is an example in which an external memory is connected to the TMP93CS44. In this example, ROM 128K byte is connected using 16 bit bus, and RAM 256K byte using 16 bit bus.

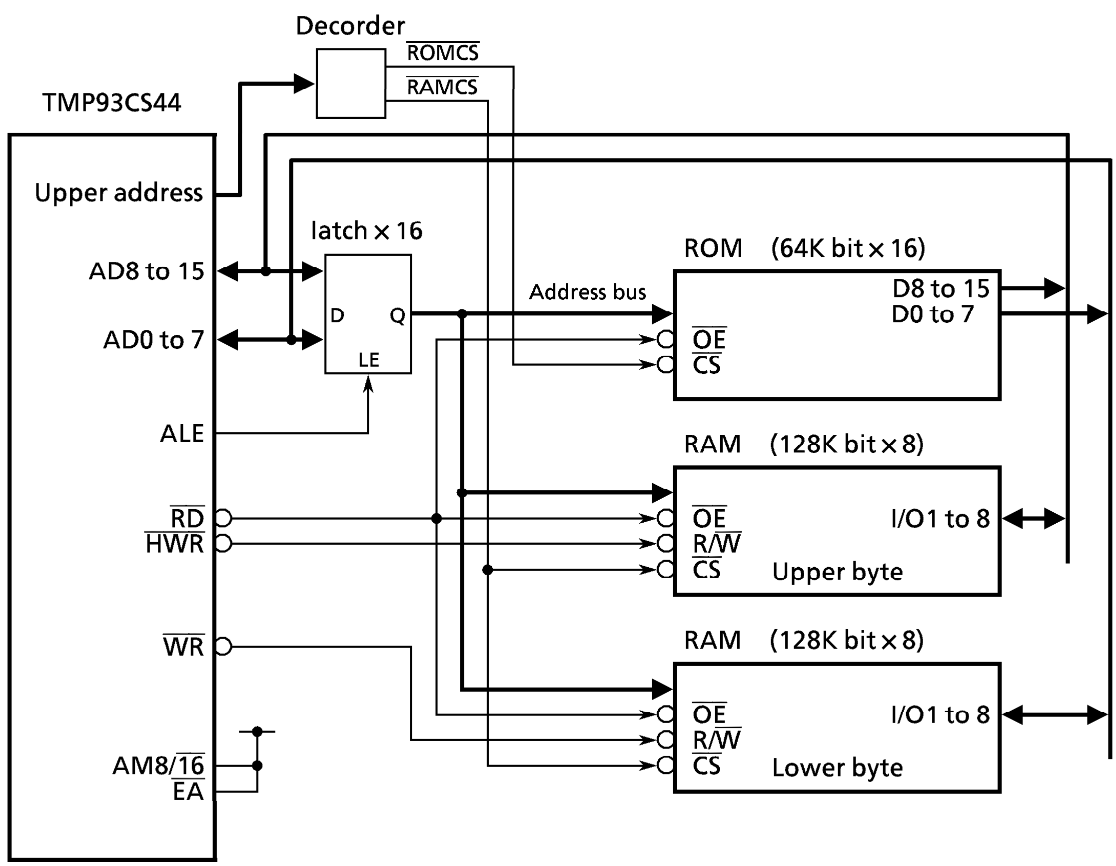


Figure 3.6 (4) Example of External Memory Connection (ROM & RAM = 16 bits)

The TMP93CS44 has built-in ROM and RAM. When ROM and RAM have insufficient capacity, it is possible to connect an external memory as the example of the external memory connection. In this example, the memory configuration is as follows.

Memory		Memory size	Address	Data bus
ROM	Internal	64K Byte	FF0000H to FFFFFFH	16 bit
	External	128K Byte	400000H to 41FFFFH	16 bit
SRAM	Internal	2K Byte	000080H to 00087FH	16 bit
	External	256K Byte	800000H to 83FFFFH	16 bit

3.7 8-bit Timers

TMP93CS44 / S45 contains four 8-bit timers (timers 0, 1, 2, 3), 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 (4 timers)
- 16-bit interval timer mode (2 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, 1), and Figure 3.7 (2) shows the block diagram of 8-bit timer (timer 2, 3).
Each interval timer consists of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Besides, timer flip-flops (TFF1, TFF3), are provided for pair of timer 0/1 and 2/3.
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 (3).
The operation modes and timer flip-flops of the 8-bit timer are controlled by five control registers T10MOD, T32MOD, TFFCR, TRUN and TRDC.

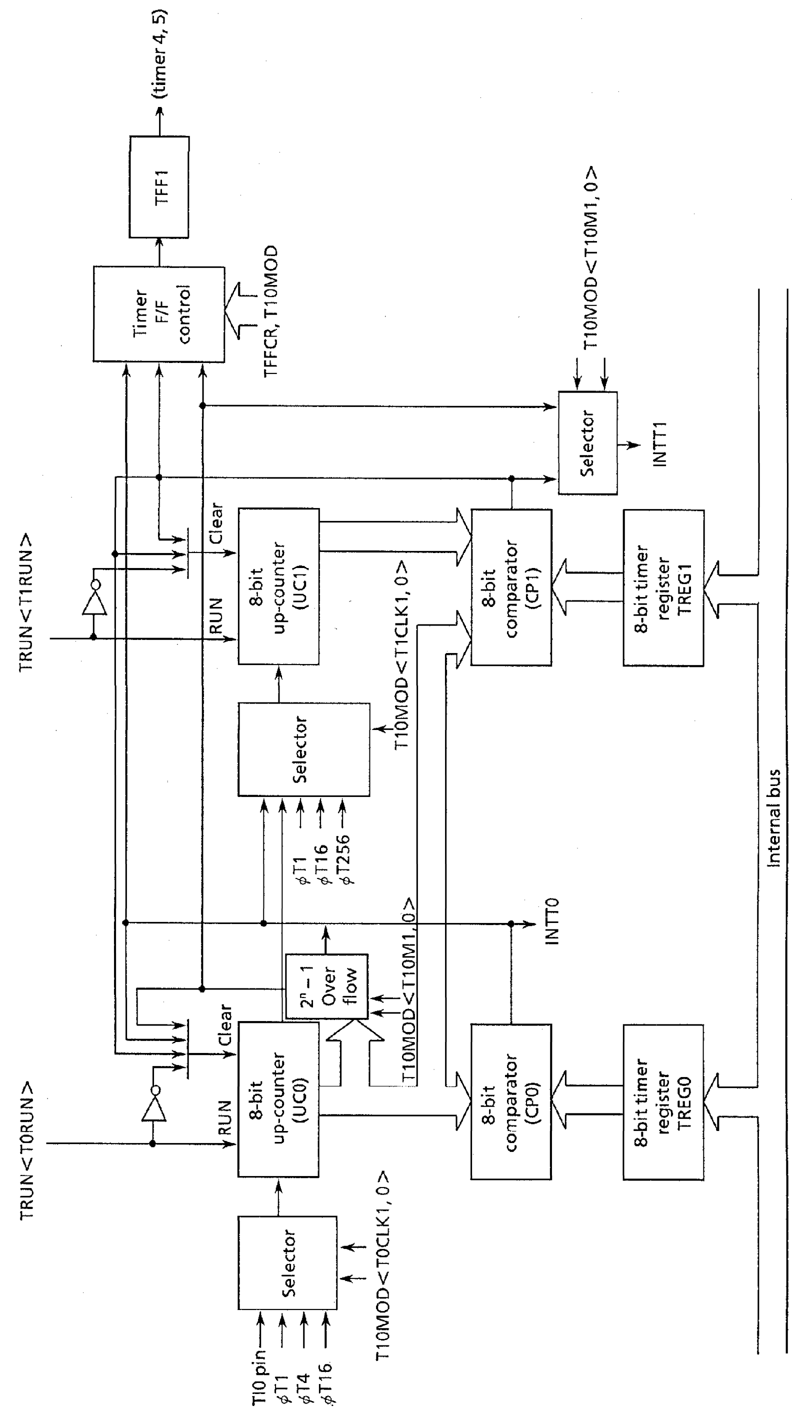


Figure 3.7 (1) Block Diagram of 8-bit Timers (Timers 0 and 1)

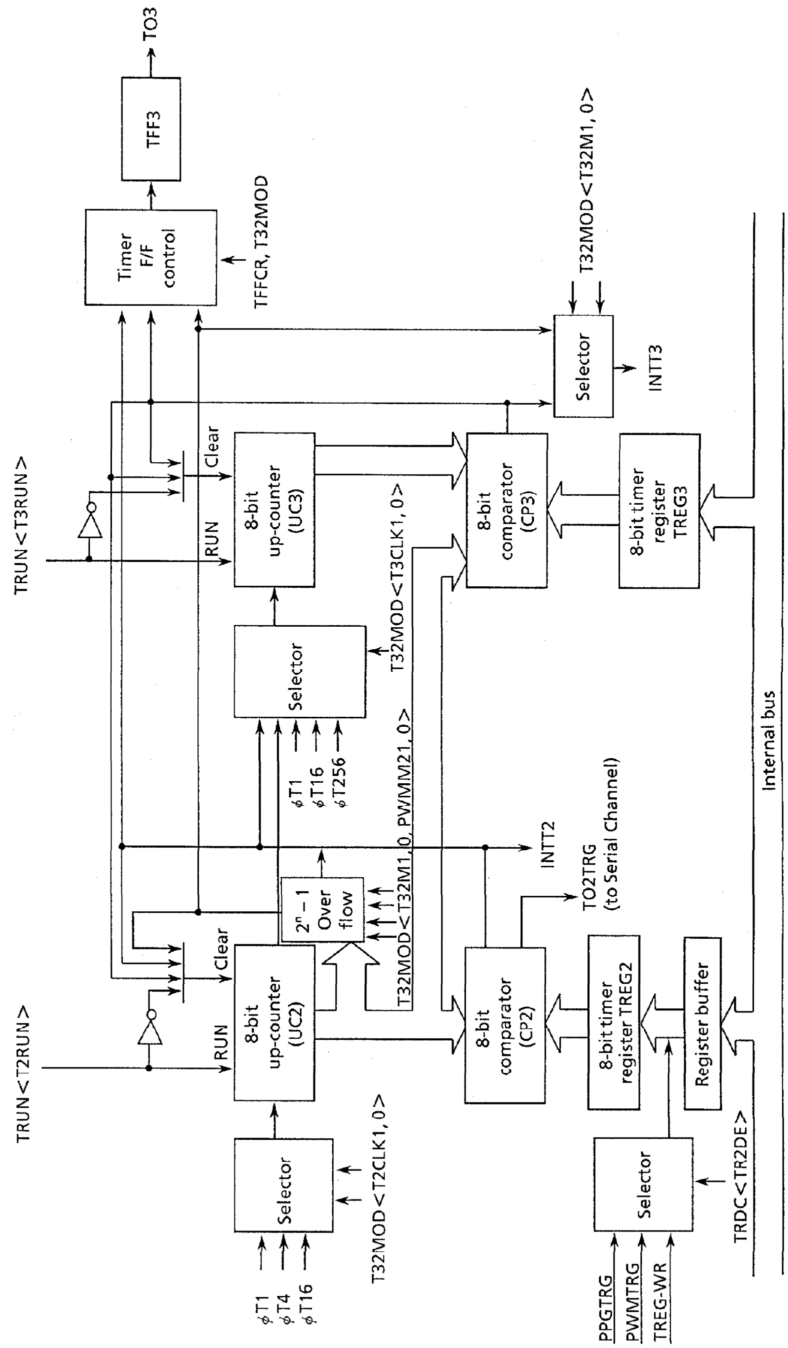


Figure 3.7 (2) Block Diagram of 8-bit Timers (Timers 2 and 3)

① Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit Timer 0, 1, 2, 3, 16-bit Timer 4, 5 and Serial Interface 0, 1.
Figure 3.7 (3) shows the block diagram. Table 3.7 (1) shows prescaler clock resolution into 8, 16-bit Timer.

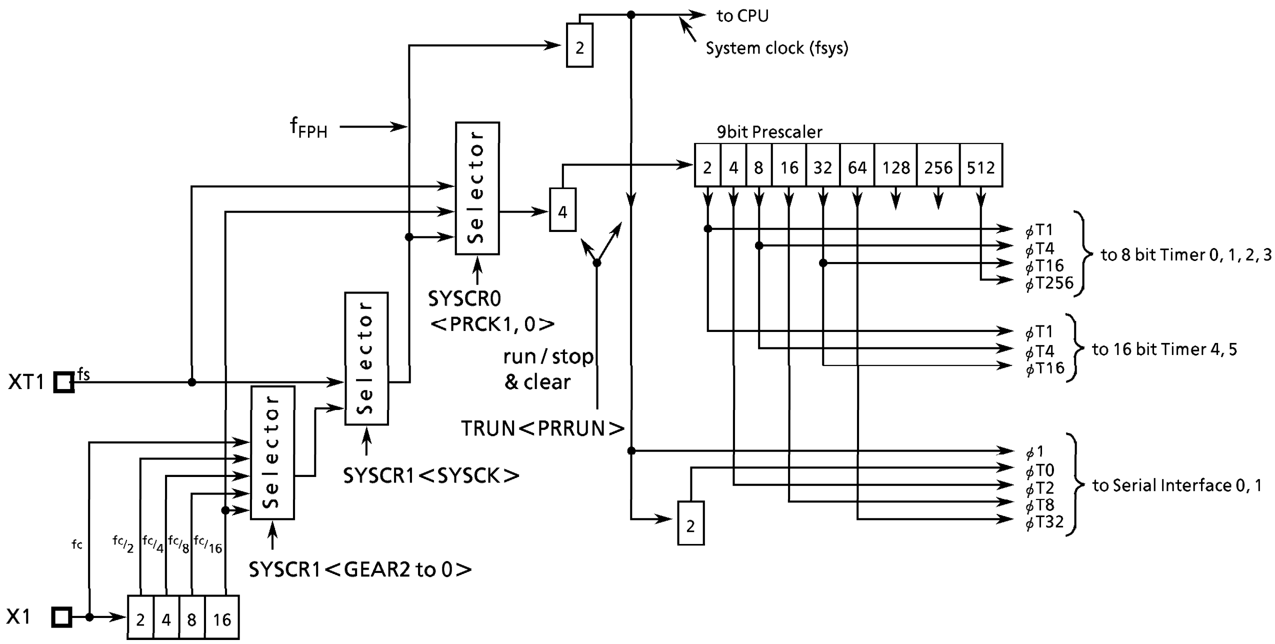


Figure 3.7 (3) The Block Diagram of Prescaler

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

at fc = 20 MHz, fs = 32.768 kHz

Select system clock <SYSCK>	Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 μs)	$fs/2^7$ (4 ms)	$fs/2^{11}$ (62.5 ms)
0 (fc)		000 (fc)	$fc/2^3$ (0.4 μs)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^{11}$ (102.4 μs)
		001 (fc/2)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{12}$ (204.8 μs)
		010 (fc/4)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{13}$ (409.6 μs)
		011 (fc/8)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)	$fc/2^{14}$ (819.2 μs)
		100 (fc/16)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.6384 ms)
XXX	01 (low frequency clock)	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 μs)	$fs/2^7$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
XXX	10 (note) (fc/16 clock)	XXX	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.8 μs)	$fc/2^{15}$ (1.6384 ms)

XXX : don't care

← 16 bit Timer → (φT1, φT4, φT16)

← 8 bit Timer → (φT1, φT4, φT16, φT256)

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

The clock selected among f_{FPH} clock, fc / 16 clock, and fs clock is divided by 4 and 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 selects between 4 clock inputs : ϕT1, ϕT4, ϕT16, and ϕT256 among the prescaler output.
This 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 T10MOD and T32MOD.
The input clock of timer 0, 2 is selected from the external clock from TI0 (only timer 0) pin and the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of T10MOD / T32MOD registers.
The input clock of timer 1 and 3 differs depending on the operation mode. When set to 16-bit timer mode, the overflow outputs of timer 0 and 2 are used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (match detection signal) of timer 0, 2 according to the set value of T10MOD and T32MOD registers.

Example : When T10MOD<T10M1,0> = 01, the overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer mode).
When T10MOD<T10M1,0> = 00 and T10MOD<T1CLK1,0> = 01, ϕT1 becomes the input of timer 1 (8 bit timer mode).

Operation mode is also set by T10MOD and T32MOD registers. When reset, it is initialized to T10MOD<T10M1, 0> = 00 and T32MOD<T32M1, 0> = 00 whereby the up-counter is placed in the 8-bit timer mode.
The counting and stop & 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, TREG2, TREG3, 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 registers TREG2 are double buffer structure, each of which makes a pair with register buffer.
The timer flip-flop controll register TRDC<TR2DE> bits control whether the double buffer structure in the TREG2 should be enabled or disabled. They are disabled when <TR2DE> = 0 and enabled when they are set to 1.
In the condition of double buffer enable state, the data is transfered from the register buffer to the timer register when the 2^n-1 overflow occurs in PWM mode, or at the PPG cycle in PPG mode. Therefore, during timer mode, the double buffer can not be used.
When reset, it will be initialized to <TR2DE> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <TR2DE> to 1, and write the following data in the register buffer.

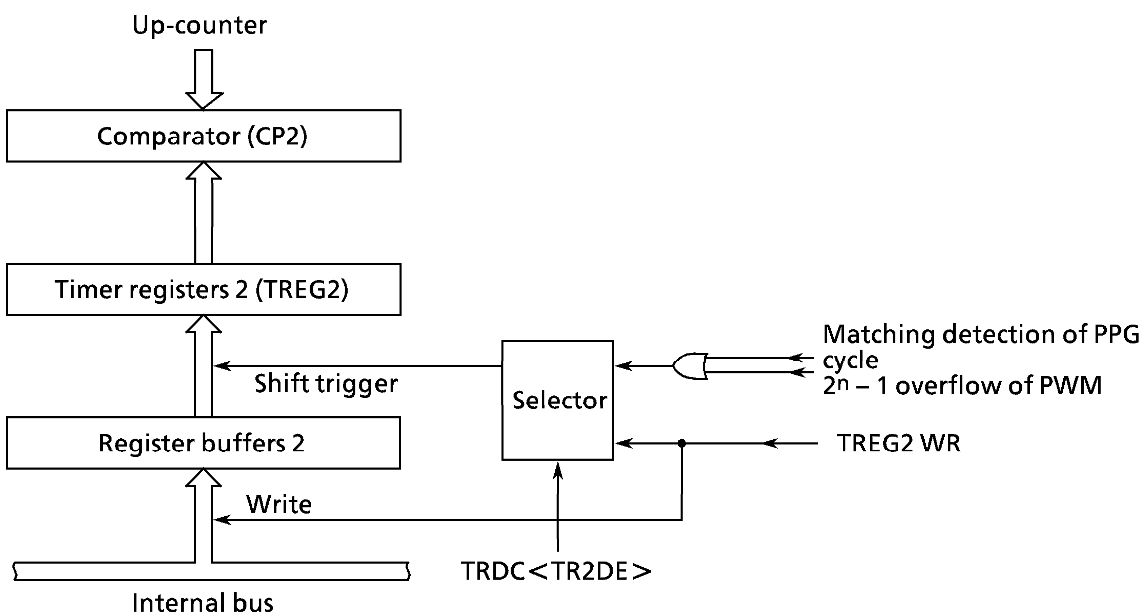


Figure 3.7 (4) Configuration of Timer Register 2

Note : Timer register and the register buffer are allocated to the same memory address. When <TR2DE> = 0, the same value is written in the register buffer as well as the timer register, while when <TR2DE> = 1 only the register buffer is written.

The memory address of each timer register is as follows.

TREG0: 000022H TREG2: 000026H

TREG1: 000023H TREG3: 000027H

All the 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, INTT2, INTT3) 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, TFF3)

The timer flip-flop (TFF1, TFF3) is a flip-flop inverted by the match detect signal (8-bit comparator output) of each interval timer.
Inverting is disabled or enabled by the timer flip-flop control register TFFCR<TFF3IE, TFF1IE>.
After reset operation, the value of TFF1, TFF3 is undefined. Writing "01" or "10" to TFFCR<TFF3C1-0, TFF1C1-0>sets "0" or "1" to TFF1, TFF3. Additionally, writing "00" to this bit inverts the value of TFF1, TFF3. (software inversion)
The signal of TFF3 is output through the TO3 pin (also used as P41). When using as the timer output, the timer flip-flop should be set by port 4 function register P4FC beforehand. The output pin of TFF1 does not exist.

TRUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	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)
T3RUN : Operation of 8-bit timer (timer3)
T2RUN : Operation of 8-bit timer (timer2)
T1RUN : Operation of 8-bit timer (timer1)
T0RUN : Operation of 8-bit timer (timer0)

Note : TRUN <bit 6> is always read as "1".

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

select gear value of high frequency

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

00–10: 1 / 4times clock

Figure 3.7 (5) Timer Operation Control Register / System Clock Control Register

T10MOD (0024H)

	7	6	5	4	3	2	1	0
bit Symbol	T10M1	T10M0			T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W				R/W			
After reset	0	0			0	0	0	0
Function	Operation mode 00 : 8 bit Timer 01 : 16 bit Timer 10 : – 11 : –				Source clock of timer1 00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		Source clock of timer0 00 : TI0 01 : ϕT1 10 : ϕT4 11 : ϕT16	

Input clock of timer 0

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

Input clock of timer 1

	T10MOD<T10M1,0> ≠ 01	T10MOD <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	

Set the operation mode of timer 0 and 1.

00	Two 8-bit timers (timer 0 and timer 1)
01	16-bit timer
10	–
11	–

Figure 3.7 (6) Timer Mode control Register (T10MOD)

T32MOD
(0028H)

	7	6	5	4	3	2	1	0
bit Symbol	T32M1	T32M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		PWM cycle 00: – 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock of timer3 00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of timer2 00: – 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 2

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

Input clock of timer 3

	T32MOD <T32M1,0>≠01	T32MOD <T32M1,0>=01
00	Comparator output of timer 2	Overflow output of timer 2
01	Internal clock ϕT1	
10	Internal clock ϕT16	(16-bit timer mode)
11	Internal clock ϕT256	

Select PWM cycle

00	—
01	(2^6-1) × Input clock frequency of timer 2
10	(2^7-1) × Input clock frequency of timer 2
11	(2^8-1) × Input clock frequency of timer 2

Set the operation mode of timer 2 and 3.

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 2) + 8-bit timer (timer 3)

Figure 3.7 (7) Timer Mode control Register (T32MOD)

TFFCR (0025H)

	7	6	5	4	3	2	1	0
bit Symbol	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read/Write	W		R/W		W		R/W	
After reset	1	1	0	0	1	1	0	0
Function	00 : Invert TFF3 01 : Set TFF3 10 : Clear TFF3 11 : don't care		TFF3 Inversion trigger 0: Disable 1: Enable	TFF3 Inversion source 0: Timer 2 1: Timer 3	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : don't care		TFF1 Inversion trigger 0: Disable 1: Enable	TFF1 Inversion source 0: Timer 0 1: Timer 1

Select inverse signal of timer F/F3
("Don't care" except in 8-bit timer mode)

0	Inversion by timer 2
1	Inversion by timer 3

Inversion of Timer F/F3 (TFF3)

0	Disable invert
1	Enable invert

Control of Timer F/F3 (TFF3)

00	Invert the value of TFF3 (software inversion)
01	Set TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care

Select inverse signal of timer F/F1
("Don't care" except in 8-bit timer mode)

0	Inversion by timer 0
1	Inversion by timer 1

Inversion of Timer F/F1 (TFF1)

0	Disable invert
1	Enable invert

Control of Timer F/F1 (TFF1)

00	Invert the value of TFF1 (software inversion)
01	Set TFF1 to "1".
10	Clear TFF1 to "0".
11	Don't care

Note : TFFCR<TFF3C1 to 0, TFF1C1 to 0> is always read as "1".

Figure 3.7 (8) Timer Flip-flop Control Register (TFFCR)

TRDC (0029H)

	7	6	5	4	3	2	1	0
bit Symbol							TR2DE	–
Read/Write							R/W	
After reset							0	0
Function							0: Duble Buffer Disable 1: Duble Buffer Enable	(Note) Always write "0"

Operation of Timer Register 2 Double butter

00	Disable
01	Enable

Figure 3.7 (9) Timer Register Double Buffer Control Register (TRDC)

(1) 8-bit timer mode

Four interval timers 0, 1, 2, 3, can be used independently as 8-bit interval timer.

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

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 T10MOD and TREG1 register, respectively. Then, enable interrupt INTT1 and start the counting of timer 1.

Example : To generate timer 1 interrupt every 1 seconds at fs = 32 kHz, set each register in the following manner.

※ Clock Condition
- system clock : low frequency (fs)
- 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".
T10MOD   ← 0 0 X X 1 0 - -     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 : X: don't care –; no change

Use the table 3.7 (1) for selecting the input clock.

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 is included in timer 1 and 3.
The timer flip-flop (TFF3) is inverted at constant intervals, and its status is output to timer output pin (TO3). The output pin of TFF1 does not exist.

Example : To output a 2.4 μs square wave pulse from TO3 pin at fc = 20 MHz, set each register in the following procedures. Either timer 2 or timer 3 may be used, but this example uses timer 3.

※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	f_{FPH}

	7 6 5 4 3 2 1 0	
TRUN ←	– X – – 0 – – –	Stop timer 3, and clear it to "0".
T32MOD ←	0 0 X X 0 1 – –	Set the 8-bit timer mode, and select φT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG1 ←	0 0 0 0 0 0 1 1	Set the timer register at 2.4 μs ÷ φT1 ÷ 2 = 3.
TFFCR ←	1 0 1 1 – – – –	Set TFF3 to "0", and set to invert by the match detect signal from timer 3.
P4CR ←	– – – – – – 1 –	Select P41 as TO3 pin.
P4FC ←	– X X – X X 1 X	
TRUN ←	1 X – – 1 – – –	Start timer 3 counting.

Note : X ; don't care – ; no change

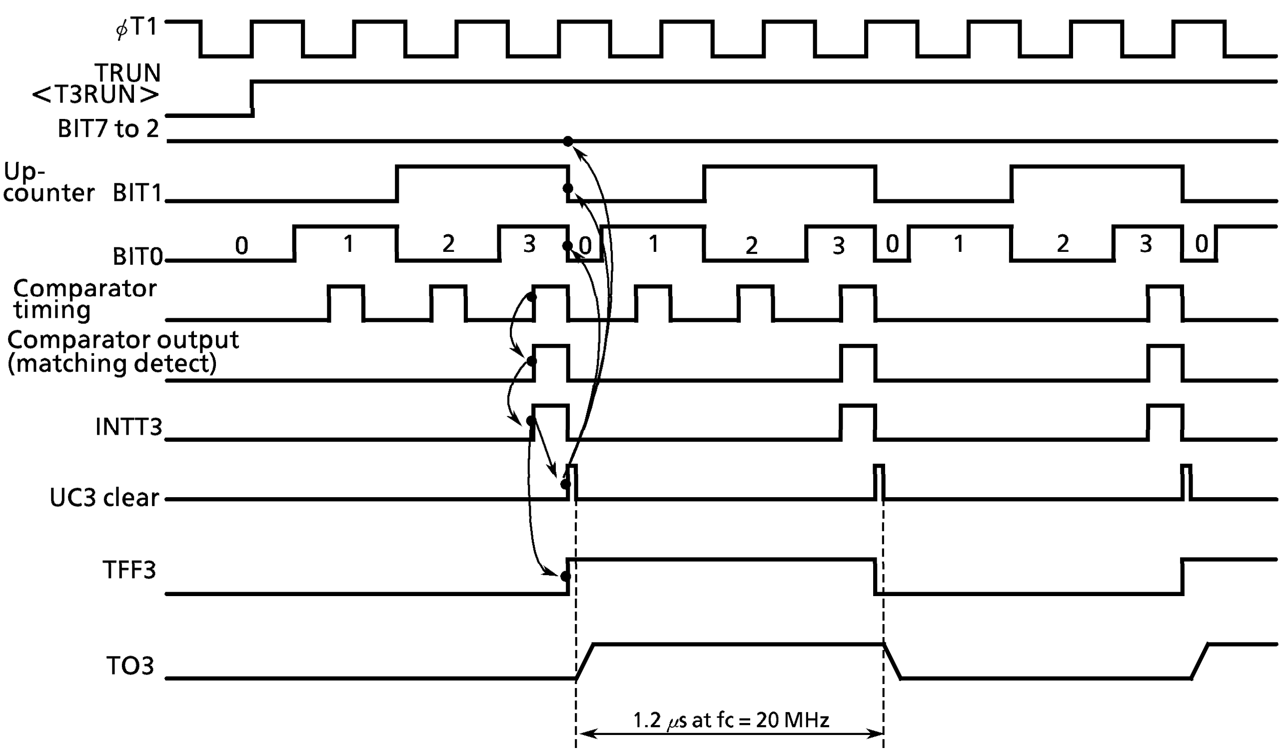


Figure 3.7 (10) Square Wave (50% Duty) Output Timing Chart

③ Making timer 1 count up by match signal from timer 0 comparator
(Same function is achieved by using timer 3 and timer 2)

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

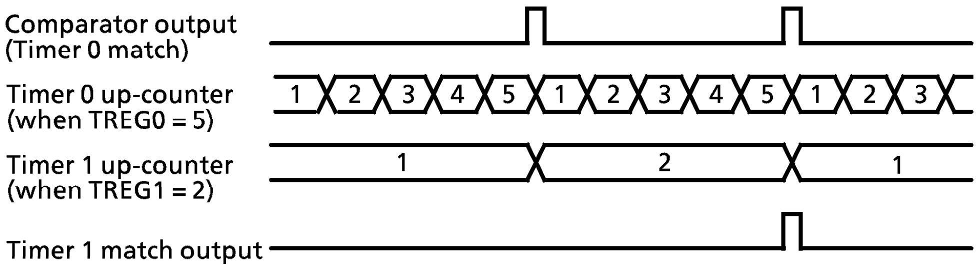


Figure 3.7 (11) Timer 1 count up by timer 0

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1 or timer 2 and timer 3.
To make a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 0/1 mode register T10MOD<T10M1,0> to "01".
When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1 and 3, regardless of the set value of T10MOD<T1CLK1,0>and T32MOD<T3CLK1, 0>. Table 3.7 (1) shows the relation between the cycle of timer (interrupt) and the selection of input clock.

The lower 8 bits of the timer (interrupt) cycle are set by the timer register TREG0 or TREG2, and the upper 8 bits are set by TREG1 or TREG3. Note that TREG0 and TREG2 always must be set first. (Writing data into TREG0 and TREG2 disables the comparator temporarily, and the comparator is restarted by writing data into TREG1 and TREG3.)

Setting example : To generate an interrupt INTT3 every 0.4 seconds at fc = 20 MHz, set the following values for timer registers TREG2 and TREG3.

※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	f_{FPH}

When counting with input clock of ϕT16 (6.4 μs at 20 MHz)

0.4 s ÷ 6.4 μs = 62500 = F424H

Therefore, set TREG3 = F4H and TREG2 = 24H, respectively.

The comparator match signal is output from timer 2 each time the up-counter UC2 matches TREG2, where the up-counter UC2 is not be cleared.
With the timer 3 comparator, the match detect signal is output at each comparator timing when up-counter UC3 and TREG3 values match. When the match detect signal is output simultaneously from both comparators of timer 2 and timer 3, the up-counters UC2 and UC3 are cleared to “0”, and the interrupt INTT3 is generated. If inversion is enabled, the value of the timer flip-flop TFF3 is inverted.

Example : When TREG3 = 04H and TREG2 = 80H

Figure 3.7 (12) Timer output by 16-bit timer mode

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

Square wave pulse can be generated at any frequency and duty by timer 2. The output pulse may be either low-active or high-active. In this mode, timer 3 cannot be used.
Timer 2 outputs pulse to TO3 pin (also used as P41).
In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up-counter (UC2) matches the timer registers TREG2 and TREG3.
However, it is required that the set value of TREG2 is smaller than that of TREG3.
Though the up-counter (UC3) of timer 3 is not used in this mode, UC3 should be set for counting by setting TRUN<T3RUN> to 1.
Figure 3.7 (14) shows the block diagram for this mode.

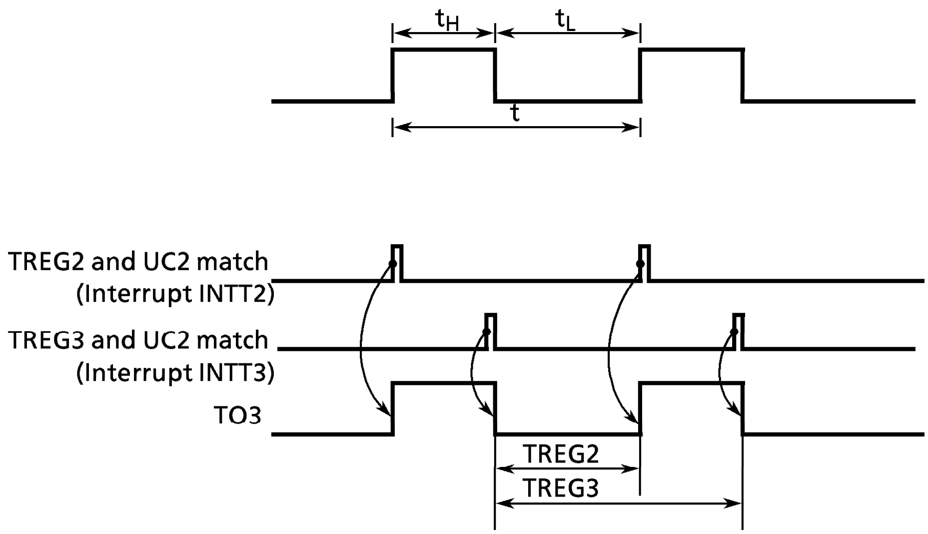


Figure 3.7 (13) 8 bit PPG output waveforms

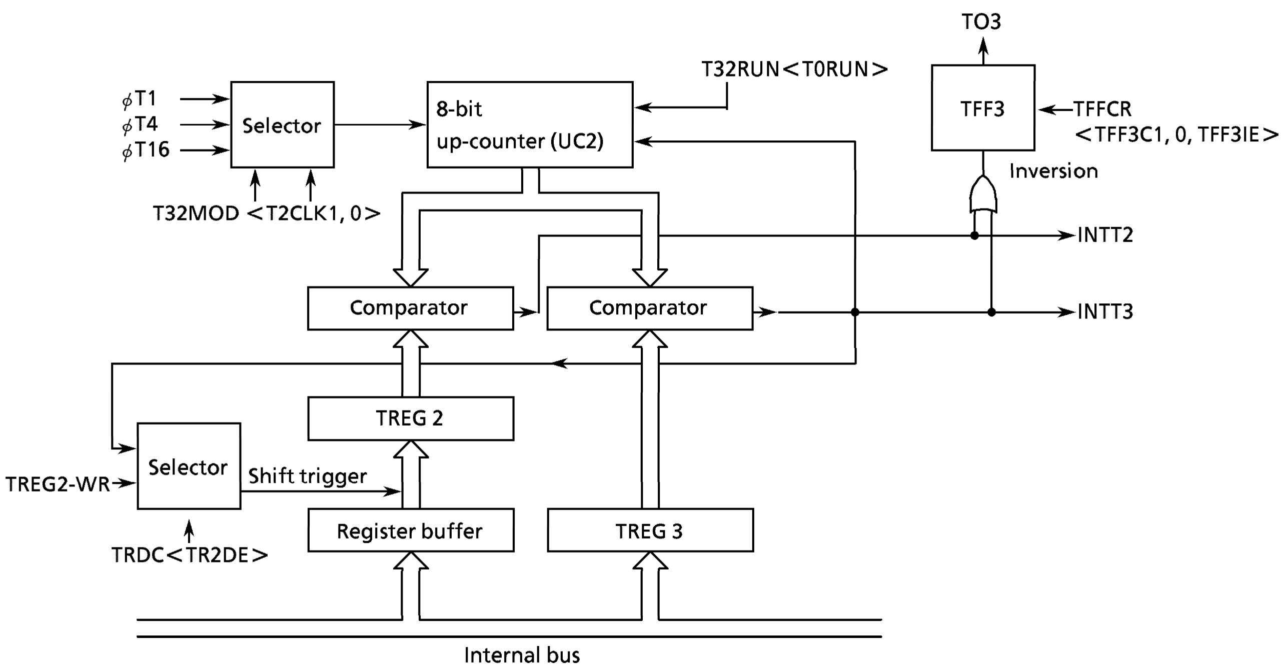


Figure 3.7 (14) Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TREG2 is enabled in this mode, the value of register buffer will be shifted in TREG2 each time TREG3 matches UC2.
Use of the double buffer makes easy the handling of low duty waves (when duty is varied).

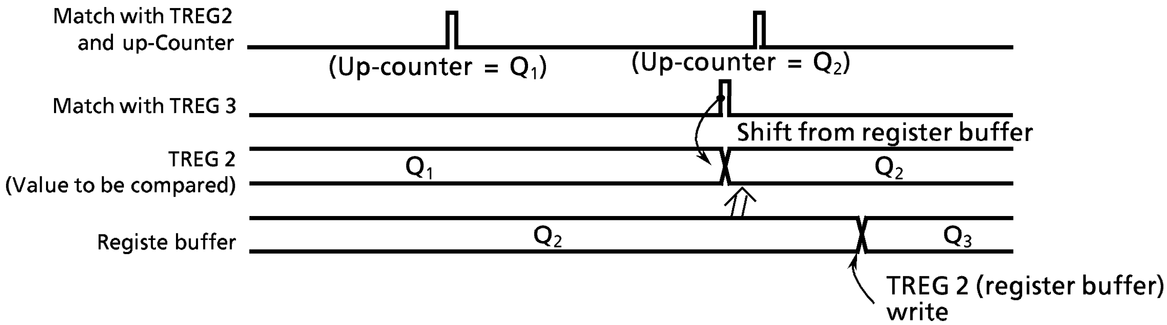
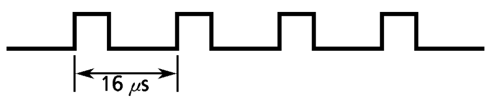


Figure 3.7 (15) Operation of Register buffer

Example : Generating 1 / 4 duty 62.5 kHz pulse (at fc = 20 MHz)



※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	f_{FPH}

- Calculate the value to be set for timer register.
  To obtain the frequency 62.5 kHz, the pulse cycle t should be : t = 1/62.5 kHz = 16 μs.
  Given ϕT1 = 0.4 μs (at 20 MHz),
  16 μs ÷ 0.4 μs = 40
  Consequently, to set the timer register 3 (TREG3) to TREG3 = 40 = 28H
  and then duty to 1/4, t × 1/4 = 16 μs × 1/4 = 4 μs
  4 μs ÷ 0.4 μs = 10
  Therefore, set timer register 2 (TREG2) to TREG2 = 10 = 0AH.

```
          7 6 5 4 3 2 1 0
TRUN   ←  - X - - 0 0 - -     Stop timer 2, 3 and clear it to "0".
T32MOD ←  1 0 X X X X 0 1     Set the 8-bit PPG mode, and select φT1 as input clock.
TREG2  ←  0 0 0 0 1 0 1 0     Write "0AH".
TREG3  ←  0 0 1 0 1 0 0 0     Write "28H".
TFFCR  ←  0 1 1 X - - - -     Sets TFF3 and enable the inversion and double buffer
                              enable.
          └┬┘
           └─────────────────→Writing "10" provides negative logic pulse.
P4CR   ←  - - - - - - 1 -  ┐
P4FC   ←  - X X - X X 1 X  ┤  Set P41 as the TO3 pin.
TRUN   ←  1 X - - 1 1 - -  ┘  Start timer 2 and timer 3 counting.
```

Note : X ; Don't care   – ; no change

(4) 8-bit PWM Output mode

This mode is valid only for timer 2. In this mode, maximum 8-bit resolution of PWM pulse can be output.
PWM pulse is output to TO3 pin (also used as P41) when using timer 2. Timer 3 can also be used as 8-bit timer.
Timer output is inverted when up-counter (UC2) matches the set value of timer register TREG2 or when 2^n-1 (n=6, 7, or 8; specified by T32MOD<PWM21 to 20>) counter overflow occurs. Up-counter UC0 is cleared when 2^n-1 counter overflow occurs.
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) $\neq$ 0

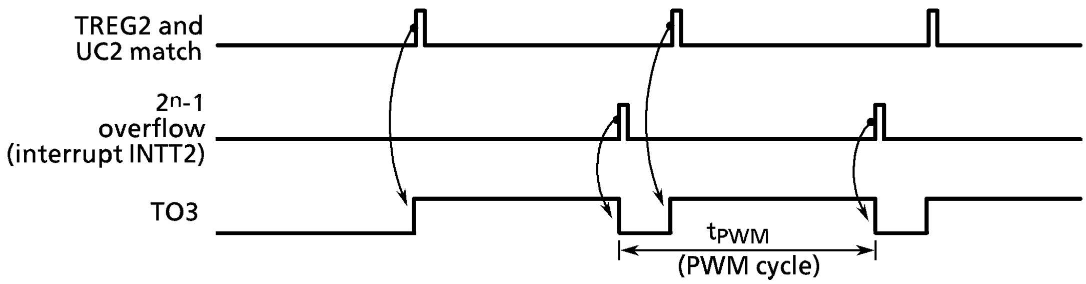


Figure 3.7 (16) 8-bit PWM waveforms

Figure 3.7 (17) shows the block diagram of this mode.

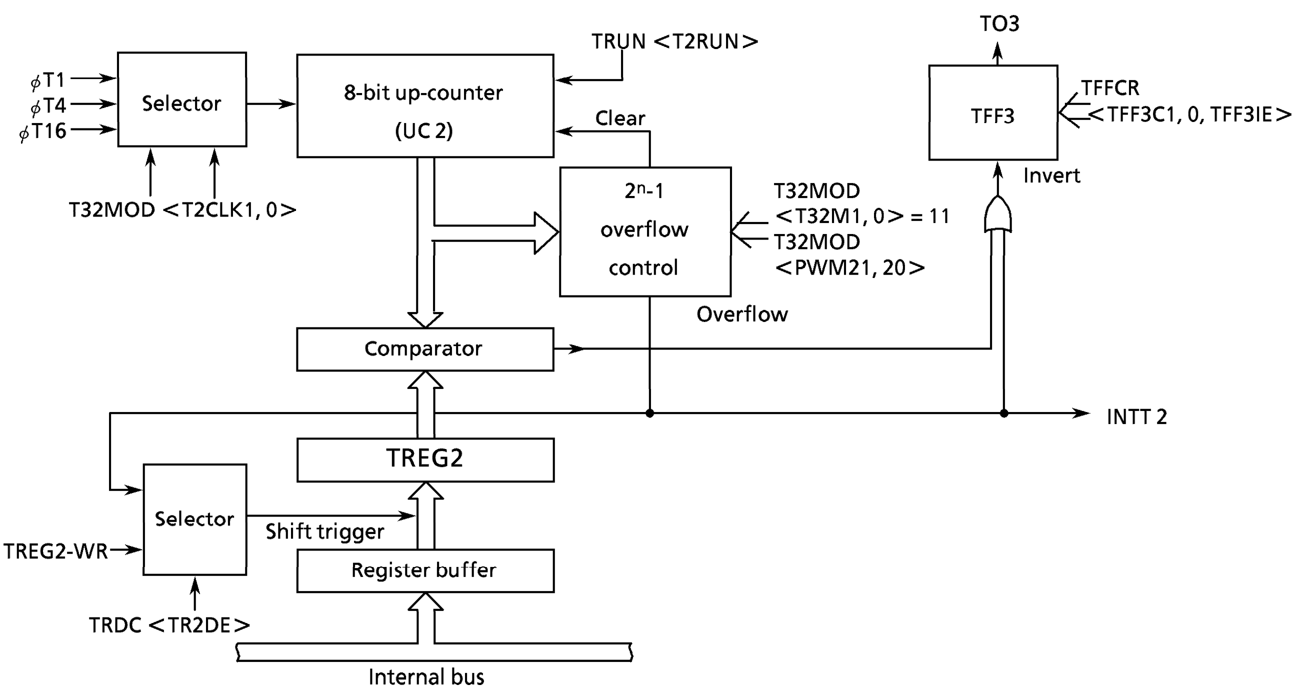


Figure 3.7 (17) Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TREG2 if 2^n-1 overflow is detected when the double buffer of TREG2 is enabled.
Use of the double buffer makes easy the handling of small duty waves.

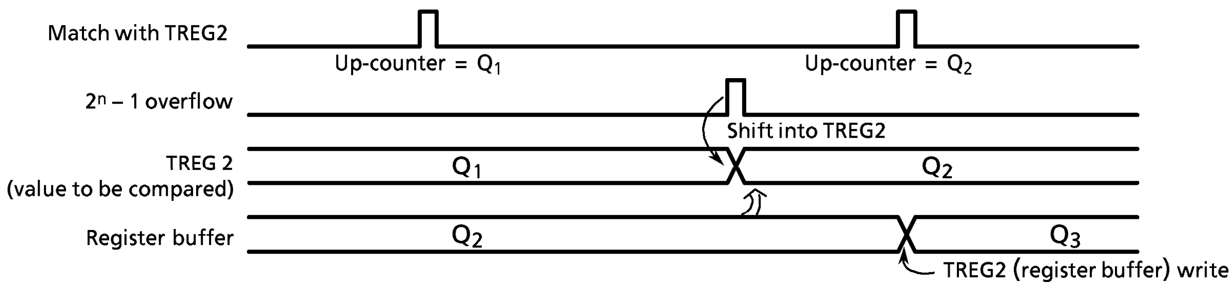
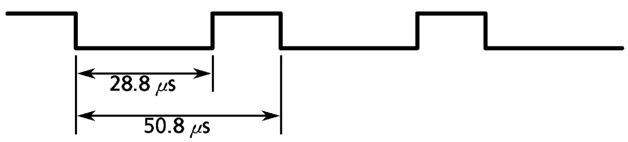


Figure 3.7 (18) Operation of Register buffer

Example : To output the following PWM waves to TO3 pin at fc = 20 MHz.



※ Clock Condition

system clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	f_{FPH}

To realize 50.8 μs of PWM cycle by $\phi T1 = 0.4$ μs (at fc = 20 MHz),

$$50.8\ \mu s \div 0.4\ \mu s = 127 = 2^n - 1$$

Consequently, n should be set to 7.
As the period of low level is 28.8 μs, for $\phi T1 = 0.4$ μs,
set the following value for TREG2.

$$28.8\ \mu s \div 0.4\ \mu s = 72 = 48H$$

```
           MSB             LSB
            7 6 5 4 3 2 1 0
TRUN   ←  - X - - - 0 - -      Stop timer 2, and clear it to "0".
T32MOD ←  1 1 1 0 - - 0 1      Set 8-bit PWM mode (cycle: 2^7 - 1) and select φT1 as the input
                               clock.
TREG2  ←  0 1 0 0 1 0 0 0      Writes "48H".
TFFCR  ←  1 0 1 X - - - -      Clears TFF3, enable the inversion and double buffer.

P4CR   ←  - - - - - - 1 -   }
P4FC   ←  - X X - X X 1 X   }  Set P41 as the TO3 pin.
TRUN   ←  1 X - - - 1 - -      Start timer 2 counting.
```

Note : X ; Don't care − ; no change

Table 3.7 (2) PWM Cycle

at fc = 20 MHz, fs = 32.768 kHz

select prescaler clock <PRCK1, 0>	select system clock <SYSCK>	Gear value <GEAR2 to 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	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms
	0 (fc)	000 (fc)	25.2 μs	100.8 μs	403.2 μs	50.8 μs	203.2 μs	812.8 μs	102.0 μs	408.0 μs	1.63 ms
		001 (fc/2)	50.4 μs	201.6 μs	806.4 μs	101.6 μs	406.4 μs	1.63 ms	204.0 μs	816.0 μs	3.26 ms
		010 (fc/4)	100.8 μs	403.2 μs	1.61 ms	203.2 μs	812.8 μs	3.26 ms	408.0 μs	1.63 ms	6.53 ms
		011 (fc/8)	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.52 ms	816.0 μs	3.26 ms	13.06 ms
		100 (fc/16)	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11ms
01 (low frequency clock)	XXX	XXX	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms
10 (fc/16 clock)	XXX	XXX	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11ms

XXX : don't care

(5) Timer Mode Setting Registers

Table 3.7 (3) shows the list of 8-bit timer modes.

Table 3.7 (3) Timer Mode Setting Registers

Register name	T10MOD / T32MOD				TFFCR
Name of function in register	T10M / T32M	PWM2	T1CLK / T3LK	T0CLK / T2CLK	TFF1IS / TFF3IS
Function	Timer mode	PWM cycle	Upper timer input clock	Lower timer input clock	Timer F/F invert signal select
16-bit timer mode	01	* –	–	External clock (only Timer 0), 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 (only Timer 0), ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG × 1channel	* 10	* –	* –	* External clock (only Timer 0), 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 (only Timer 0), ϕT1,ϕT4,ϕT16 (00, 01, 10, 11)	* –
8-bit timer × 1channel	11	–	ϕT1, ϕT16 , ϕT256 (01, 10, 11)	–	Output disabled

Note : - ; Don't care
*; Don't set in T10MOD

3.8 16-bit Timers / Event Counters

The TMP93CS44 / TMP93CS45 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
  Can be used following operation modes by capture function.
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

Timer / event counter consists of 16-bit up-counter, two 16-bit timer registers, 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 4 control registers: T4MOD/T5MOD, T4FFCR / T5FFCR, TRUN and T45CR.

Figure 3.8 (1), (2) shows 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, and thus only the operation of Timer 4 will be explained below.

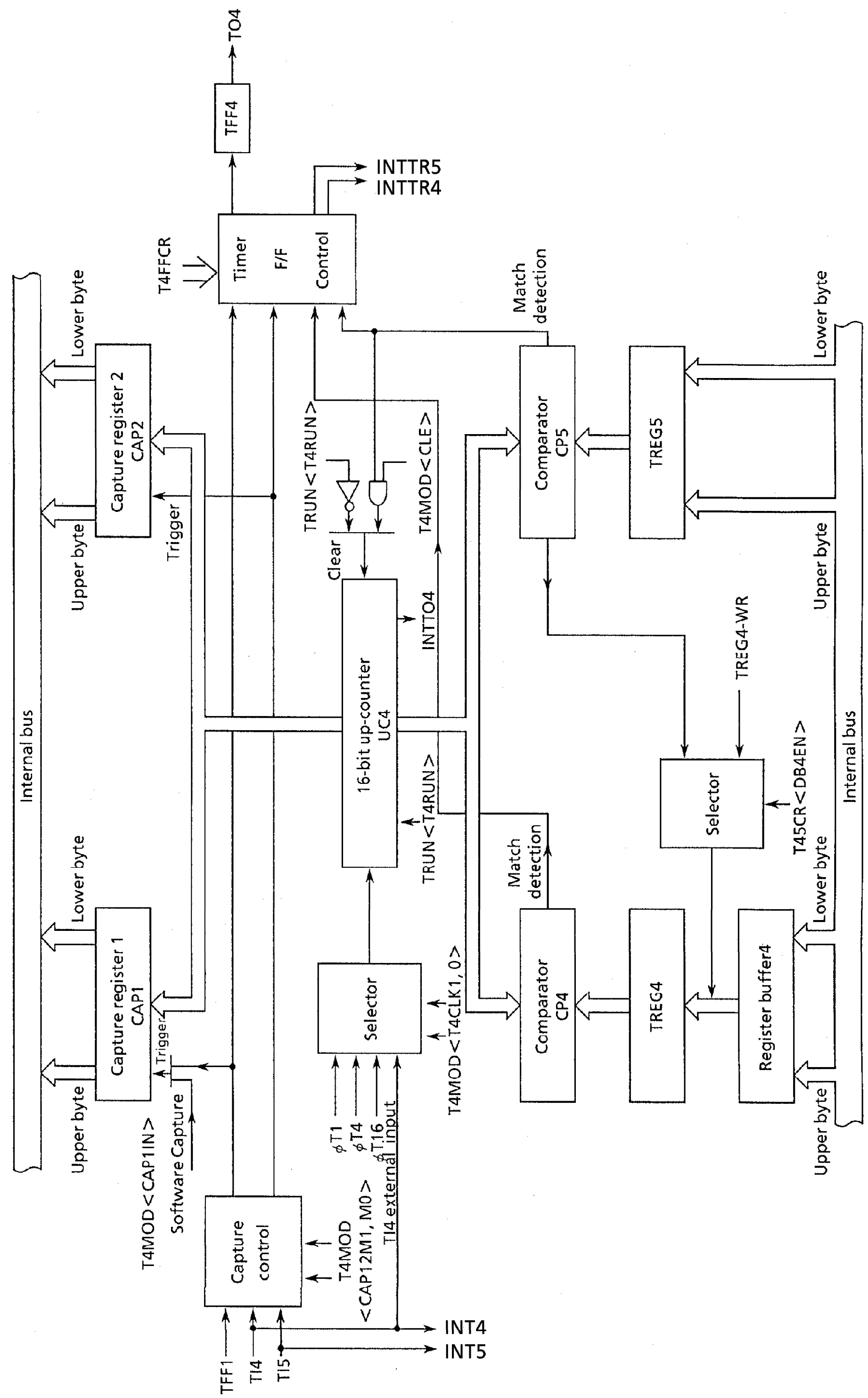


Figure 3.8 (1) Block Diagram of 16-Bit Timer (Timer 4)

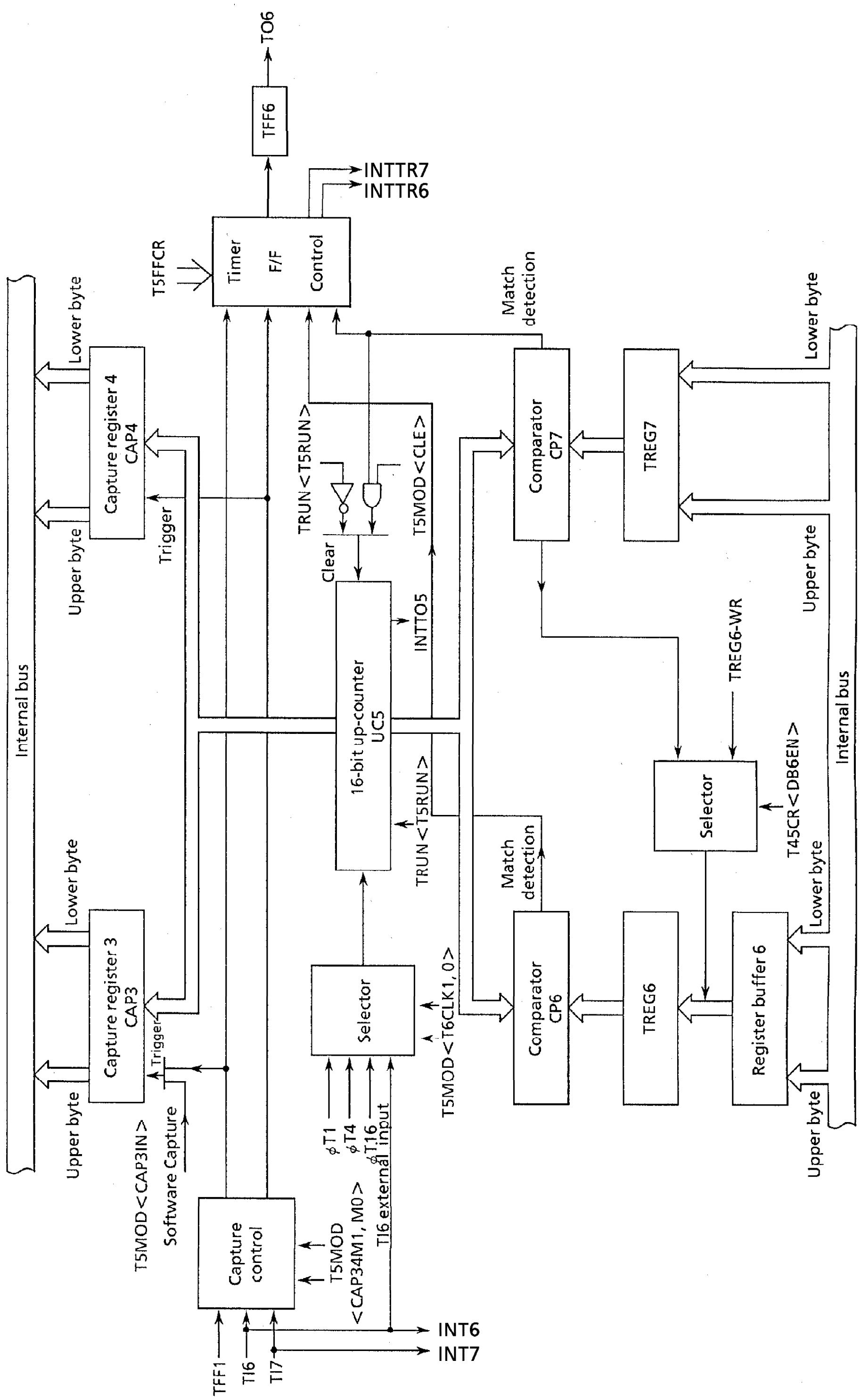


Figure 3.8 (2) Block Diagram of 16-Bit Timer (Timer 5)

T4MOD (0038H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W				
After reset			1	0	0	0	0	0
Function			0: Soft-Capture 1: don't care	Capture timing 00: Disable INT4 occurs at rising edge. 01: TI4↑ TI5↑ INT4 occurs at rising edge. 10: TI4↑ TI4↓ INT4 occurs at falling edge. 11: TFF1↑ TFF1↓ INT4 occurs at rising edge.		1: UC4 Clear Enable	Timer 4 source clock 00: TI4 pin 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 input clock

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

Clearing the up-counter UC4

00	Clear disable
01	Clear by match with TREG5.

Capture timing of timer4

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

Software capture

00	The up-counter4 value is loaded to CAP1.
01	Don't care

Figure 3.8 (3) 16-Bit Timer Controller Register (T4MOD)

T4FFCR
(0039H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write			R/W				W	
After reset			0	0	0	0	1	1
Function			TFF4 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: don't care	
			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.

Timer flip-flop 4 (TFF4) invert trigger

00	Trigger disable (Invert prohibition)
01	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

Figure 3.8 (4) 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				
After reset			1	0	0	0	0	0
Function			0: Soft-Capture 1: don't care	Capture timing 00: Disable INT6 occurs at rising edge. 01: TI6↑ TI7↑ INT6 occurs at rising edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rising edge.		1: UC5 Clear Enable	Timer 5 source clock 00: TI6 pin 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 5 input clock

00	External clock (TI6)
01	ϕT1
10	ϕT4
11	ϕT16

Clearing the up-counter UC5

00	Clear disable
01	Clear by match with TREG7

Timer 5 Capture timing

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

Software capture

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

Figure 3.8 (5) 16-bit Timer Control Register (T5MOD)

T5FFCR (0049H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
Read/Write			R/W				W	
After reset			0	0	0	0	1	1
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		

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.

Timer flip-flop 6 (TFF6) invert trigger

00	Trigger disable (Invert prohibition)
01	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 3.8 (6) 16-Bit Timer5 F/F Control (T5FFCR)

T45CR (003AH)

	7	6	5	4	3	2	1	0
bit Symbol	QCU						DB6EN	DB4EN
Read/Write	R / W						R/W	
After reset	0						0	0
Function	Watchdog / Warming Up Timer Control						Double buffer 0: Disable 1: Enable Double buffer of TREG6	Double buffer of TREG4

Double buffer control

0	Disable
1	Enable

DB6EN : Double buffer of TREG6
DB4EN : Double buffer of TREG4

Watchdog Timer / Warming Up Timer input control

0	Use 7 stage Binary Counter
1	not Use 7 stage Binary Counter (note 1)

Note 1 : In case of unused 7 state binary counter as a warming-up timer, the stable clock must be input from external circuit.

Note 2 : Bit 6 to 2 of T45CR Is read as “1”.

Figure 3.8 (7) 16-bit Timer Trigger Control Register (T45CR)

TRUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	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

Note : Bit 6 of TRUN is read as "1".

PRRUN : Operation of prescaler
T5RUN : Operation of 16-bit timer (timer5)
T4RUN : Operation of 16-bit timer (timer4)
T3RUN : Operation of 8-bit timer (timer3)
T2RUN : Operation of 8-bit timer (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	1	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}	1 / 4 times clock
01	fs	
10	fc / 16	
11	(reserved)	

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

① Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit Timer 0, 1, 2, 3, 16-bit Timer 4, 5 and serial Interface 0, 1.
Figure 3.8 (9) shows the block diagram. Table 3.8 (1) shows prescaler clock resolution into 8, 16-bit Timer.

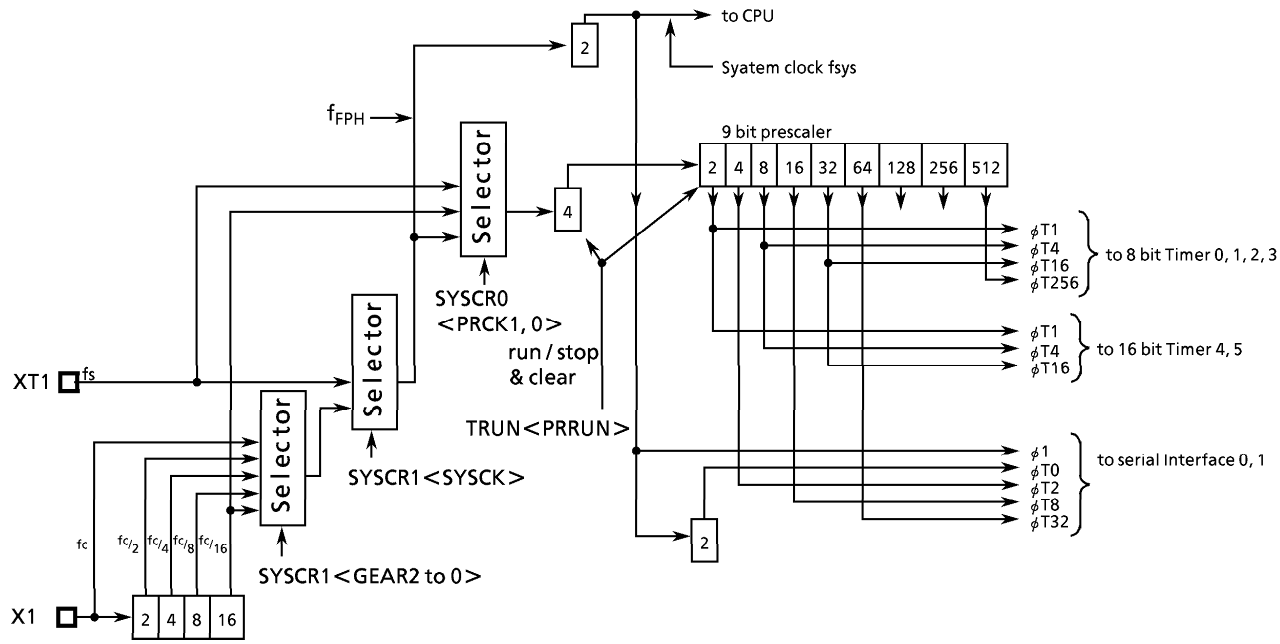


Figure 3.8 (9) The Block Diagram of Prescaler

Table 3.8 (1) Prescaler Clock Resalation to 8, 16 bit Timer

at fc = 20 MHz, fs = 32.768 kHz

Select system clock <SYSCK>	Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^{3}$ (244 μs)	$fs/2^{5}$ (977 μs)	$fs/2^{7}$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
0 (fc)		000 (fc)	$fc/2^{3}$ (0.4 μs)	$fc/2^{5}$ (1.6 μs)	$fc/2^{7}$ (6.4 μs)	$fc/2^{11}$ (102.4 μs)
		001 (fc/2)	$fc/2^{4}$ (0.8 μs)	$fc/2^{6}$ (3.2 μs)	$fc/2^{8}$ (12.8 μs)	$fc/2^{12}$ (204.8 μs)
		010 (fc/4)	$fc/2^{5}$ (1.6 μs)	$fc/2^{7}$ (6.4 μs)	$fc/2^{9}$ (25.6 μs)	$fc/2^{13}$ (409.6 μs)
		011 (fc/8)	$fc/2^{6}$ (3.2 μs)	$fc/2^{8}$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)	$fc/2^{14}$ (819.2 μs)
		100 (fc/16)	$fc/2^{7}$ (6.4 μs)	$fc/2^{9}$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.6384 ms)
XXX	01 (low frequency clock)	XXX	$fs/2^{3}$ (244 μs)	$fs/2^{5}$ (977 μs)	$fs/2^{7}$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
XXX	10 (note) (fc/16 clock)	XXX	$fc/2^{7}$ (6.4 μs)	$fc/2^{9}$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.6384 ms)

XXX : don't care

← 16 bit Timer → (φT1, φT4, φT16)

← 8 bit Timer → (φT1, φT4, φT16, φT256)

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

The clock selected among f_{FPH} clock, fc / 16 clock, and fs clock is divided by 4 and 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 16 bit Timer 4, 5 selects between 3 clock inputs : ϕT1, ϕT4, and ϕT16 among the prescaler outputs.
This 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> 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 P42 / INT4 pin) can be selected. When reset, it will be initialized to <T4CLK1,0> =00 to select TI4 input mode. Counting or stop & clear of the counter is controlled by timer operation control register TRUN<T4RUN>.
When clearing is enabled, up-counter UC4 will be cleared to zero each time it coincides matches the timer register TREG5. The "clear enable/disable" is set by T4MOD<CLE>.
If clearing is disabled, the counter operates as a free-running counter.
A timer overflow interrupt (INTTO4) is generated when UC4 overflow occurs.

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 8 bits in order.

Timer 4

TREG 4		TREG 5	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
000031H	000030H	000033H	000032H

Timer 5

TREG 6		TREG 7	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
000041H	000040H	000043H	000042H

TREG4 to 7 are write-only registers, so they can not be read by software.
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.
After reset, TREG4 and TREG5 are undefined. To use the 16-bit timer after reset, data should be written beforehand.
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 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 latch 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.

Timer 4

CAP 1		CAP 2	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
000035H	000034H	000037H	000036H

Timer 5

CAP 3		CAP 4	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
000045H	000044H	000047H	000046H

CAP 1 to 4 are read-only registers, so it cannot be written by software.

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<CAP12M 1, 0>.

- When T4MOD<CAP12M 1, 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 P42/INT4) input, while data is loaded to CAP2 at the rise edge of TI5 pin (also used as P43 / INT5) input.

- 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.

- 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<CAP1IN> 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 (INTTR4, INTTR5) 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>. After reset, the value of TFF4 is undefined. TFF4 will be inverted when “00” is written in T4FFCR<TFF4C1,0>. Also it is set to “1” when “01” is written, and set to “0” when “10” is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P44). Timer output should be specified by the function register of Port 4. (See Register for Port 4 in figure 3.5. (11).)

(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   ← X  X  0  0  0  0  1  1      Disable trigger.
T4MOD    ← X  X  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 P42 / 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.
P4CR     ← -  -  -  -  -  -  0  -      Set P42 to input mode
INTET54  ← 1  1  0  0  1  0  0  0      Enable INTTR5 and sets interrupt level 4, while
                                       disables INTTR4.
T4FFCR   ← X  X  0  0  0  0  1  1      Disable trigger.
T4MOD    ← X  X  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 : X ; don't care   - ; no change
When used as an event counter, set the prescaler in RUN mode.
(TRUN<PRRUN> = "1")

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

Square wave pulse can be generated at any frequency and duty by timer 4. The output pulse may be either low-active or high-active.
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 P44). 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	
T45CR	←	0 X X X X X - 0	Double Buffer of TRG4 disable
TRUN	←	- X - 0 - - - -	Stop timer 4.
TREG4	←	* * * * * * * *	Set the duty. (16-Bit)
TREG5	←	* * * * * * * *	Set the cycle. (16-Bit)
T45CR	←	0 X X X X X - 1	Double Buffer of TREG4 enable (Change the duty and cycle at the interrupt INTTR5)
T4FFCR	←	X X 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	←	X X 1 0 0 1 * * (** = 01, 10, 11)	Select the internal clock for the input, and disable the capture function.
P4CR	←	- - - 1 - - - -	Assign P44 as TO4.
P4FC	←	- X X 1 X X - X	
TRUN	←	1 X - 1 - - - -	Start timer 4.

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

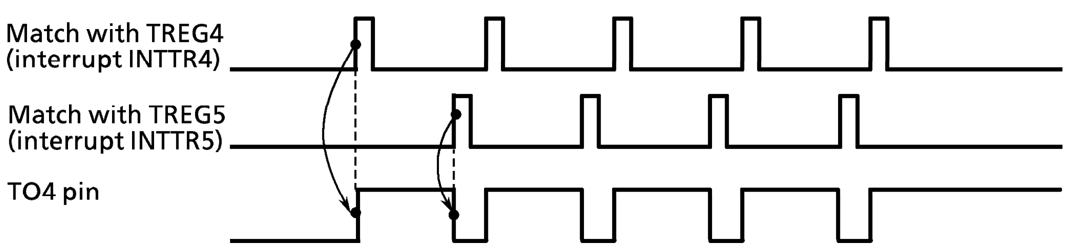


Figure 3.8 (10) Programmable Pulse Generation (PPG) Output Waveforms

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.

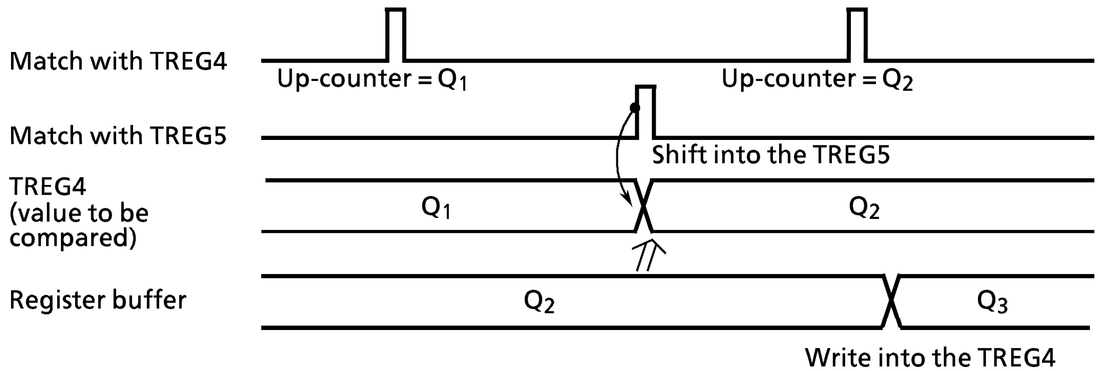


Figure 3.8 (11) Operation of Register Buffer

Shows the block diagram of this mode.

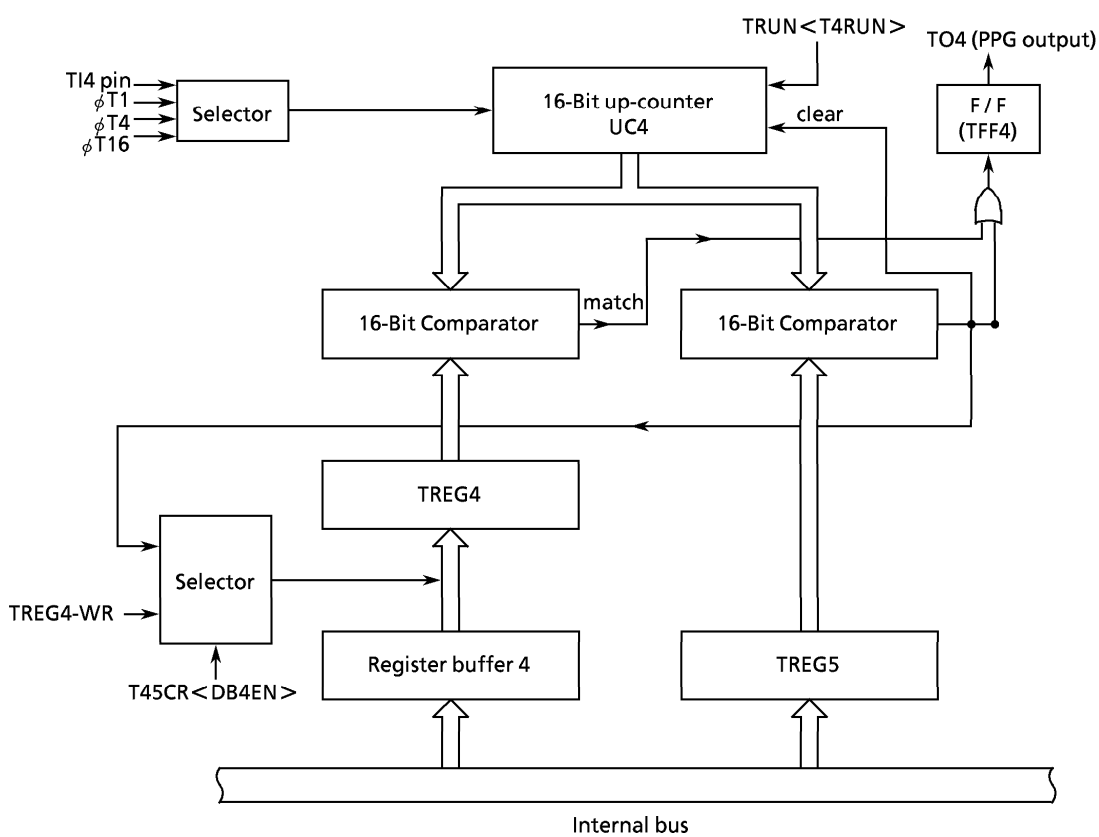


Figure 3.8 (12) Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

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

① One-shot pulse output from external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot Pulse Output from External Trigger Pulse

Set to T4MOD<CAP12M1, 0> = 01.
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.
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 "11" and 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 after one-shot pulse is output.
The (c), (d) and (p) correspond to c, d and p in figure 3.8 (13).

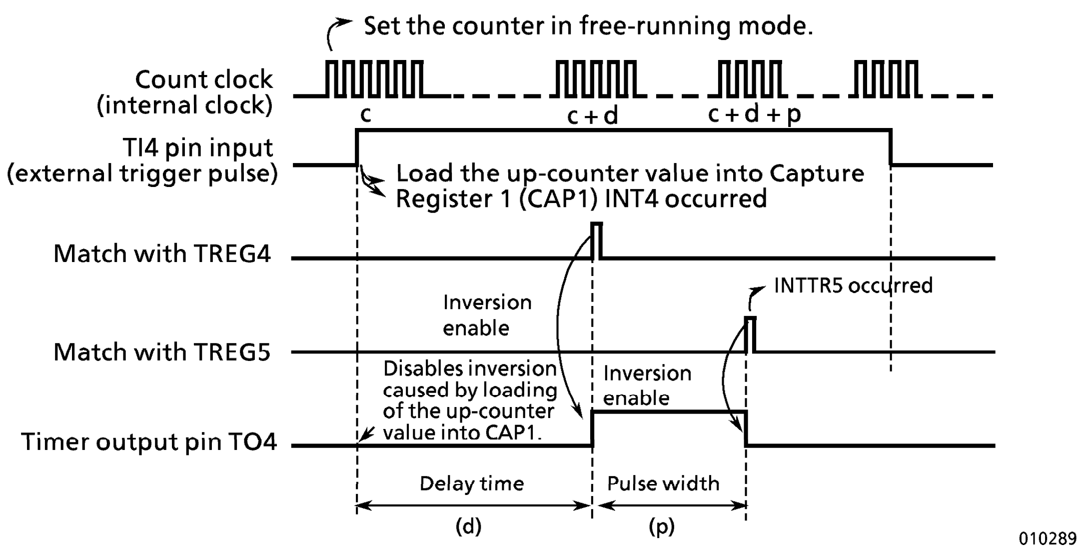


Figure 3.8 (13) One-Shot Pulse Output (with Delay)

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

※ Clock Condition
- system clock : High frequency (fc)
- clock gear : 1 (fc)
- prescaler clock : f_{FPH}

Main setting

```
T4MOD   ← - - 1 0 1 0 0 1      Keep counting (Free-runnig)
                               Count with φT1.
                               Load the up-counter value into CAP1 at the rise edge of TI4 pin input.
T4FFCR  ← X X 0 0 0 0 1 0      Clear TFF4 to zero.
                               Disable TFF4 inversion.
P4CR    ← - - - 1 - - - -   }
P4FC    ← - X X 1 X X - X   }  Select P44 as the TO4 pin.
INTE45  ← - - - - 1 1 0 0      Enable INT4, and disable INTTR4 and INTTR5.
INTET54 ← 1 0 0 0 1 0 0 0
TRUN    ← 1 X - 1 - - - -      Start timer 4.
```

Setting of INT4

```
TREG4   ← CAP1+3ms/φT1
TREG5   ← TREG4+2ms/φT1
T4FFCR  ← X X - - 1 1 - -
                               Enable TFF4 inversion when the up-counter value matches TREG4 or 5.
INTET54 ← 1 1 0 0 - - - -      Enable INTTR5.
```

Setting of INTTR5

```
T4FFCR  ← X X - - 0 0 - -
                               Disable TFF4 inversion when the up-counter value matches TREG4 or 5.
INTET54 ← 1 0 0 0 - - - -      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.

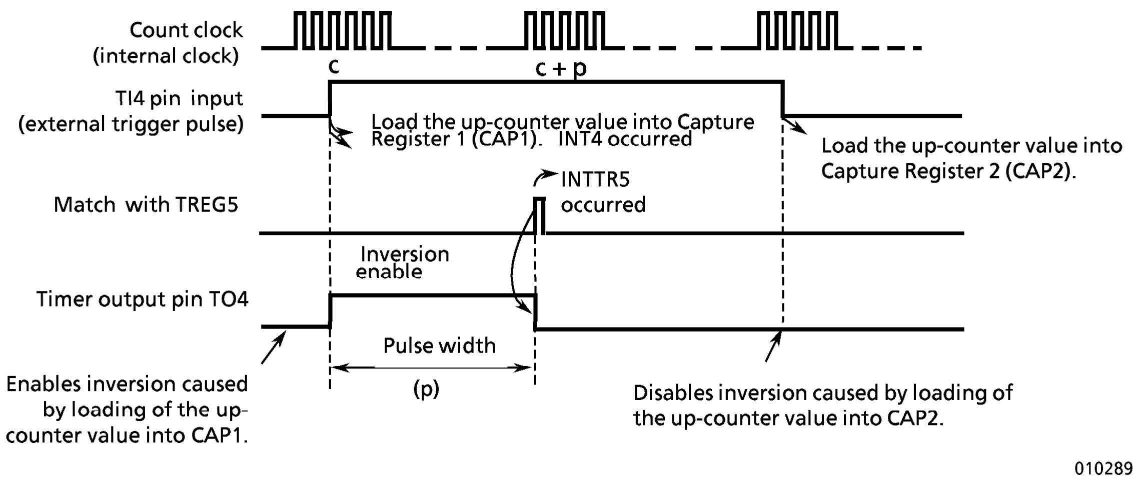


Figure 3.8 (14) One-Shot Pulse Output (without Delay)

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. Set to T4MOD<CAP12M1, 0> = 11.
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.

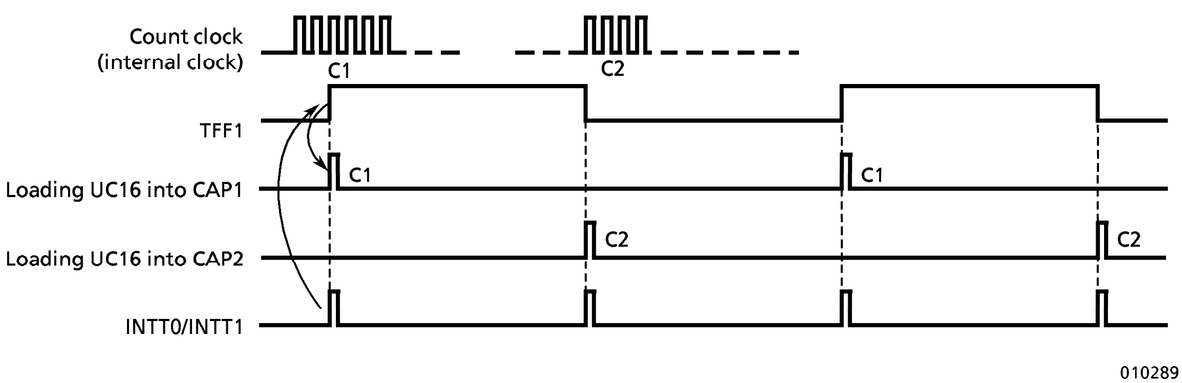


Figure 3.8 (15) Frequency Measurement

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

③ Pulse Width Measurement

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer / event counter counting (free-running) with the internal clock input, the external pulse is input through the 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 $100 \times 0.8\ \mu s = 80\ \mu s$.

Additionally, the pulse width which is over the UC4 maximum count time specified by the clock source can be measured by changing software.

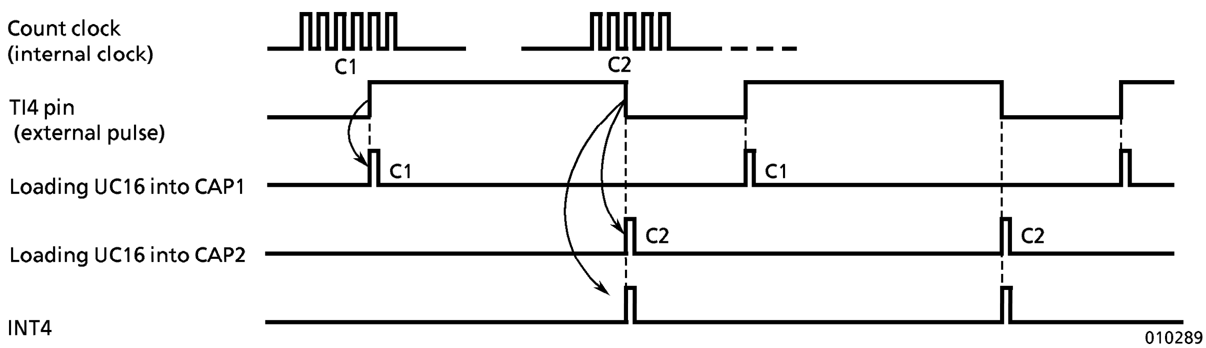


Figure 3.8 (16) Pulse Width Measurement

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.

The width of "L" level can be measured by multiplying the difference between the first C2 and the second C1 at the second INT4 interrupt and the internal clock cycle together.

④ Time Difference Measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through 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.
The time difference between these pulses can be obtained by multiplying the value subtracted CAP 1 from CAP 2 and the internal clock cycle together at which loading the up-counter value into CAP 1 and CAP 2 has been done.

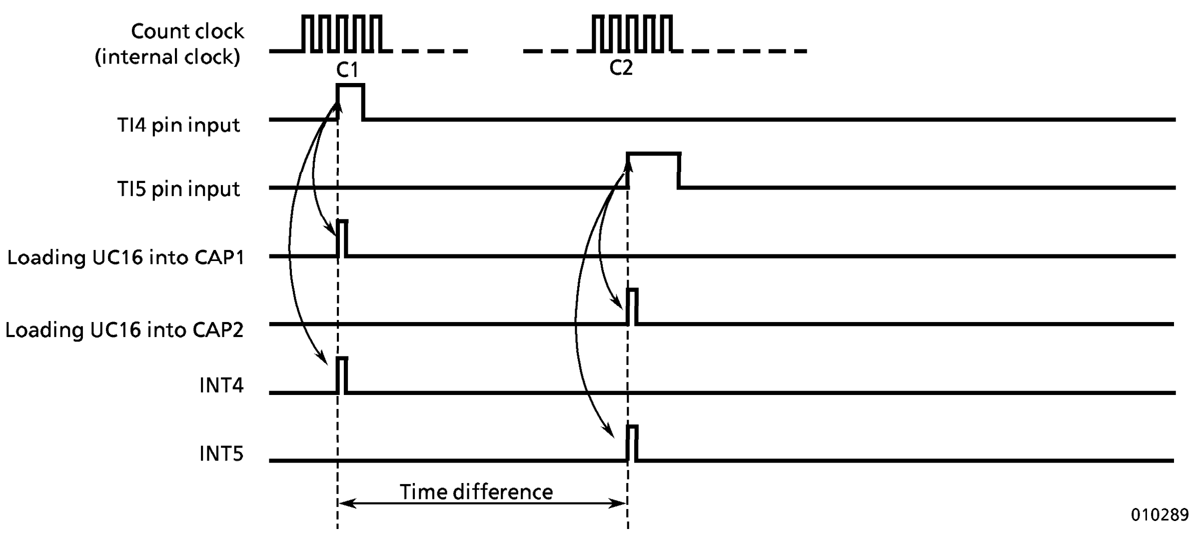


Figure 3.8 (17) Time Difference Measurement

3.9 Serial Channel

TMP93CS44 / TMP93CS45 contains 2 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 0 and 1) — Mode 0 : To transmit and receive I/O data using the synchronizing signal SCLK for extending I/O.
- 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 3.9 (1) shows the data format (for one frame) in each mode.

Serial Channel 0 and 1 can be used independently.

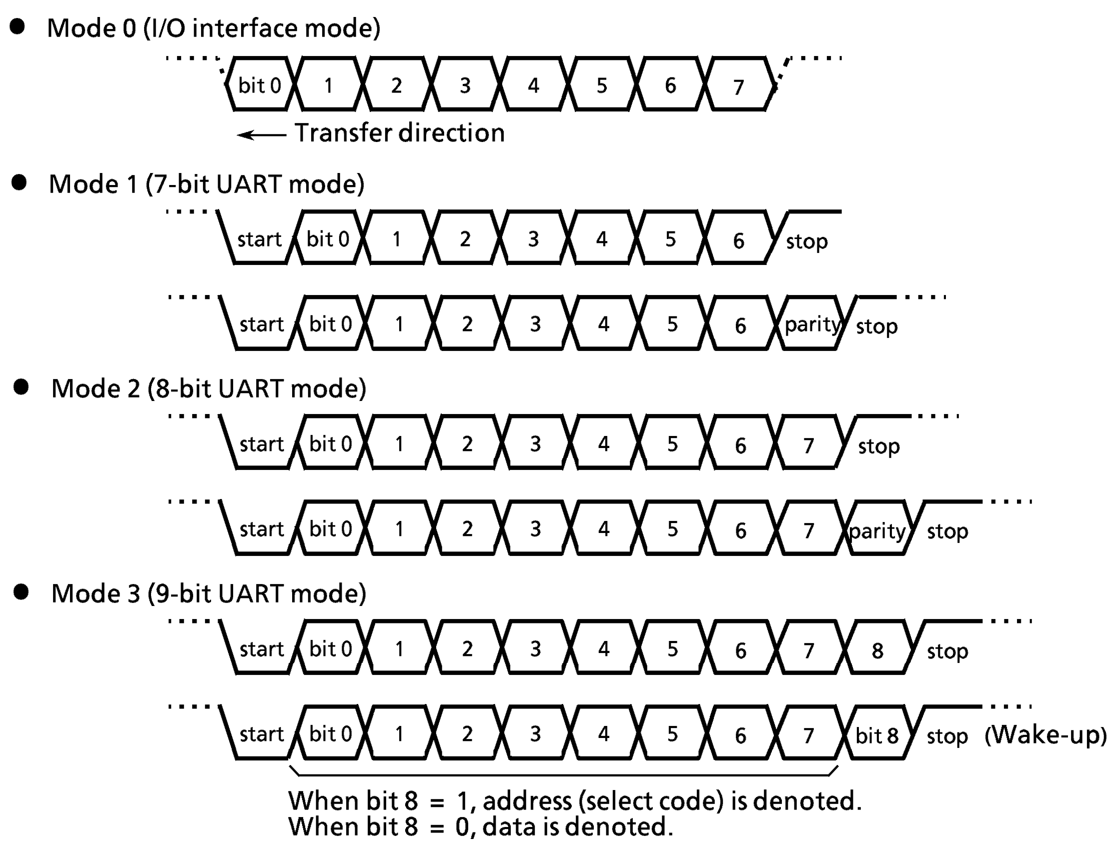


Figure 3.9 (1) Data Formats

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 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (there is no $\overline{\text{RTS}}$ pin, so any 1 port must be controlled by software), it is possible to halt data send until the 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 4 clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the internal prescaler (shared by 8-bit / 16-bit timer) by the value 1 to 16. In addition, serial channel 0 / 1 can operated by using external input clock (SCLK).
In I/O interface mode, it is possible to input synchronous signals as well as to transmit or receive data by external clock.

3.9.1 Control Registers

The serial channel 0 is controlled by 3 control registers SC0CR, SC0MOD and BR0CR. Transmitted and received data are stored in register SC0BUF.
The serial channel 1 has same registers (SC1CR, SC1MOD, BR1CR and SC1BUF).

SC0MOD
(0052H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE0	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data Bit 8	Hand shake Function 0: $\overline{CTS0}$ disable 1: $\overline{CTS0}$ 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: TO2 Trigger 01: baud rate generator 0 10: Internal clock Φ1 11: External clock (SCLK0 input)	

Serial transmission clock (For UART mode)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	External clock (SCLK0 input)

Serial transmission mode

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

Wake up Function
(Don't care in the modes other than 9-bit UART)

0	Disable
1	Enable

Receiving Function

0	Receive Disable
1	Receive Enable

Hand shake function ($\overline{CTS0}$ Pin) enable

0	Disable (always Transferable)
1	Enable

Transmission data bit 8

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit 8

Figure 3.9 (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	Undefined	0	0	0	0	0	0	0
Function	Received data Bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error Overrun	Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud Rate Generator 1: SCLK0 Pin input

Serial transmission clock (For I/O interface mode) (Note1)

0	Baud rate generater
1	SCLK0 Pin input

Edge selection in SCLK pin input mode

0	Transmits and receives (↑) data at rise 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. (Note 2)

Enable parity addition

0	Prohibition (disable)
1	Permission (enable)

Addition / check of even parity

0	Odd parity
1	Even parity

Receving data bit 8

8-bit UART mode (parity)	Stores received parity bit
9-bit UART mode	Stores received data bit 8

Note 1 : To use baud rate generator, set TRUN<PRRUN> to "1, putting the prescaler in RUN mode.
Note 2 : As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

Figure 3.9 (3) Serial Control Register (channel0, SC0CR)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	—		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	(Note) Always write "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 to 1111	1 division (not divided) to 15 divisions	(Note 2)

Selecting the input clock of baud rate generator

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

Note1 : To use baud rate generator, set TRUN<PRRUN> to "1", putting the prescaler in RUN mode.
Note2 : "1 division" of baud rate generator can be used only UART mode. Do not set it in I/O interface mode.
Note3 : Bit 6 of BR0CR is read as "1".
Note4 : Don't read from or write to BR0CR register during sending or receiving.

Figure 3.9 (4) Serial Channel Control (channel 0, BR0CR)

SC0BUF (0050H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	RB07	RB06	RB05	RB04	RB03	RB02	RB01	RB00
	TB07	TB06	TB05	TB04	TB03	TB02	TB01	TB00
Read/Write	R (Receiving) / W (Transmission)							
After reset	Undefined							

Figure 3.9 (5) Serial Transmission / Receiving Buffer Registers (channel 0, SC0BUF)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE1	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfered data Bit 8	Hand shake 0: CTS 1 disable 1: CTS 1 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: TO2 Trigger 01: Baud rate generator 10: Internal clock Φ1 11: Exterrnal clock (SCLK1 input)	

Serial transmission clock (For UART mode)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	Exterrnal clock (SCLK1 input)

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
(Don't care in the modes other than 9-bit UART)

0	Disable
1	Enable

Receiving control

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS1}$ Pin) enable

0	Disable (Always Transferable)
1	Enable

Transmission data bit 8

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit 8

Figure 3.9 (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	Undefined	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	Parity	Framing	0: SCLK1 ( ) 1: SCLK1 ( )	0: Baud rate generator 1: SCLK1 Pin input

Serial transmission clock
(For I/O interface mode) (Note1)

0	Baud rate generate
1	SCLK1 Pin input

Edge selection in SCLK pin input mode

0	Transmits and receives ( ) data at rise 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

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit 8

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

Note 2 : As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

Figure 3.9 (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		R/W					
After reset	0		0	0	0	0	0	0
Function	(Note) Always write "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 to 1111	1 division (not divided) to 15 divisions

(Note 2)

Selecting the input clock of baud rate generator

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

Note1 : To use baud rate generator, set TRUN<PRRUN> to "1", putting the prescaler in RUN mode.
Note2 : "1 division" of baud rate generator can be used only UART mode. Do not set it in I/O interface mode.
Note3 : Bit 6 of BR1CR is read as "1".
Note4 : Don't read from or write to BR1CR register during sending or receiving.

Figure 3.9 (8) Baud Rate Generator Control Register (channel 1, BR1CR)

SC1BUF (0054H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	RB17	RB16	RB15	RB14	RB13	RB12	RB11	RB10
	TB17	TB16	TB15	TB14	TB13	TB12	TB11	TB10
Read/Write	R (Receiving) / W (Transmission)							
After reset	Undefined							

Figure 3.9 (9) Serial Transmission / Receiving Buffer Registers (channel 1, SC1BUF)

P6FC (0016H)

	7	6	5	4	3	2	1	0
bit Symbol			P65F		P63F	P62F		P60F
Read/Write			W		W	W		W
After reset			0		0	0		0
Function			0: PORT 1: SCLK1		0: PORT 1: TxD1	0: PORT 1: SCLK0		0: PORT 1: TxD0

Prohibit Read modify write

Setting TXD0 output P60

0	Port
1	TxD0 output (Channel 0)

Setting SCLK0 output P62

0	Port
1	SCLK0 output (Channel 0)

Setting TXD1 output P63

0	Port
1	TxD1 output (Channel 1)

Setting SCLK1 output P65

0	Port
1	SCLK1 output (Channel 1)

Figure 3.9 (10) Port 6 Function Register (P6FC)

ODE (0058H)

	7	6	5	4	3	2	1	0
bit Symbol					ODE34	ODE33	ODE63	ODE60
Read/Write					R/W			
After reset					0	0	0	0
Function					P34 0: CMOS 1: OPEN Drain	P33 0: CMOS 1: OPEN Drain	P63 0: CMOS 1: OPEN Drain	P60 0: CMOS 1: OPEN Drain

Setting P60 as Open-drain output

0	CMOS output
1	Open-drain output

Setting P63 as Open-drain output

0	CMOS output
1	Open-drain output

Setting P33 as Open-drain output

0	CMOS output
1	Open-drain output

Setting P34 as Open-drain output

0	CMOS output
1	Open-drain output

Note : Bit 7 to 4 of ODE are read as “1”.

Figure 3.9 (11) Port 3 / Port 6 Open Drain Enable Register (ODE)

3.9.2 Configuration

Figure 3.9 (12) shows the block diagram of the serial channel 0.

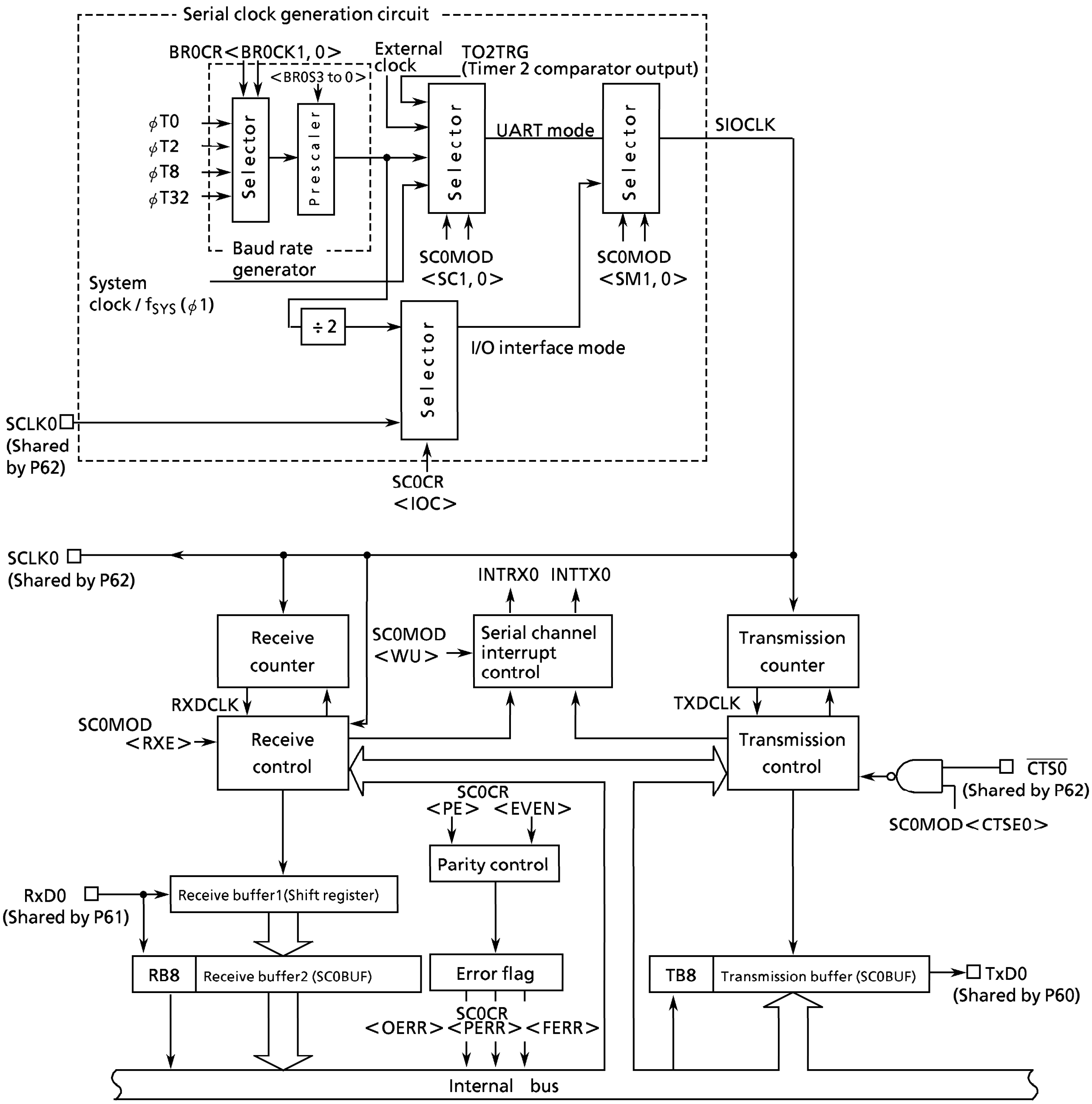


Figure 3.9 (12) Block Diagram of the Serial Channel 0

Figure 3.9 (13) shows the block diagram of the serial channel 1.

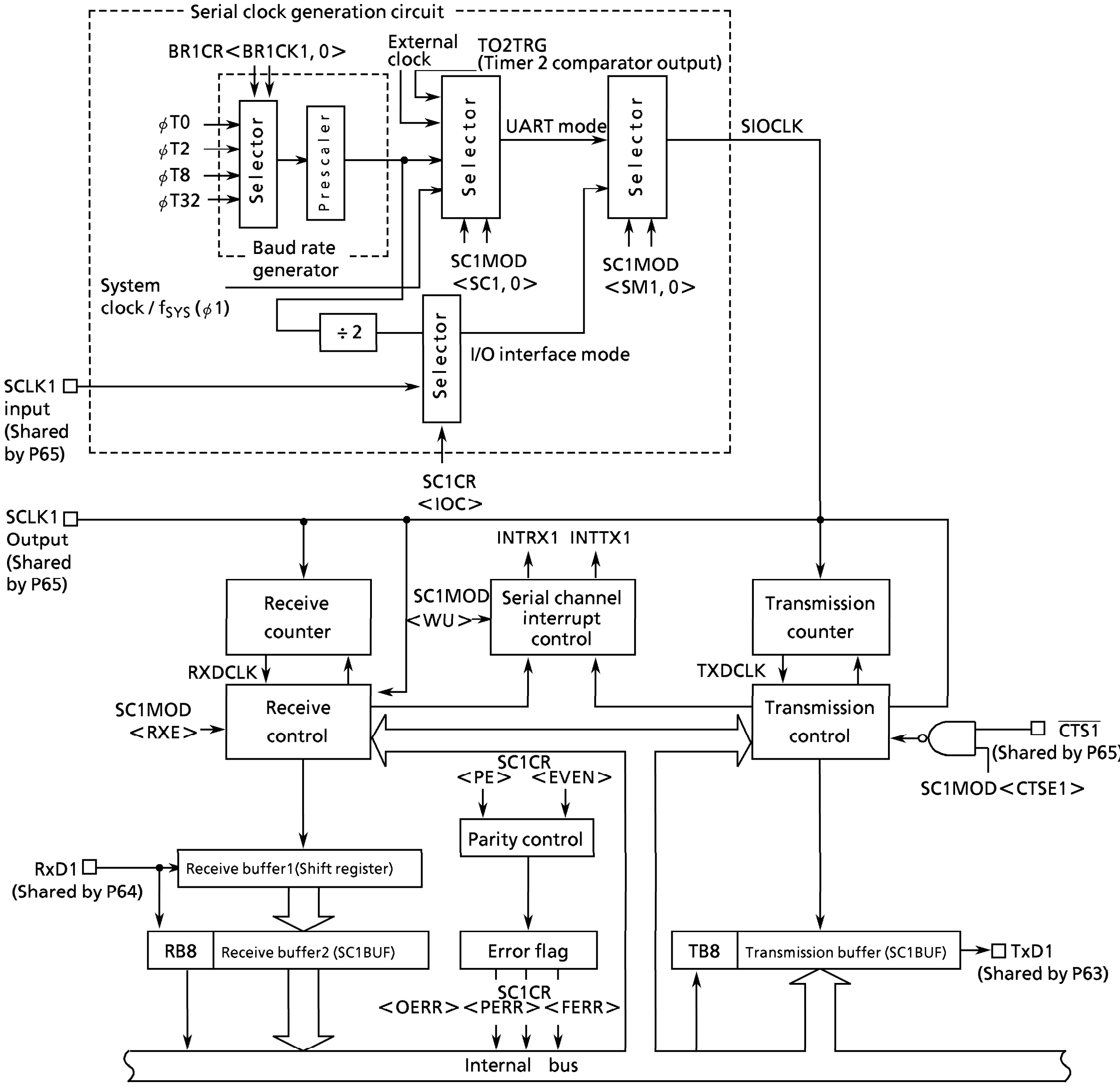


Figure 3.9 (13) Block Diagram of the Serial Channel 1

① Prescaler

There are 9 bit prescaler and prescaler clock selection registers to generate input clock for 8 bit Timer0, 1, 2, 3, 16 bit Timer4, 5, and Serial Interface0, 1.
Figure 3.9 (14) shows the block diagram. Table 3.9 (1) shows prescaler clock resolution into the baud rate generator.

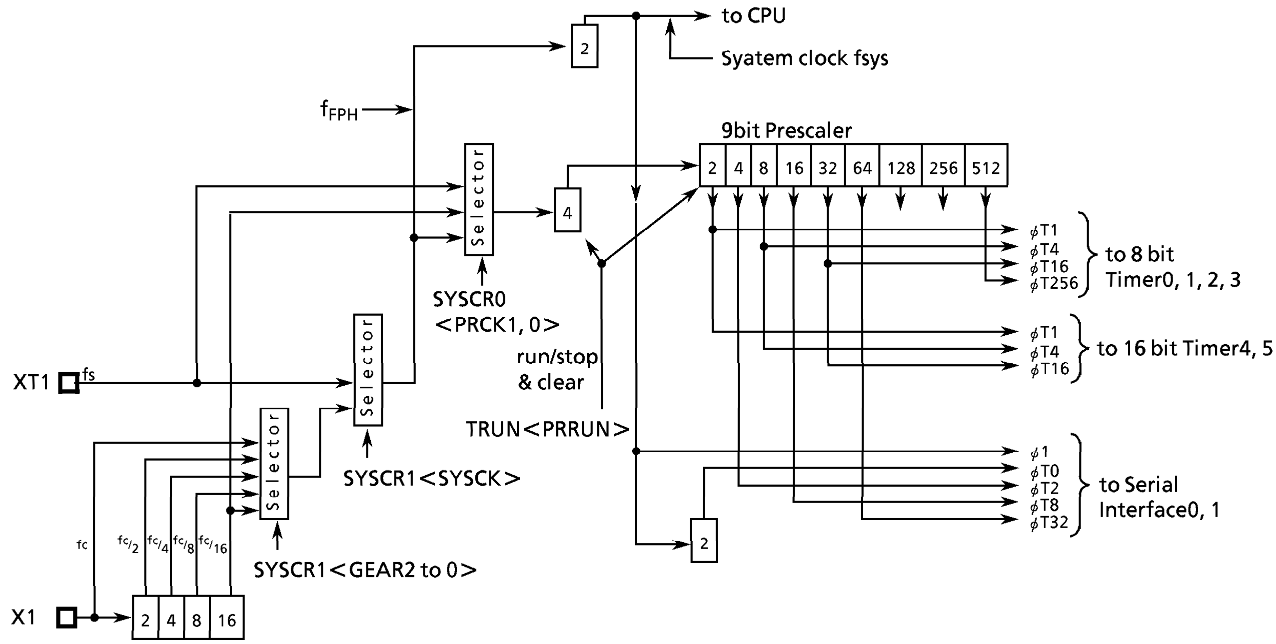


Figure 3.9 (14) The Block Diagram of Prescaler

Table 3.9 (1) Prescaler Clock Resolution to Baud Rate Generator

Select system clock <SYSCK>	Select Prescaler Clock <PRCK1, 0>	Gear value <GEAR2 to 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 clock selected among f_{FPH} clock, fc/16 clock, and fs clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1, 0>.
Resetting sets <PRCK1, 0> to "00" and selects the f_{FPH} clock input divided by 4.
The Baud Rate Generator selects between 4 clock inputs : ϕT0, ϕT2, ϕT8, and ϕT32 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 reduce the power consumption of 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, ϕT0, ϕT2, ϕT8, or ϕT32 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 1 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

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

- I/O interface mode

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

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:

※ Clock Condition

System clock	:	High Frequency (fc)
clock gear	:	1 (fc)
prescaler clock	:	f_{FPH}

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

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

The maximum baud rate of this baud rate generator is 307.2K bps.

Table 3.9 (2) shows an example of the transfer rate in UART mode.

Also with 8-bit timer 2, the serial channel can get a transfer rate. Table 3.9 (3) shows an example of baud rate using timer 2.

Table 3.9 (2) Selection of UART 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	1	153.600	38.400	9.600	2.400
↑	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
↑	16	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	10	19.200	4.800	1.200	0.300
14.745600	1	230.400	57.600	14.400	3.600
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	12	19.200	4.800	1.200	0.300
17.2032	7	38.400	9.600	2.400	0.600
↑	14	19.200	4.800	1.200	0.300
19.6608	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	16	19.200	4.800	1.200	0.300

Note1 : Transfer rate in I/O interface mode is 8 times faster than the values given in the above table.

Note2 : This table is calculated when fc is selected as a system clock, fc/1 as a clock gear, and the system clock as a prescaler clock.

Table 3.9 (3) Selection of UART Transfer Rate (2)
(When Timer 2 (input Clock ϕ T1) is used)

Unit (Kbps)

TREG2 \ fc	19.6608 MHz	14.7456 MHz	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	153.6	115.2	96		76.8	62.5	48
2H	76.8	57.6	48		38.4	31.25	24
3H	51.2	38.4	32	31.25			16
4H	38.4	28.8	24		19.2		12
5H	30.72	23.04	19.2				9.6
8H	19.2	14.4	12		9.6		6
AH	15.36	11.52	9.6				4.8
10H	9.6	7.2	6		4.8		3
14H	7.68	5.76	4.8				2.4

How to calculate the transfer rate (when timer 2 is used):

$$\text{Transfer rate} = \frac{f_{FPH}}{\text{TREG2} \times 8 \times 16}$$

(When Timer 2 (input clock ϕT1) is used)

Note1 : Timer 2 match detect signal cannot be used as the transfer clock in I/O interface mode.

Note2 : This table is calculated when fc is selected as a system clock, fc/1 as a clock gear, and the 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
When in SCLK output mode with the setting of SC0CR<IOC> = "0", the basic clock will be generated by dividing by 2 the output of the baud rate generator described before. When in SCLK input mode with the setting of SC0CR<IOC> = "1", the rising edge or falling edge will be detected according to the setting of SC0CR<SCLKS> register to generate the basic clock.

2) UART mode
According to the setting of SC0CR <SC1, 0>, the above baud rate generator clock, internal clock $\phi 1$ (max 625Kbps at fc = 20 MHz), the match detect signal from timer 0, or external clock SCLK0 will be selected to generate the basic clock SIOCLK.

④ Receiving Counter

The receiving counter is a 4-bit binary counter used in UART mode and counts up by SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 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

When in SCLK 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 SCLK 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) 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 3 samples, it is recognized as start bit and the receiving operation is started.
Data being received are also evaluated by the rule of majority.

⑥ Receiving Buffer

To prevent overrun error, the receiving buffer has a double buffer structure.
Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data is stored in the receiving buffer 1, the stored data are transferred to the receiving buffer 2 (SC0BUF), generating an interrupt INTRX0. 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> is 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 UART mode and, counts by SIOCLK clock, generating TxDCLK every 16 clock pulses.

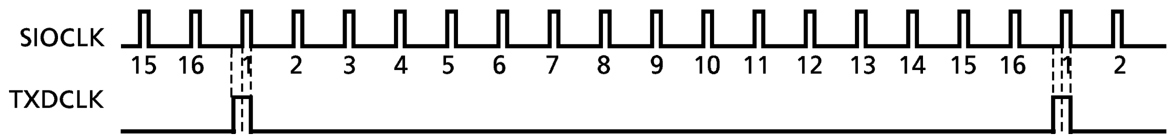


Figure 3.9 (15) Generation of Transmission Clock

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 of 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) UART mode

When transmission data are 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

The serial channels use the $\overline{\text{CTS0}}$ pin to transmit data in units of frames, thus preventing an overrun error. Use SC0MOD<CTSE0> to enable or disable the handshake function.
When $\overline{\text{CTS0}}$ goes high, data transmission is halted after the completion of the current transmission and is not restarted until $\overline{\text{CTS0}}$ returns to low. An INTTX0 interrupt is generated to request the CPU for the next data to transmit. When the CPU write the data to the transmit buffer, processing enters standby mode.
An $\overline{\text{RTS}}$ pin is not provided, but a handshake function can easily be configured if the receiver sets any port assigned to the $\overline{\text{RTS}}$ function to high (in the receive interrupt routine) after data receive, and requests the transmitter to temporarily halt transmission.

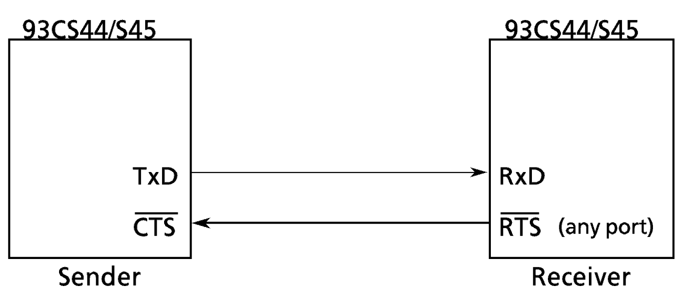
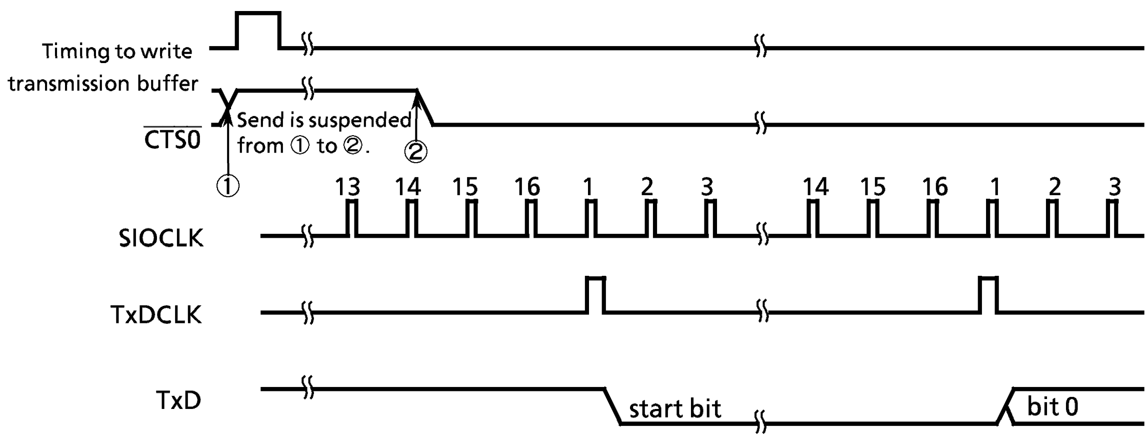


Figure 3.9 (16) Handshake Function



Note 1 : If the $\overline{\text{CTS}}$ 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 $\overline{\text{CTS}}$ signal falls.

Figure 3.9 (17) Timing of $\overline{\text{CTS}}$ (Clear to send)

⑨ Transmission Buffer

Transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU. 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 SC0BUF, 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 are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SC0BUF), 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 are stored in receiving buffer 2 (SC0BUF), an overrun error will occur.

2. Parity error <PERR>

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

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 Signal Generation Timing

1) In I/O Interface mode

Timing for send interrupt generation	SCLK0 output mode	Immediately after rise of last SCLK0 signal (See Figure 3.9 (20))
	SCLK0 input mode	Immediately after rise (rising mode) or fall (falling mode) of last SCLK0 signal (See Figure 3.9 (21).)
Timing for receive interrupt generation	SCLK0 output mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.9 (22).)
	SCLK0 input mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.9 (33).)

2) In UART mode

Receive

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Around center of bit 8	Around center of parity bit	Around center of stop bit
Timing for framing error generation	Around center of stop bit	Around center of stop bit	Around center of stop bit
Timing for parity error generation	————	Around center of parity bit	←
Timing for overrun error generation	Around center of bit 8	Around center of parity bit	Around center of stop bit

Send

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Immediately before stop bit sent	←	←

3.9.3 Operational Description

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

This mode is used to increase the number of I/O pins of for transmitting or receiving data to or from the external shifter register.
This mode includes SCLK output mode to output synchronous clock SCLK0 and SCLK input mode to input external synchronous clock SCLK0.

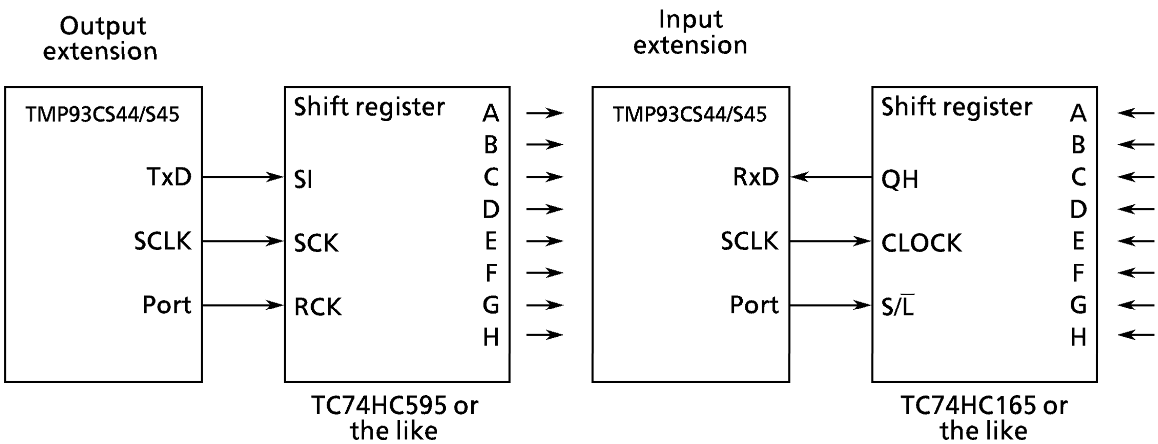


Figure 3.9 (18) Example of SCLK Output Mode Connection

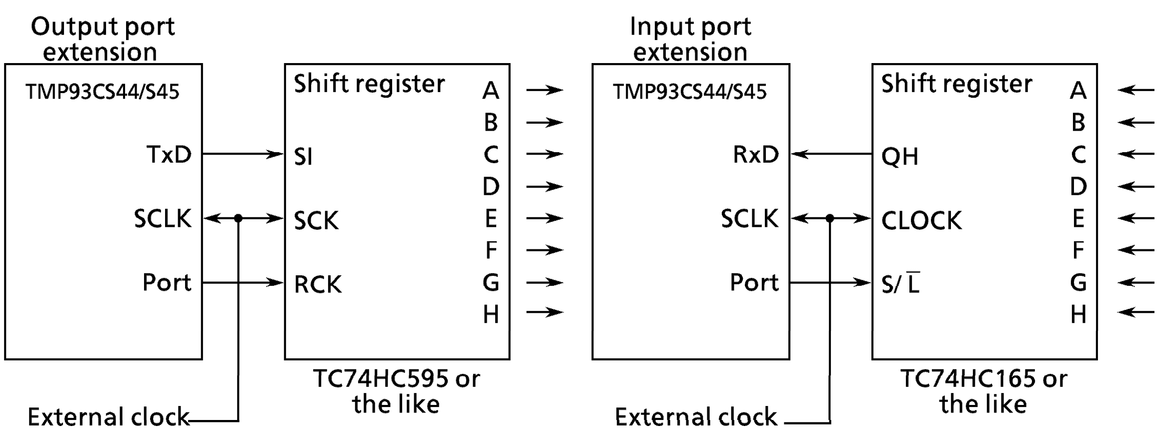


Figure 3.9 (19) Example of SCLK Input Mode Connection

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<ITX0C> will be set to generate INTTX0 interrupt.

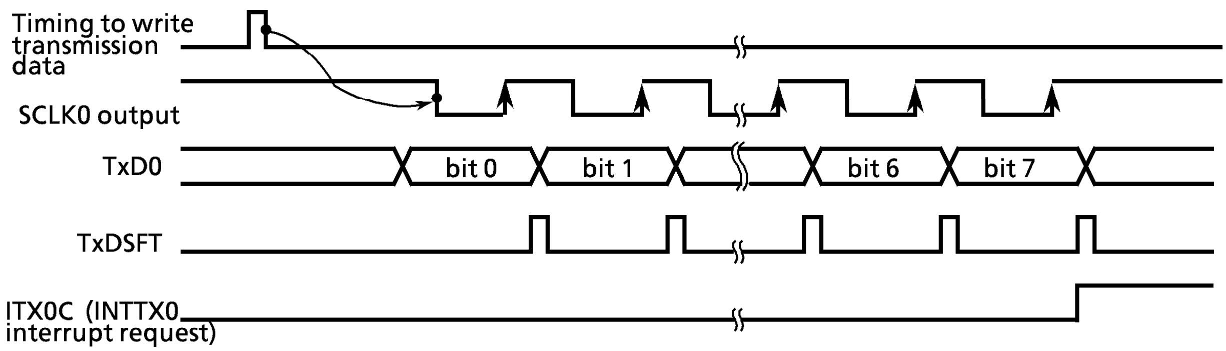


Figure 3.9 (20) Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

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.

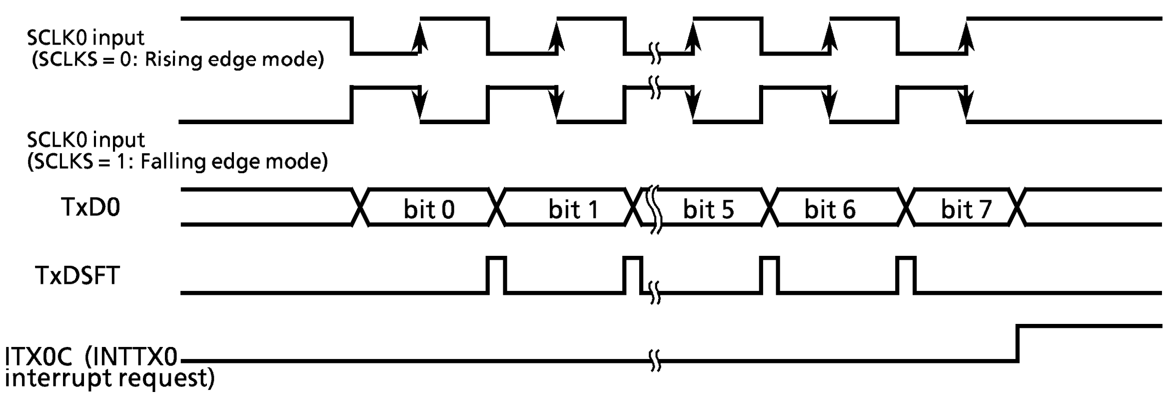


Figure 3.9 (21) Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

In SCLK output mode, synchronous clock is outputted from SCLK0 pin and the data are 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.

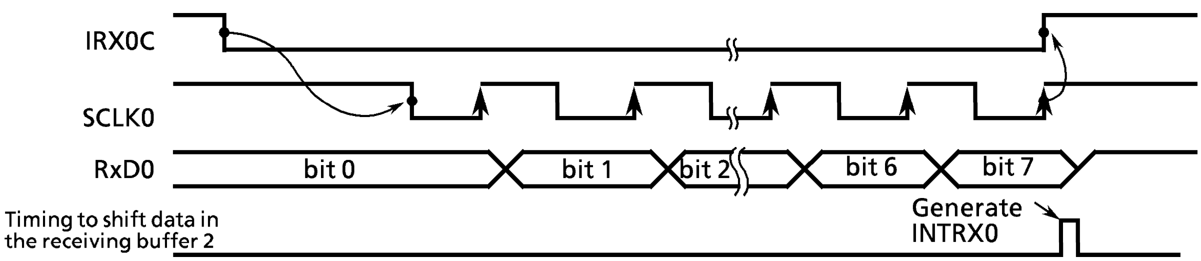


Figure 3.9 (22) Receiving Operation in I/O Interface Mode (SCLK Output Mode)

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.

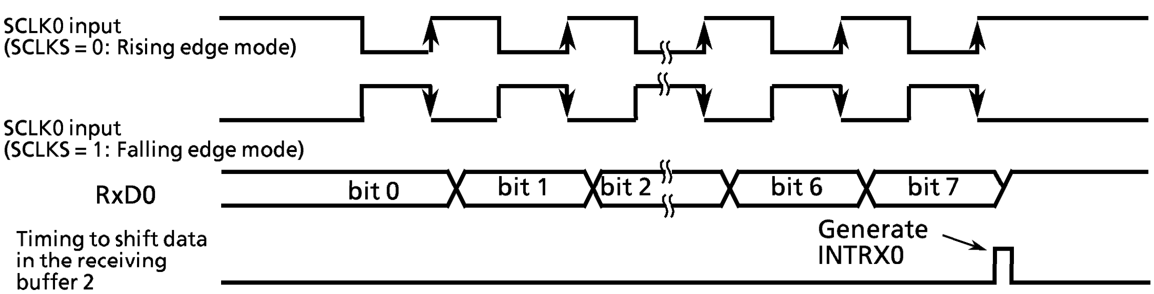


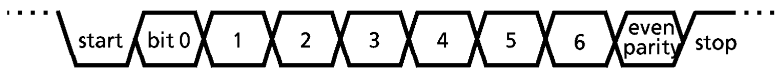
Figure 3.9 (23) Receiving Operation in I/O Interface Mode (SCLK Input Mode)

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

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

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.



← Direction of transmission (transmission rate: 2400 bps at fc = 12.288 MHz)
※ Clock Condition
System clock : High Frequency (fc)
clock gear : 1 (fc)
prescaler clock : f_{FPH}

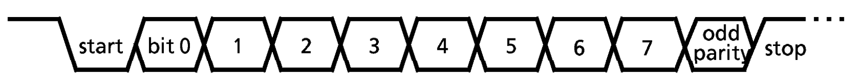
```
          7 6 5 4 3 2 1 0
P6CR   ←  - - - - - - - 1    } Select P60 as the TxD pin.
P6FC   ←  X X - X - - X 1    }
SC0MOD ←  X 0 - X 0 1 0 1    Set 7-bit UART mode.
SC0CR  ←  X 1 1 X X X 0 0    Add an even parity.
BR0CR  ←  0 X 1 0 0 1 0 1    Set transfer rate at 2400 bps.
TRUN   ←  1 X - - - - - -    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)

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.



← Direction of transmission (transmission rate: 9600 bps at fc = 12.288 MHz)
※ Clock Condition
System clock : High Frequency (fc)
clock gear : 1 (fc)
prescaler clock : f_{FPH}

Main setting

```
          7 6 5 4 3 2 1 0
P6CR   ← - - - - - - 0 -     Select P61 (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.
```

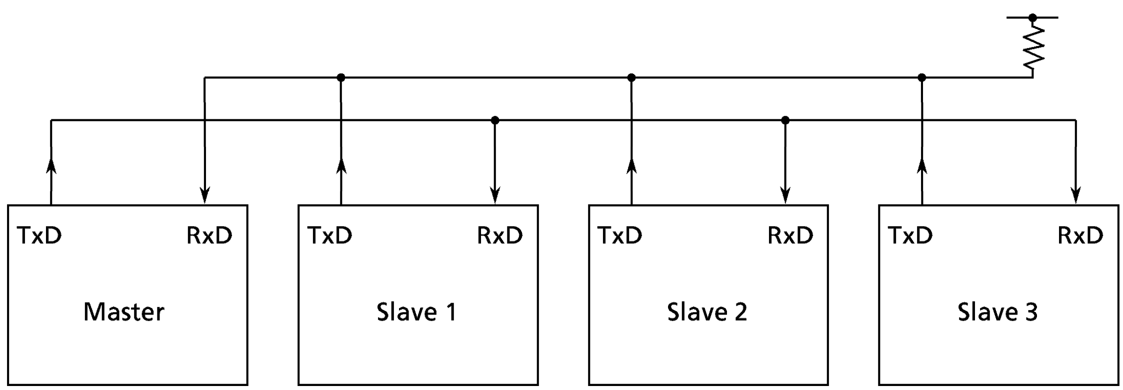
Note : X ; don't care - ; no change

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

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 SC0MOD <TB8>, while in receiving it is stored in SC0CR<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.

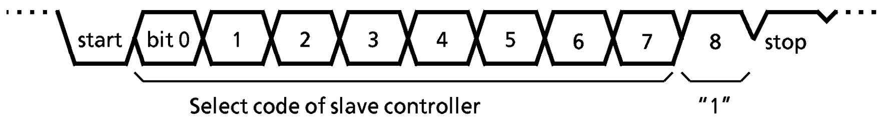
Figure 3.9 (24) Serial Link Using Wake-Up Function

Protocol

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

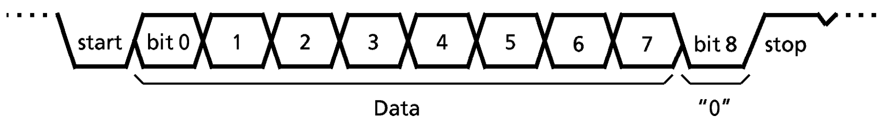
② Set SC0MOD<WU> bit of each slave controller to "1" to enable data receiving.

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".



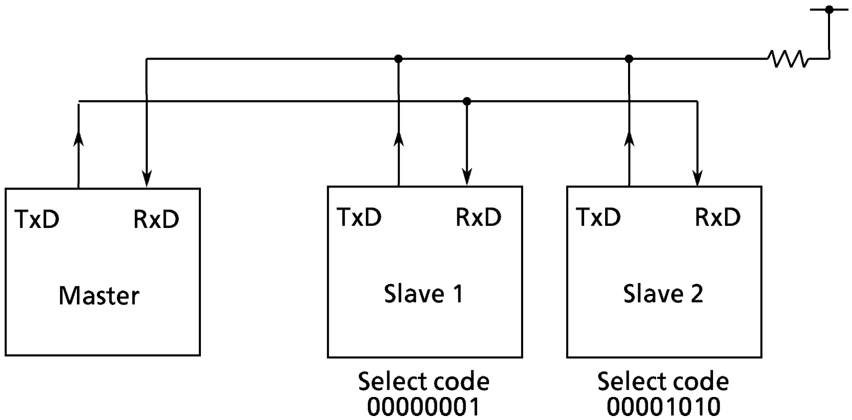
④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.

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 $\phi 1$ 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

```
P6CR   ← - - - - - - 0 1   } Select P60 as TxD0 pin and P61 as RxD0 pin.
P6FC   ← X X - X - - X 1   }
INTES0 ← 1 1 0 0 1 1 0 1    Enable INTTX0 and set the interrupt level 4.
                            Enable INTRX0 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

```
P6CR   ← - - - - - - 0 1   } Select P61 as RxD0 pin and P60 as TxD0 pin (open drain
P6FC   ← X X - X - - X 1   } output).
ODE    ← 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".
```

3.10 Serial Bus Interface (SBI)

The TMP93CS44/S45 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I²C bus.
The serial bus interface is connected to an external device through P33 (SDA) and P34 (SCL) in the I²C bus mode; and through P32 (SCK), P33 (SO), and P34 (SI) in the clocked-synchronous 8-bit SIO mode.
TMP93CS44/S45 has no an arbitration function which is necessary when two or more master devices scramble for the bus control. In master mode, other devices which are connected on the same bus need be slave devices. (single master)

Setting of every pins is as follows.

	ODE<ODE34, 33>	P3CR<P34C, P33C, P32C>	P3FC<P32M, P34F, P33F, P32F>
I²C Bus Mode	11	11X	X110
Clock synchronous 8-bit SIO Mode	XX	011 010	1111

X : Don't care

3.10.1 Configuration

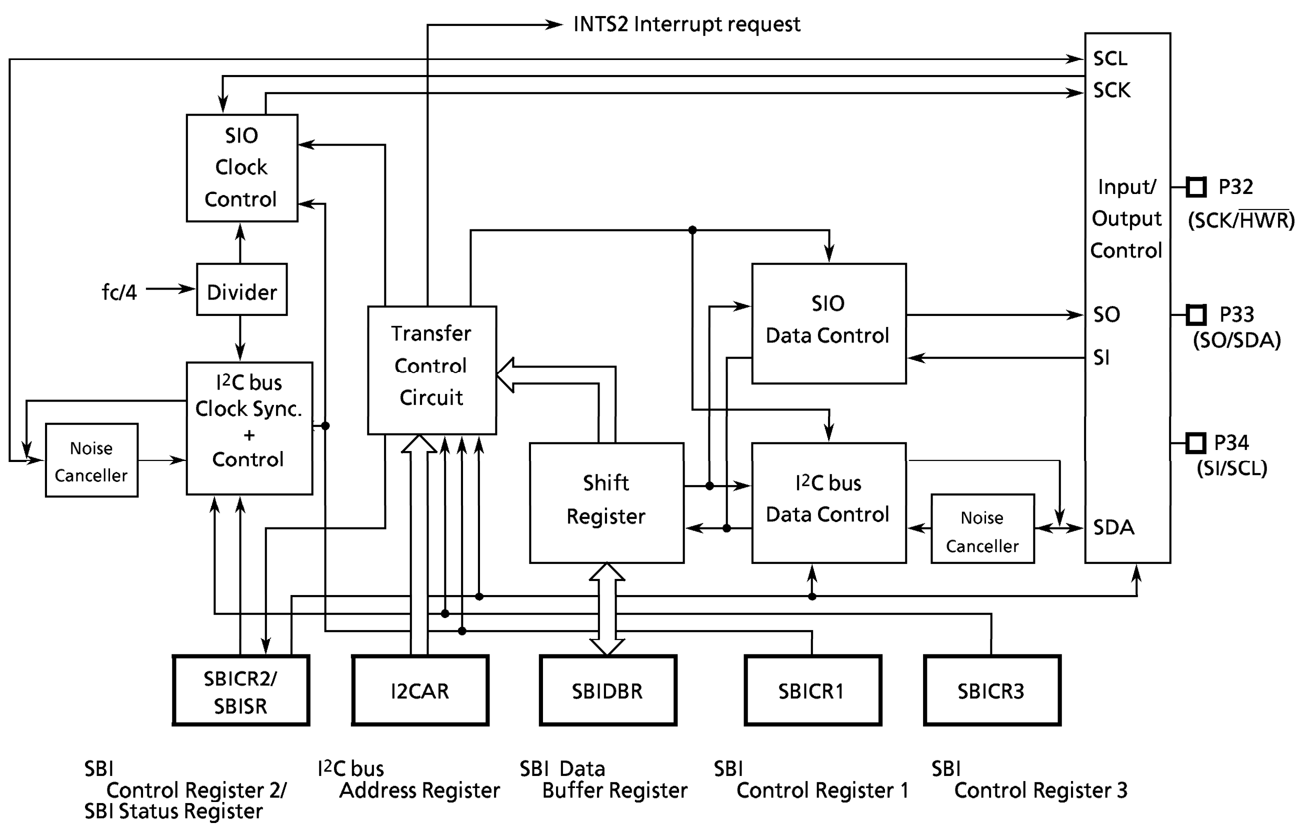


Figure 3.10 (1) Serial Bus Interface (SBI)

3.10.2 Serial Bus Interface (SBI) Control

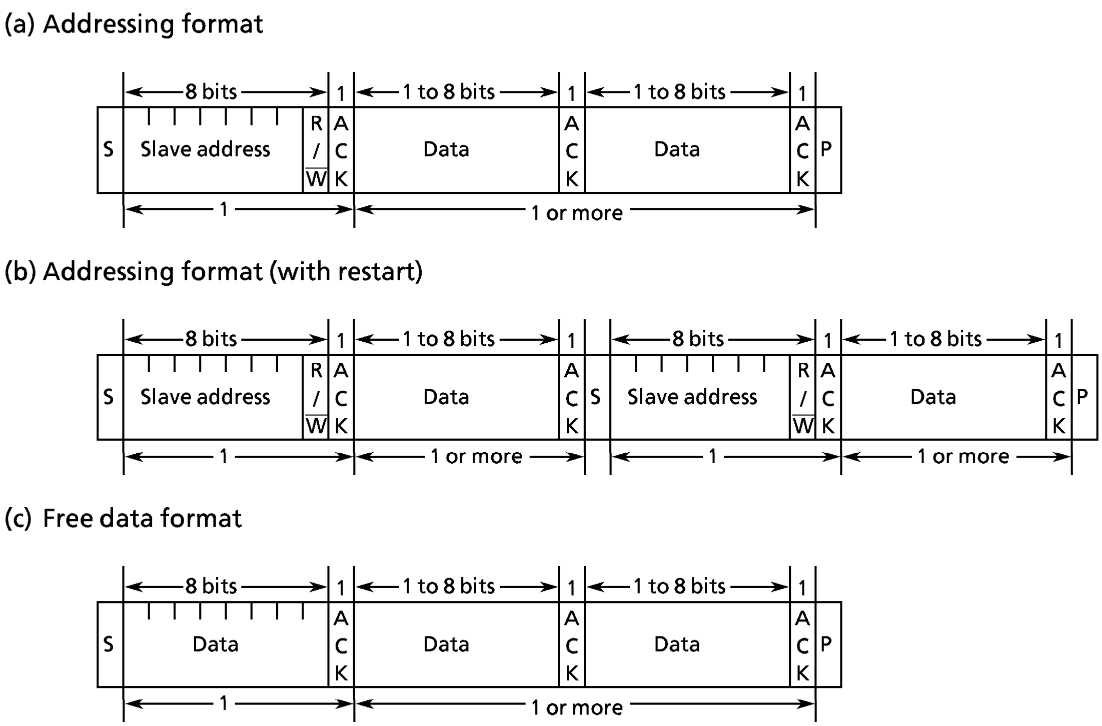
The following reginsters are used for control and operation status monitoring when using the serial bus interface (SBI).

- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface control register 3 (SBICR3)
- Serial bus interface data buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

The above registers differ depending on an mode to be used.
Refer to Section "3.10.4 I^2C bus Mode Control" and "3.10.6 Clocked-synchronous 8-bit SIO Mode Control".

3.10.3 The Data Formats in the I2C Bus Mode

The data formats when using the TMP93CS44 / S45 in the I^2C bus mode are shown below.



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

Figure 3.10 (2) Data format in the I^2C Bus Mode

3.10.4 I²C Bus Mode Control

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

Serial Bus Interface Control Register 1

SBICR1 (004BH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0
Read/Write	W			R/W		W		
After reset	0	0	0	0		0	0	0
Function	Number of transferred bits (Note1)			Acknowledge mode specification		Serial clock selection (Note2)		

Serial clock selection

000	454.5 kHz	System clock : fc Clock gear : fc/1 fc = 20MHz (Output on SCL PIN)
001	263.2 kHz	
010	142.9 kHz	
011	74.6 kHz	
100	38.2 kHz	
101	19.3 kHz	
110	9.71 kHz	
111	(reserved)	

Acknowledge mode specification

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

Number of transferred bits

<BC2 to 0>	<ACK> = 0		<ACK> = 1	
	Number of Clock	Bits	Number of Clock	Bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Note 1 : Set <BC2 to 0> to "000" before switching to a clock-synchronous 8-bit SIO mode.
Note 2 : Refer to sentence of 3.10.4 (3) Serial Clock.

Figure 3.10 (3) - 1 Register for I²C Bus Mode

Serial Bus Interface Control Register 2

SBICR2 (004EH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	–	–
Read/Write	W (R/W Note1)				W			
After reset	0	0	0	1	0	0	0	0
Function	Master / slave selection	Trans-mitter / receiver selection	Start /stop genera-tion	Cancel INTS2 request	Serial bus interface operating mode selection (Note 2)			

SBI operating mode selection

00	Port mode (serial bus interface output disable)
01	SIO mode
10	I^2C bus mode
11	(Reserved)

Cancel INTS2 request

0	–
1	Cancel interrupt service request

Stare / stop generation

0	Gnerate the stop condition
1	Gnerate the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / slave selection

0	Slave
1	Master

Note 1 : This register functions as the SBISR by reading.

Note 2 : Switch a mode to the port mode after confirming that the bus is free.
Switch a mode to the I^2C bus mode and the clocked-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.10 (3) - 2 Register for I^2C Bus Mode

Serial Bus Interface Status Register

SBISR (004EH)

Prohibit read-modify-write.

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master / stave selection status monitor	Transmitter / receiver selection status monitor	I^2C bus status monitor	INTS2 request status monitor	Noise detection monitor	Slave address match detection monitor	"GENERAL CALL" detection monitor	Last received bit monitor

Last received bit monitor

0	Last received bit "0"
1	Last received bit "1"

"GENERAL CALL" detection monitor

0	Non "GENERAL CALL" detected
1	"GENERAL CALL" detected

Slave address match detection monitor

0	Non slave address match or "GENERAL CALL" detected
1	Slave address match or "GENERAL CALL" detected

Noise detection monitor

0	Non noise detected
1	Noise detected

INTS2 request status

0	Interrupt service requested
1	Interrupt service canceled

I^2C bus status monitor

0	Bus free
1	Bus busy

Transmitter / receiver selection status monitor

0	Receiver
1	Transmitter

Master / slave selection status monitor

0	Slave
1	Master

Note : Bits 7 to 2 of this register function as the SBICR2 by writing.

Figure 3.10 (3) - 3 Register for I^2C Bus Mode

Serial Bus Interface Control Register 3

SBICR3 (004FH)

	7	6	5	4	3	2	1	0
bit Symbol								SWRTS
Read/Write								R/W
After reset								0
Function								Software reset 0: Don't care 1: Initialize SBI

Software reset

0	Don't care
1	Initialize SBI (After initializing SBI, SWRST is automatically cleared to 0).

Serial Bus Interface Data Buffer Register

SBIDBR (004CH)
Prohibit read-modify-write.

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Note1 : When writing the send data, start from the MSB (bit7).

I^2C Bus Address Register

I2CAR (004DH)
Prohibit read-modify-write.

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Slave address selection.							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10 (3) - 4 Registers for I^2C Bus Mode

(1) Acknowledge mode specification

Set SBICR1<ACK> to “1” for operation in the acknowledge mode. The TMP93CS44 / S45 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.
Set <ACK> to “0” for operation in the non-acknowledge mode. The TMP93CS44 / S45 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

In the acknowledgment mode, when the TMP93CS44/S45 is the slave mode, clocks are counted for the acknowledge signal. During the clock for the acknowledge signal, when a received slave address matches to a slave address set to the I2CAR or a “GENERAL CALL” is received, the SDA pin is set to low level generating an acknowledge signal.
After a received slave address matches to a slave address set to the I2CAR and a “GENERAL CALL” is received, in the transmitter mode during the clock for the acknowledge signal, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode, the SDA pin is set to low level generating an acknowledge signal
In the non-acknowledgment mode, when the TMP93CS44/S45 is the slave mode, clocks for the acknowledge signal are not counter.

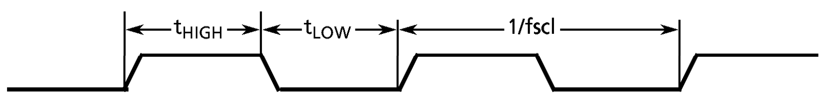
(2) Number of transfer bits

SBICR1<BC2 to 0> are used to select a number of bits for transmitting and receiving data.
Since <BC2 to 0> are cleared to "000" as a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, <BC2 to 0> retain a specified value.

(3) Serial clock

① Clock source

SBICR1 <SCK2 to 0> are used to select a maximum transfer frequency output on the SCL pin in the master mode.



$t_{LOW} = 2^n/f_{FPH}$

$t_{HIGH} = 2^n/f_{FPH} + 12/f_{FPH}$

$$fscl = 1/(t_{Low} + t_{HIGH}) = \frac{f_{FPH}}{2 \times 2^n + 12}$$

<SCK2 to 0> (bits2 to 0 in the SBICR1)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 3.10 (4) Clock Source

② Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which drives a clock line to low-level, in the first place, invalidates a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.
The TMP93CS44 / S45 has a clock synchronization function for normal data transfer even when more than one master exists on a bus.
The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

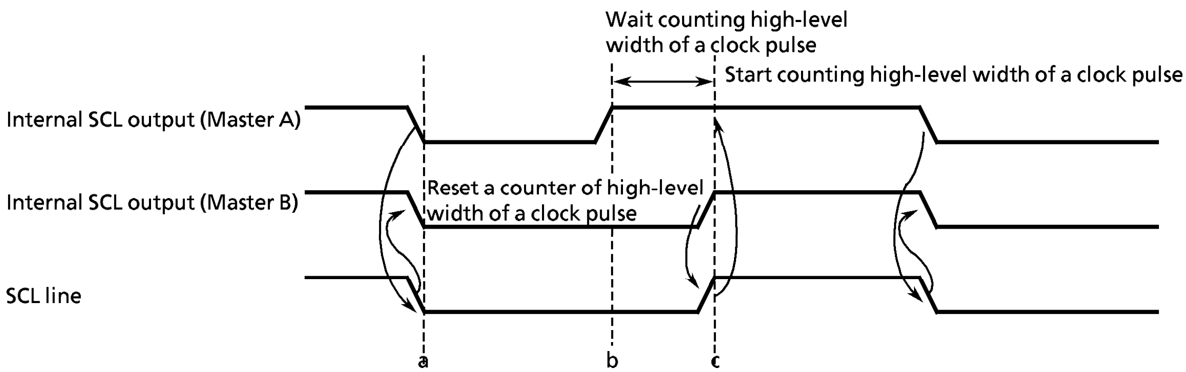


Figure 3.10 (5) Clock Synchronization

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

(4) Slave address and Address recognition mode specification

To operate the TMP93CS44/S45 in the addressing format which recognizes the slave address, set I2CAR<ALS> to “0” and set the slave address to the I2CAR<SA6 to 0>.
To operate the serial bus interface circuit in the free data format which does not recognize the slave address, set <ALS> to “1”. When the TMP93CS44/S45 used in the free data format, the slave address and the direction bit are not recognized. They are handled as data just after generation of start conditions.

(5) Master/slave selection

Set SBICR2 <MST> to “1” for operating the TMP93CS44 / S45 as a master device. <MST> is cleared to “0” by the hardware after a stop condition on a bus is detected or arbitration is lost.

(6) Transmitter/receiver selection

Set SBICR2 <TRX> to “1” for operating the TMP93CS44 / S45 as a transmitter. Set <TRX> to “0” for operation as a receiver. When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2CAR or the GENERAL CALL is received (all 8-bit data are “0” after the start condition), <TRX> is set to “1” by the hardware if the direction bit (R/W) sent from the master device is “1”, and is set to “0” by the hardware if the bit is “0”. In the master mode, after the acknowledge signal is returned from the slave device , <TRX> is set to “0” by the hardware if a transmitted direction bit is “1”, and set to “1” by the hardware if it is “0”. When the acknowledge signal is not returned, the current condition is maintained.
<TRX> is cleared to “0” by the hardware after the stop condition on the I²C bus is detected or arbitration is lost.

The following shows <TRX> change conditions in each mode and <TRX> after changing.

Mode	Direction bit	Change condition	<TRX> after changing
Slave mode	0	A received slave address is the same as a value set to I2CAR.	0
	1		1
Master mode	0	ACK signal is returned.	1
	1		0

When the TMP93CS44/S45 operates in the free data format, the slave address and the direction bit are not recognized. They are handled as data just after generating a start condition. The TRX was not changed by the hardware.

(7) Start/Stop Condition generation

When SBICR2 <BB> is “0”, the start condition and 8-bit data are output by writing “1” to SBICR2<MST, TRX, BB, PIN>. It is necessary to set “1” to SBICR1<ACK> beforehand.

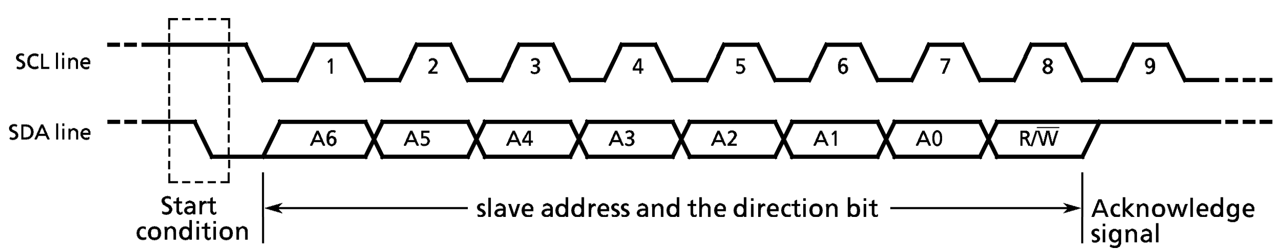


Figure 3.10 (6) Start Condition Generation and Slave Address Generation

When SBICR2 <BB> is “1”, a sequence of generating the stop condition is started by writing “1” to <MST, TRX, PIN> and “0” to <BB>. Do not modify the contents of <MST, TRX, BB, PIN> until the stop condition is generated on a bus.
When a stop condition is generated and the SCL line on the bus is set to low level by another device, a stop condition is generated after releasing the SCL line.

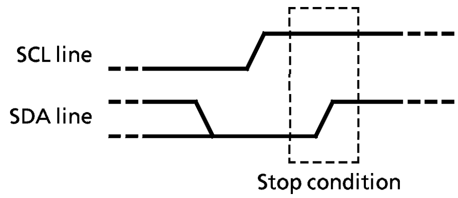


Figure 3.10 (7) Stop Condition Generation

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

(8) Cancel interrupt service request

When the TMP93CS44/S45 is the master mode and transferring a number of clocks set by the SBICR1<BC2 to 0> and the SBICR1<ACK> is complete, a serial bus interface interrupt request (INTS2) is generated.
In the slave mode, the INTS2 is generated when the received slave address is the same as the value set to the I2CAR and an acknowledge signal is output, when a "GENERAL CALL" is received and an acknowledge signal is output, or when transferring / receiving data is complete after the received slave address is the same as the value set to the I2CAR and a "GENERAL CALL" is received.
When the serial bus interface interrupt request occurs, the SBISR<PIN> is cleared to "0". During the time that the PIN is "0", the SCL pin is set to low level.
Either writing or reading data to or from the SBIDBR sets the <PIN> to "1".
The time from the <PIN> being set to "1" until the SCL pin is released takes t_{LOW}.
Although the <PIN> can be set to "1" by the program, the <PIN> is not cleared to "0" when it is written "0".

(9) Serial bus interface operation mode selection

SBICR2 <SBIM1, 0> is used to specify the serial bus interface operation mode. Set <SBIM1, 0> to "10" when used in the I²C bus mode after confirming that input signal via port is high level.
Switch a mode to port after making sure that a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus in the I²C bus mode, a bus arbitration procedure is implemented in order to guarantee the contents of the transferred data.
A data on the SDA line is used for bus arbitration of the I²C bus.
The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master A and Master B output the same data until point "a". After Master A outputs "L" and Master B "H", the SDA line of the bus is wire-AND and the SDA line is driven to the low level by Master A. When the SCL line of the bus is pulled up at point "b", the slave device reads data on the SDA line, that is, data in Master A. A data transmitted from Master B becomes invalid. The state in Master B is called "arbitration lost". B master device which loses arbitration releases the SDA pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

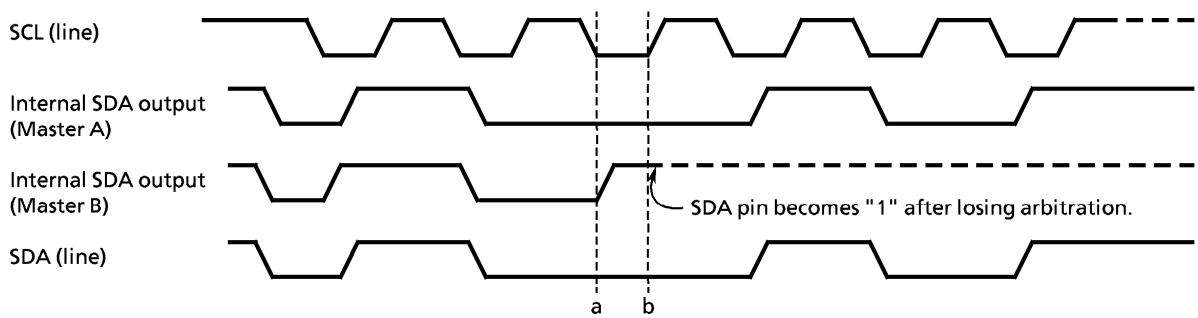


Figure 3.10 (8) Arbitration Lost

The TMP93CS44 / S45 compares levels of the SDA line of the bus with those of the TMP93CS44 / S45 internal SDA output at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and SBISR <AL> is set to “1”.
When <AL> is set to “1”, <MST, TRX> are set to “0” and the mode is switched to the slave receiver mode. The TMP93CS44 / S45 generates the clock pulse until data is transmitted when <AL> is “1”.
<AL> is set to “0” by writing/reading data to/from the SBIDBR or writing data to the SBICR2.

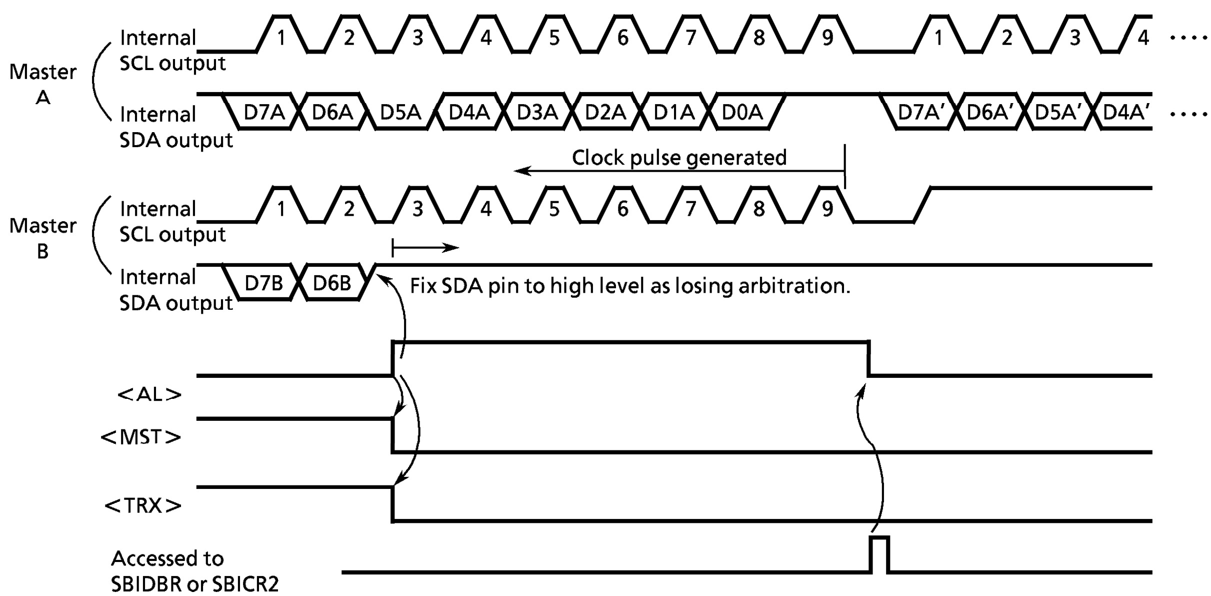


Figure 3.10 (9) Example of when TMP93CS44 / S45 is a Master device

(11) Slave address match detection monitor

SBISR <AAS> is set to “1” in the slave mode, in the address recognition mode (I2CAR <ALS> = 0) when receiving the GENERAL CALL or the slave address with the same value that is set to the I2CAR. When <ALS> is “1”, <AAS> is set to “1” after receiving the first 1-word of data. <AAS> is set to “0” by writing/reading data to/from a data buffer register.

(12) GENERAL CALL detection monitor

SBISR<AD0> is set to “1” in the slave mode, when the GENERAL CALL is received (all 8-bit data are “0” after the start condition). <AD0> is set to “0” when the start or stop condition is detected on a bus.

(13) Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is sent to SBISR <LRB>. In the acknowledge mode, immediately after the INTS2 interrupt request is generated, the acknowledge signal is read by reading the contents of <LRB>.

(14) Software Reset Function

Software reset function is used to initialize the SBI which is rocked by external noise, etc. When SBICR3 <SWRST> is set to "1", the internal reset signal pulse is generated and inputted into the SBI circuit.

All command registers and state registers are initialized to initial values. <SWRST> is automatically set to "0" after the SBI circuit is initialized.

(15) Serial Bus Interface Data Buffer Register (SBIDBR)

The SBIDBR register can read out the receiving data and write the sending data.
After the start condition generated in the master mode, set the slave address and the direction bit in this register.

(16) I^{2}C BUS Address Register (I2CAR)

I2CAR<SA6 to 0> sets the slave address when the TMP93CS44/S45 are operated as the slave devices. Setting I2CAR<ALS> = "0", the slave address output from master device is recognized, and the data format is changed to the addressing format. Setting I2CAR<ALS> = "1", the slave address is not recognized, and the data format is changed to the free data format.

3.10.5 Data Transfer in I2C Bus Mode

(1) Device Initialization

First, set SBICR1 <ACK, SCK2 to 0>. Specify “0” to bits 7 to 5 and 3 in the SBICR1.
Set the slave address <SA6 to 0> and <ALS> to I2CAR (<ALS> = 0 when the addressing format).
Subsequently, set “0” to <MST, TRX, BB>; “1” to <PIN>; “10” to <SBIM1, 0>; and “0” to bits 0 and 1 in the SBICR2. The slave receiver mode is set.

Note 1 : The initialization of the serial bus interface circuit must be complete within the time from all devices which are connected to the bus have initialized to any device does not generate a start condition. If not, there is a possibility that another device starts transferring before an end of the initialization of the serial bus interface circuit. Data can not be received correctly.

(2) Start Condition and Slave Address Generation

Confirm a bus free status (when SIBSR<BB> = 0).
Set the SBICR1 <ACK> to “1” and specify a slave address and a direction bit to be transmitted to the SBIDBR.
When the SBISR<BB> is “0”, the start condition are generated and the slave address and the direction bit which are set to the SBIDBR are output on a bus by wiring “1” to the SBICR2 <MST, TRX, BB> and PIN. An INTS2 interrupt request occurs at the 9th falling edge of the SCL clock cycle, and the <PIN> is cleared to “0”. The SCL pin is pulled down to the low-level while the <PIN> is “0”. When an interrupt request occurs, the <TRX> changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1 : Do not write a slave address to be output to the SBIDBR while data are transferred. If data is written to the SBIDBR, data to been outputting may be destroyed.

Note 2 : Do not start transferring due to another mater from writing a slave address to be output to the SBIDBR to writing a start condition generation command to the SBICR2. The serial bus interface circuit malfunctions because it has not an arbitration function.

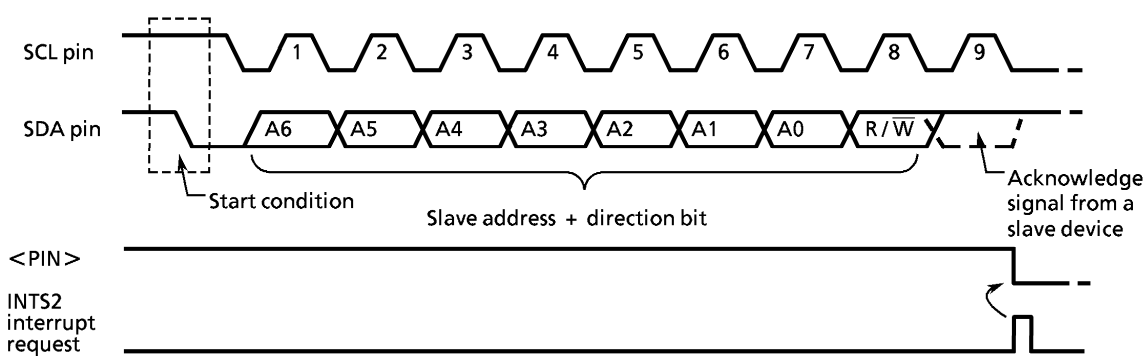


Figure 3.10 (10) Start Condition Generation and Slave Address Transfer

(3) 1-word Data Transfer

Test SBISR <MST> by the INTS2 interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

① When <MST> is “1” (Master mode)

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

When <TRX> is “1” (Transmitter mode)

Check SBISR <LRB>. When <LRB> is “1”, a receiver does not request data. Implement the process to generate the stop condition (described later) and terminate data transfer.
When <LRB> is “0”, the receiver requests new data. When the next transmitted data is 8-bits, write it to the SBIDBR. When the next transmitted data is other than 8 bits, set SBICR1 <BC2 to 0>, set SBICR1 <ACK> to “1”, and write the transmitted data to the SBIDBR. After writing the data, SBISR <PIN> becomes “1”, the serial clock pulse is generated for transferring a new 1-word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, the INTS2 interrupt request occurs. <PIN> becomes “0” and the SCL pin is set to the low level. If the data to be transferred is more than one word in length, repeat the procedures from <LRB> test mentioned above.

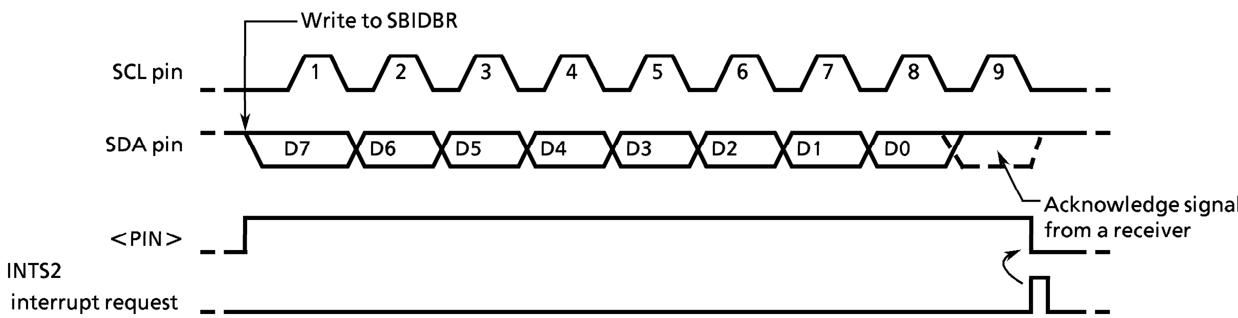


Figure 3.10 (11) Example of when <BC2 to 0> = "000", <ACK> = "1" (Transmitter mode)

When <TRX> is "0" (Receiver mode)

When the next transmitted data is 8 bits, write the transmitted data to the SBIDBR. When the next transmitted data is other than 8 bits, set SBICR1<BC2 to 0> again. Set <ACK> to "1" and read the received data from the SBIDBR to release the SCL line. The read data is undefined immediately after the slave address is set.) After the data is read, <PIN> becomes "1". The TMP93CS44/S45 outputs the serial clock pulse to the SCL pin to transfer new 1-word of data and sets the SDA pin to "0", when the acknowledge signal is set to low level at the final bit.
The INTS2 interrupt request then occurs and <PIN> becomes "0". The SCL pin is set to the low level. The TMP93CS44/S45 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

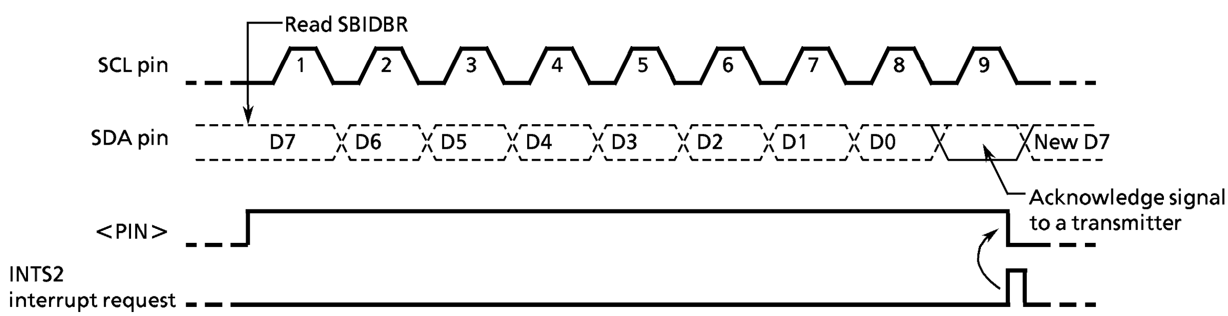


Figure 3.10 (12) Example of when <BC2 to 0> = "000", <ACK> = "1" (Receiver mode)

In order to terminate the transmitting data to the transmitter, set <ACK> to "0" before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set <BC2 to 0> to "001" and read the data. The TMP93CS44/S45 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line of the bus keeps the high level. The transmitter receives the high-level signal as the ACK signal. The receiver indicates to the transmitter that data transfer is complete.
After 1-bit data is received and the interrupt request has occurred, the TMP93CS44/S45 generates the stop condition and terminates data transfer.

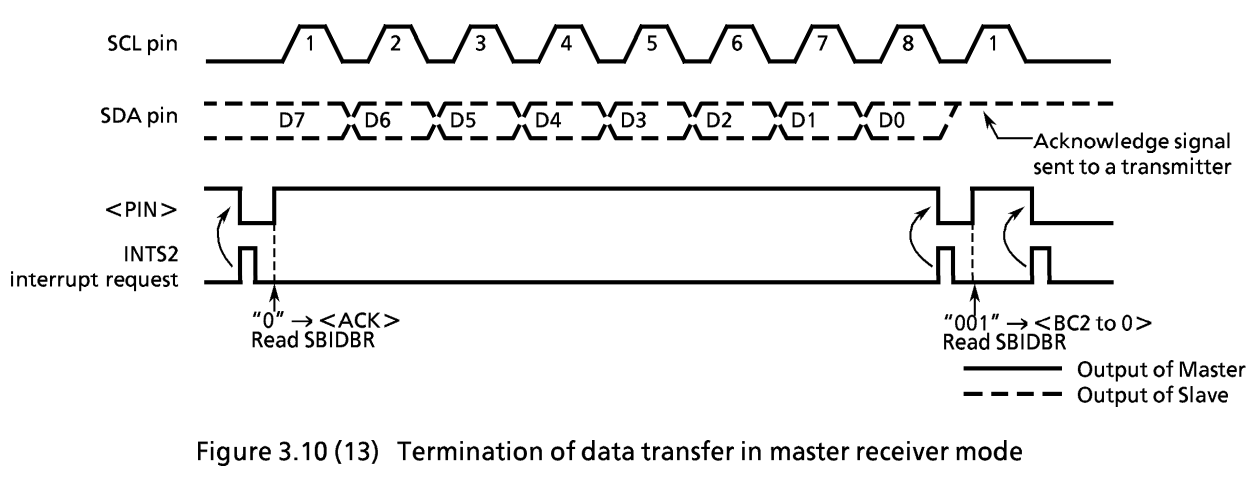


Figure 3.10 (13) Termination of data transfer in master receiver mode

② When < MST> is “0” (Slave mode)

In the slave mode, the TMP93CS44/S45 operates either in normal slave mode or in recovery process after a noise detection.
In the slave mode, an INTS2 interrupt request occurs when the serial bus interface circuit receives a slave address or a “GENERAL CALL” from the master device, or when a “GENERAL CALL” is received and data transfer is complete after matching a received slave address. In the master mode, the TMP93CS44/S45 operates in a slave mode if a noise is detected. An INTS2 interrupt request occurs when word data transfer terminates after a noise detection. When an INTS2 interrupt request occurs, the SBISR <PIN> is reset, and the SCL pin is set to low level. Either reading or writing from or to the SBIDBR or setting the <PIN> to “1” releases the SCL pin after taking t_{LOW} time.
The TMP93CS44/S45 tests the SBISR <AL>, the SBISR <TRX>, the <AAS>, and the <AD0> and implements processes according to conditions listed in the next table.

Table 3.10 (1) Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	0	1	0	In the slave receiver mode, the TMP93CS44/S45 receives a slave address of which the direction bit sent from the master is "1".	Set the number of bits in 1 word to <BC2 to 0> and write transmitted data to the SBIDBR.
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Check <LRB>. If <LRB> is set to "1", set <PIN> to "1" since the receiver does not request next data. Then, clear <TRX> to “0” release the bus. If <LRB> is set to "0", set the number of bits in a word to <BC2 to 0> and write transmitted data to the SBIDBR since the receiver requests next data.
0	1	0	0	The TMP93CS44/S45 loses arbitration when transmitting a slave address or data and terminates transferring word data.	Read the SBIDBR for setting <PIN> to "1" (reading dummy data) or write "1" to <PIN>.
	0	1	1/0	In the slave receiver mode, the TMP93CS44/S45 receives a slave address or GENERAL CALL of which the direction bit sent from the master is "0".	
		0	1/0	In the slave receiver mode, the TMP93CS44/S45 terminates receiving of 1-word data.	Set the number of bits in a word to <BC2 to 0> and read received data from the SBIDBR.

(4) Stop Condition Generation

When SBISR <BB> is "1", a sequence of generating a stop condition is started by writing "1" to SBICR2 <MST, TRX, PIN>, and "0" to <BB>. Do not modify the contents of <MST, TRX, BB, PIN> until the stop condition is generated on the bus. When a SCL line of bus is pulled down by other device the TMP93CS44/S45 generates a stop condition after they release a SCL line.

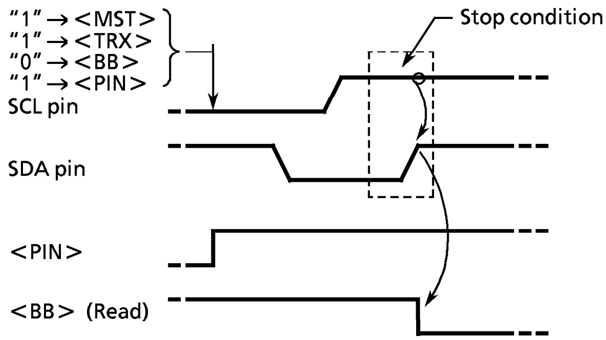


Figure 3.10 (14) Stop Condition Generation

(5) Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart the TMP93CS44/S45.
Clear "0" to the <MST>, <TRX>, and <BB> and set "1" to the <PIN>. The SDA pin retains the high level and the SCL pin is released. Since a stop condition is not generated on the bus, the bus is assumed to be in a busy state from other devices. Test the <BB> until it becomes "0" to check that the SCL pin of the TMP93CS44/S45 is released. Test the <LRB> until it becomes "1" to check that the SCL line of the bus is not set to low level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure (2).
In order to meet setup time when restarting, take at least 4.7 μs of waiting time by software from the time of restarting to confirm that the bus is free until the time to generate the start condition.

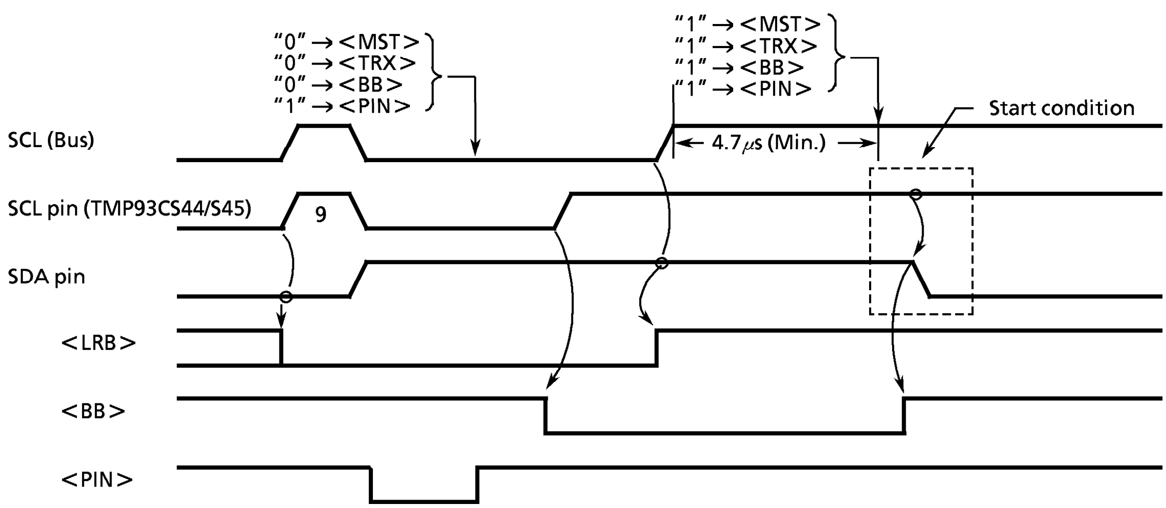


Figure 3.10 (15) Timing diagram when restarting the TMP93CS44 / S45

3.10.6 Clock-synchronous 8-bit SIO Mode Control

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

Serial Bus Interface Control Register 1

SBICR1 (004BH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	W					W		
After reset	0	0	0	0		0	0	0
Function	Indication transfer start / stop 0: Stop 1: start	Continue / Abort transfer 0: Continue 1: Abort	Transfer mode select 00: 8-bit transmit 01: (reserved) 10: 8-bit transmit / receive 11: 8-bit receive			Serial clock selection		

Serial clock selection

000	fc/2^5 (625 kHz)	System clock : fc Clock gear : fc/1 fc = 20MHz (Output on SCK PIN)
001	fc/2^6 (312.5 kHz)	
010	fc/2^7 (156.3 kHz)	
011	fc/2^8 (78.13 kHz)	
100	fc/2^9 (39.06 kHz)	
101	fc/2^{10}(19.53 kHz)	
110	fc/2^{11}(9.77 kHz)	
111	External clock (input from SCK pin)	

Transfer mode select

00	8-bit transmit mode
01	reserved
10	8-bit transmit / receive mode
11	8-bit receive mode

Continue / abort transfer

0	Continue transfer
1	Abort transfer (automatically cleared after abort)

Indicate transfer start / stop

0	Stop
1	Start

Note : Set <SIO> to "0" and <SIOINH> to "1" when setting the transfer mode and the serial clock.

Serial Bus Interface Data Buffer Register

SBIDBR (004CH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Figure 3.10 (16) - 1 Registers for SIO Mode

Serial Bus Interface Control Register 2

SBICR2 (004EH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	SBIM1	SBIM0	–	–
Read/Write					W			
After reset	0	0	0	1	0	0	0	0
Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (reserved)			

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disable)
01	SIO mode
10	I^2C bus mode
11	(reserved)

Note 1 : Switch a mode to port after data transfer is complete.

Note 2 : Switch a mode to SIO mode after confirming that input signals via port are high level.

Serial Bus Interface Status Register

SBISR (004EH)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	SIOF	SEF	–	–
Read/Write					R			
After reset	0	0	0	1	0	0	0	0
Function					Serial tarnsfer operating status monitor	shift operating status monitor		

Sift operating status monitor

0	Shift operation terminated
1	Shift operation in process

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in process

Figure 3.10 (3) - 2 Registers for SIO Mode

(1) Serial Clock

① Clock source

SBICR1<SCK2 to 0> are used to select the following functions.

Internal Clock

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

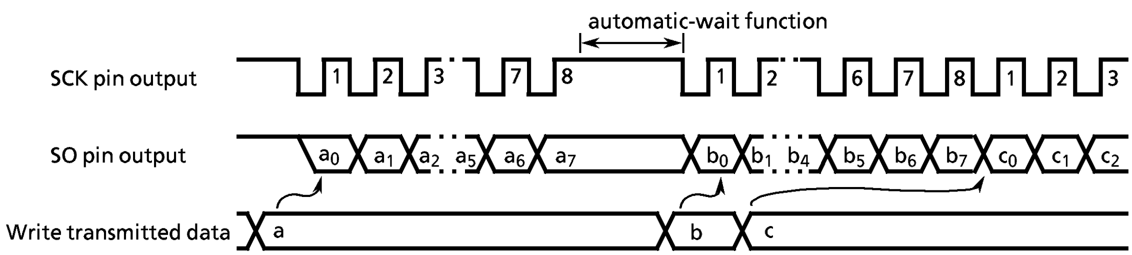
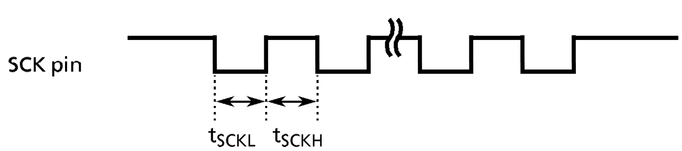


Figure 3.10 (17) Automatic-wait Function

External clock (<SCK 2 to 0> = "111")

An external clock supplied to the SCK pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4 machine cycles is required for both high and low levels in the serial clock. The maximum data transfer frequency is 625kHz (at fc = 20MHz).



t_{SCKL}, t_{SCKH} > 4 tcyc *Note : tcyc = 4/f_{FPH} (in NORMAL mode, IDLE mode)*

Figure 3.10 (18) External Clock

② Shift edge

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

Leading edge shift

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

Trailing edge shift

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

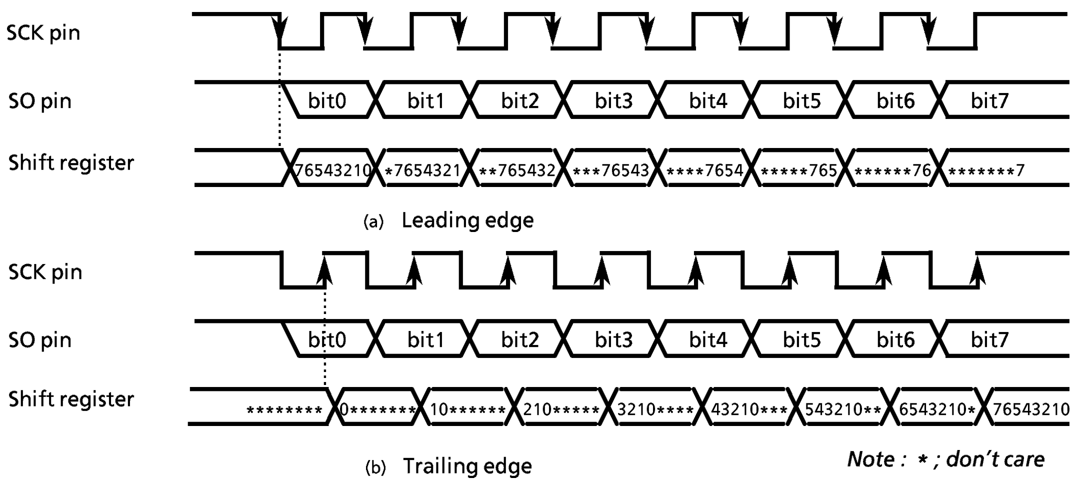


Figure 3.10 (19) Shift Edge

(2) Transfer mode

SBICR1 <SIOM1, 0> are used to select a transmit, receive, or transmit/receive mode.

① 8-bit transmit mode

Set a control register to a transmit mode and write data to the SBIDBR.
After the data is written, set SBICR1 <SIOS> to "1" to start data transfer. The transmitted data is transferred from the SBIDBR to the shift register and output to the SO pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the data is transferred to the shift register, the SBIDBR becomes empty. The INTS2 (buffer empty) interrupt request is generated to request new data.
When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new data is written, automatic-wait function is canceled.
When the external clock is used, data should be written to the SBIDBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBIDBR by the interrupt service program.
When the transmit is started, after SBISR <SIOF> goes "1" the same value as the final bit of the last data is output until the falling edge of the SCK.
Transmitting data is ended by clearing <SIOS> to "0" with the buffer empty interrupt service program or setting SBICR1 <SIOINH> to "1". When <SIOS> is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set <SIOF> to be sensed. <SIOF> is cleared to "0" when transmitting is complete. When <SIOINH> is set to "1", transmitting data stops. <SIOF> turns "0".
When the external clock is used, it is also necessary to clear <SIOS> to "0" before new data is shifted; otherwise, dummy data is transmitted and operation ends.

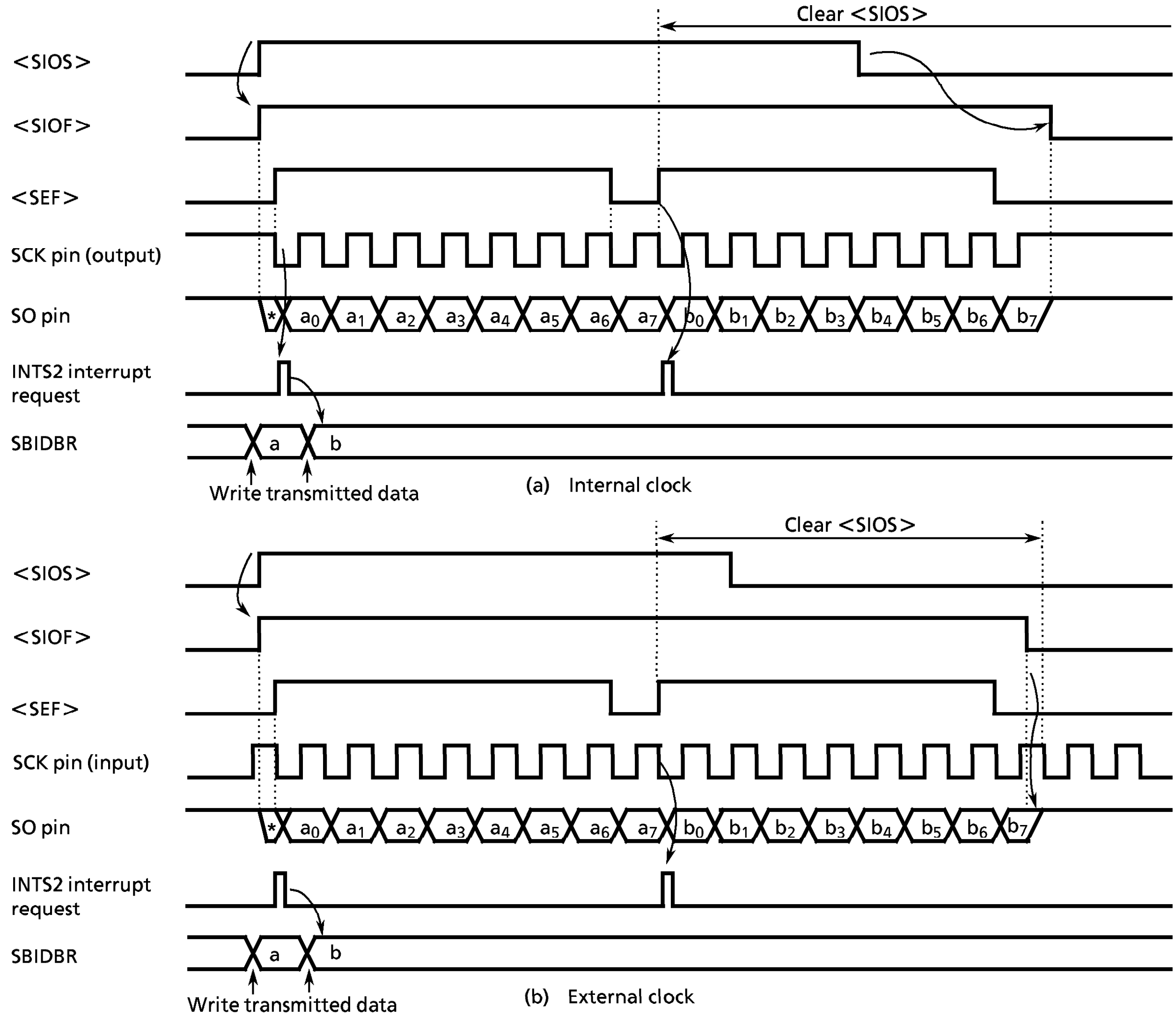


Figure 3.10 (20) Transfer Mode

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

```
STEST1 :  BIT   SEF, (SBISR)          ;  If <SEF> = 1   then   loop
          JR    NZ, STEST1
STEST2 :  BIT   2, (P3)               ;  If SCK = 0   then   loop
          JR    Z , STEST2
          LD    (SBICR1), 00000111B   ;  <SIOS> ← 0
```

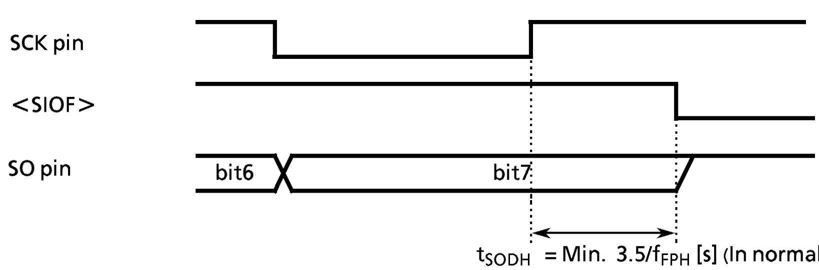


Figure 3.10 (21) Transmitted Data Hold Time at end of transmit

② 8-bit Receive Mode

Set the control register to a receive mode and SBICR1 <SIOS> to “1” for switching to the receive mode. Data is received from the SI pin to the shift register in synchronous with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBIDBR. The INTS2 (buffer full) interrupt request is generated to request to read the received data. The data is then read from the SBIDBR by the interrupt service program.
When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated until the received data is read from the SBIDBR.
When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read from the SBIDBR before next serial clock input. If the received data is not read, further data to be received is canceled. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read.
Receiving data is ended by clearing <SIOS> to “0” with the buffer full interrupt service program or setting SBICR1 <SIOINH> to “1”. When <SIOS> is cleared, received data is transferred to the SBIDBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set SBISR <SIOF> to be sensed. <SIOF> is cleared to “0” when receiving is complete. After confirming that receiving has ended, the last data is read. When <SIOINH> is set to “1”, receiving data stops. <SIOF> turns “0” (the received data becomes invalid, therefore no need to read it).

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

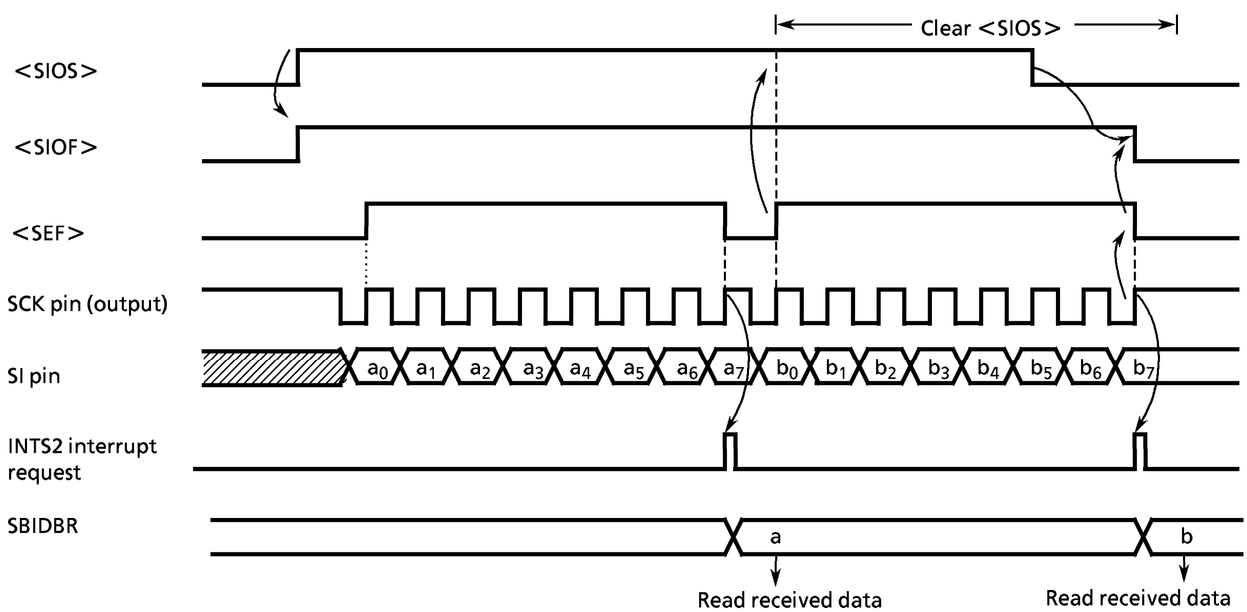


Figure 3.10 (22) Receive Mode (Example : Internal clock)

③ 8-bit Transmit/Receive Mode

Set a control register to a transmit/receive mode and write data to the SBIDBR. After the data is written, set SBICR1<SIOS> to "1" to start transmitting/receiving. When transmitting, the data is output from the SO pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI pin on the trailing edges of the serial clock. The 8-bit data is transferred from the shift register to the SBIDBR, and the INTS2 interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

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

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

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

Transmitting/receiving data is ended by clearing <SIOS> to "0" by the INTS2 interrupt service program or setting SBICR1 <SIOINH> to "1". When <SIOS> is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted/received by the program, set SBISR <SIOF> to be sensed. <SIOF> becomes "0" after transmitting/receiving is complete. When <SIOINH> is set, transmitting/receiving data stops. <SIOF> turns "0".

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

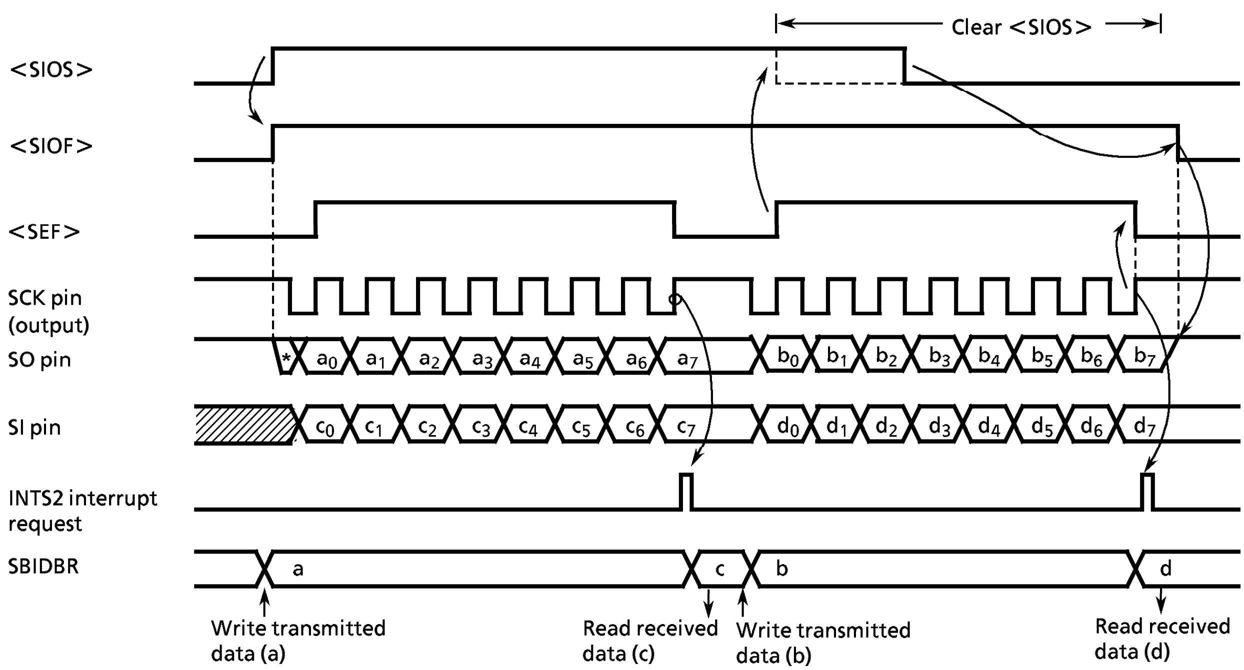


Figure 3.10 (23) Transmit/Receive Mode (Example : Internal clock)

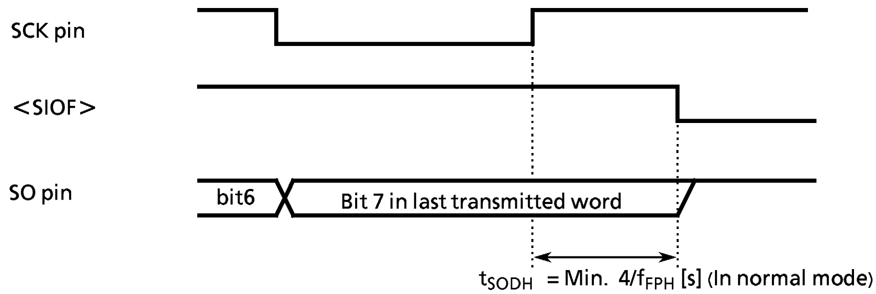


Figure 3.10 (24) Transmitted Data Hold Time at end of transmit/receive

3.11 Analog/Digital Converter

TMP93CS44 / S45 incorporate a high-speed, high-precision 10-bit analog/digital converter (A/D converter) with 8-channel analog input.
Figure 3.11 (1) is a block diagram of the A/D converter. The 8-channel analog input pins (AN0 to AN7) are also used as input-only Port5 and can be also used as input ports.

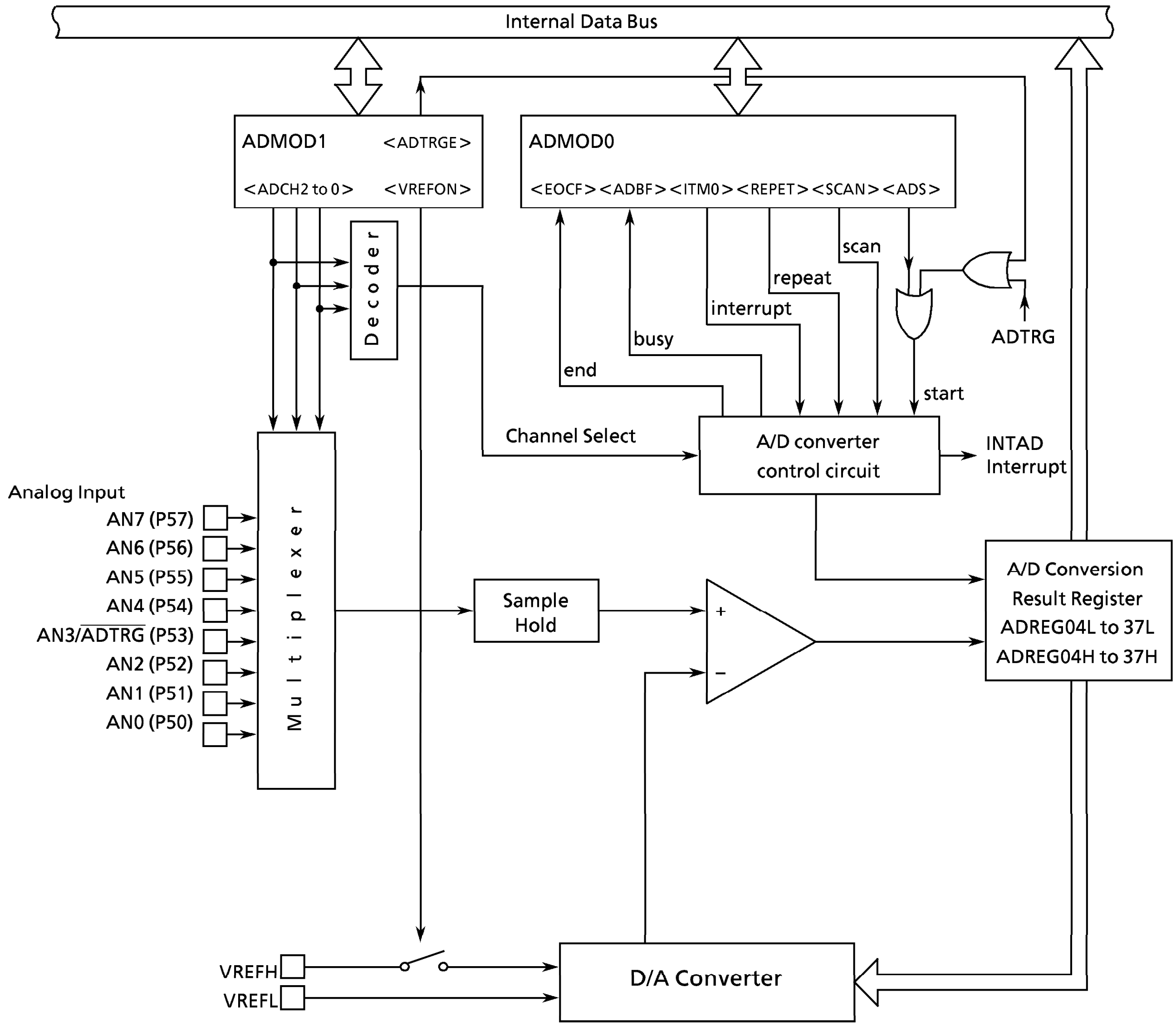


Figure 3.11 (1) Block Diagram of A/D Converter

Note 1 : When the power supply current is reduced in IDLE2, IDLE1, STOP mode, there is possible to set a standby enabling the internal comparator due to a timing. Stop operation of A/D converter before execution of "HALT" instruction.

Note 2 : In regard to the lowest operation frequency
The operation of A/D converter is guaranteed with clock of $f_{FPH} \geqq 4$ MHz (used fc clock). Is not guaranteed with fs clock.

3.11.1 Analog/Digital Converter Registers

A/D converter is controlled by two A/D mode control registers (ADMOD0 and ADMOD1). A/D conversion result is stored in eight A/D conversion result registers (ADREG04H/L, ADREG15H/L, ADREG26H/L, ADREG37H/L).

A/D Mode Control Register 0

ADMOD0 (005EH)

	7	6	5	4	3	2	1	0
Bit Symbol	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
Read/Write	R		R / W					
After reset	0	0	0	0	0	0	0	0
Function	A/D conversion end flag 0 : Conversion in progress 1 : Conversion end	A/D conversion busy flag 1 : Conversion is idle 1 : Conversion in progress	(Note) Always write "0".	(Note) Always write "0".	Specifies interrupts for fixed channel /repeat-conversion mode. 0 : Every conversion 1 : Every four conversions	Repeat mode specification 0 : Single conversion mode 1 : Repeat conversion mode	Scan mode specification 0 : Fixed-channel mode 1 : Channel scan mode	A/D conversion start 0 : Don't care 1 : Start conversion (Note)

A/D conversion start

0	Don't care
1	Start A/D conversion

(Note) Always read as "0".

A/D scan mode specification

0	A/D conversion fixed-channel mode
1	A/D conversion channel scan mode

A/D repeat mode specification

0	A/D single conversion mode
1	A/D repeat conversion mode

A/D conversion interrupt specification for fixed-channel/repeat-conversion mode

	Fixed channel/repeat conversion mode <SCAN> = "0", <REPET> = "1"
0	Generates interrupt every conversion.
1	Generates interrupt every four conversions.

A/D conversion busy flag

0	A/D conversion is idle
1	A/D conversion in progress

A/D conversion end flag

0	A/D conversion in progress
1	A/D conversion end

Figure 3.11 (2) - 1 Register for A/D Converter

A/D Mode Control Register 1

ADMOD1 (005FH)

	7	6	5	4	3	2	1	0
Bit Symbol	VREFON				ADTREG	ADCH2	ADCH1	ADCH0
Read/Write	R/W				R/W			
After reset	1				0	0	0	0
Function	String resistor 0: OFF 1: ON				External trigger start control 0 : Disable 1 : Enable	Analog input channel selection		

Analog input channel selection

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

Conversion start control by external trigger ($\overline{\text{ADTRG}}$ pin input)

0	Disable
1	Enable

Analog reference voltage control

0	OFF
1	ON

(Note) Set the <VREFON> bit to "1" before starting conversion (before writing "1" to ADMOD0<ADS>).

(Note) As the AN3 and the $\overline{\text{ADTRG}}$ are the same pin, <ADCH2 to 0> = "011" can't be set when <ADTRGE> is set to 1 and $\overline{\text{ADTRG}}$ is used.

Figure 3.11 (2) - 2 Register for A/D Converter

A/D Conversion Result Register 0/4 Low

ADREG04L (0060H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00						ADR0RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of A/D conversion result.							Conversion result stored flag 1: Exist result

A/D Conversion Result Register 0/4 High

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	Stores upper 8 bits of A/D conversion result.							

A/D Conversion Result Register 1/5 Low

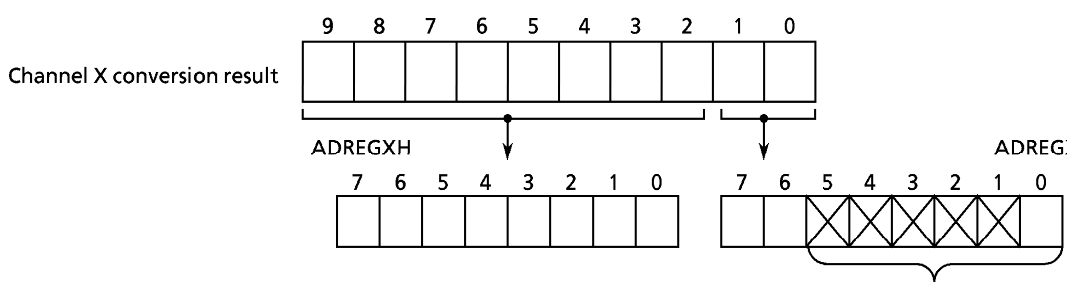
ADREG15L (0062H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10						ADR1RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of A/D conversion result.							Conversion result stored flag 1: Exist result

A/D Conversion Result Register 1/5 High

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	Stores upper 8 bits of A/D conversion result.							



- Bits 5 to 1 are always read as "1".
- Bit 0 is conversion result stored flag bit <ADRXRF>. <ADRXRF> is set to "1" when the A/D conversion result is stored. Reading either the ADREGXH or the ADREGXL registers clears <ADRXRF> to "0".

Figure 3.11 (2) - 3 Registers for A/D Converter

A/D Conversion Result Register 2/6 Low

ADREG26L (0064H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						ADR2RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of A/D conversion result.							Conversion result stored flag 1: Exist result

A/D Conversion Result Register 2/6 High

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	Stores upper 8 bits of A/D conversion result.							

A/D Conversion Result Register 3/7 Low

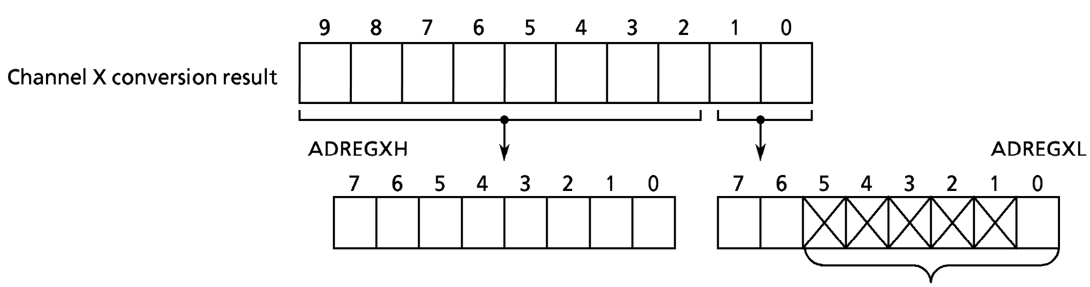
ADREG37L (0066H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30						ADR3RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of A/D conversion result.							Conversion result stored flag 1: Exist result

A/D Conversion Result Register 3/7 High

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	Stores upper 8 bits of A/D conversion result.							



- Bits 5 to 1 are always read as “1”.
- Bit 0 is conversion result stored flag bit <ADRXRF>. <ADRXRF> is set to “1” when the A/D conversion result is stored. Reading either the ADREGXH or the ADREGXL registers clears <ADRXRF> to “0”.

Figure 3.11 (2) - 4 Registers for A/D Converter

3.11.2 Operation

(1) Analog Reference Voltage

High analog reference voltage is applied to the VREFH pin, and low analog reference voltage is applied to the VREFL pin. The voltage between VREFH and VREFL is divided into 1024 increments using a string resistor. A/D conversion is based on comparing the analog input voltage with these reference voltage increments.
To turn the switch between VREFH and VREFL off, write "0" to the ADMOD1<VREFON> bit.
To start A/D conversion when the switch is off, first write "1" to <VREFON>. After that, wait at 3 μs long enough to get the stabilized oscillation, write "1" to ADMOD0<ADS>.

(2) Selecting Analog Input Channels

The procedure for selecting analog input channels depends on the operating mode of the A/D converter.

- When analog input channel is used to fix (ADMOD0<SCAN> = "0")
  To set ADMOD1<ADCH2 to 0>, selecting one channel from analog input pins AN0 to AN7.
- When analog input channel is used to scan (ADMOD0<SCAN> = "1")
  To set ADMOD1<ADCH2 to 0>, selecting one channel from 8 scan mode.

Table 3.11 (1) shows the analog input channel selection each operating mode.

A reset initializes ADMOD0<SCAN> to "0" and ADMOD1<ADCH2 to 0> to "000", selecting pin AN0 for the A/D converter input.
The pins not used as analog input channels can be used as general-purpose input ports (P5).

Table 3.11 (1) Analog Input Channel Selection

<ADCH2 to 0>	Fixed Channel <SCAN> = 0	Channel Scan <SCAN> = 1
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

(3) Starting A/D Conversion

A/D conversion starts when ADMOD0<ADS> to "1", or ADMOD1<ADTRGE> is set to "1" and the falling edge is input through $\overline{\text{ADTRG}}$ pin.
When A/D conversion starts, A/D conversion busy flag ADMOD0<ADBF>is set to "1", indicating A/D conversion is in progress.
Writing "1" to <ADS> while conversion is in progress restarts the conversion. Check the conversion result stored flag ADREGxxL<ADRxRF> to determine whether the A/D conversion data are valid at this time.
Inputting the falling edge to the $\overline{\text{ADTRG}}$ pin while conversion is in progress is invalid.

(4) A/D Conversion Modes and Completion Interrupt

Follow the four A/D conversion modes are supported.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

A/D conversion mode can selected by setting A/D Mode Control Register ADMOD0 <REPET, SCAN>. When A/D conversion end, A/D conversion completion interrupt INTAD request occurs. And the ADMOD0<EOCF> flag is set to "1" to indicate that A/D conversion has completed.

① Fixed Channel Single Conversion Mode

Fixed channel single conversion mode can be specified by setting ADMOD0 <REPET, SCAN> to "00". In this mode, conversion of the specified single channel is executed once only. After conversion is completed, ADMOD0<EOCF> is set to "1", ADMOD0<ADBF> is cleared to "0" and occurs INTAD interrupt request.

② Channel Scan Single Conversion Mode

Channel scan single conversion mode can be specified by setting ADMOD0 <REPET, SCAN> to "01". In this mode, conversion of the specified channel are executed once only. After conversion is completed, ADMOD0<EOCF> is set to "1", ADMOD0<ADBF> is cleared to "0" and occurs INTAD interrupt request.

③ Fixed Channel Repeat Conversion Mode

Fixed channel repeat conversion mode can be specified by setting ADMOD0 <REPET, SCAN> to "10". In this mode, conversion of the specified single channel is executed repeatedly. After conversion is completed, ADMOD0 <EOCF> is set to "1", ADMOD0<ADBF> remains "1" not changed to "0". The timing of INTAD interrupt request can selected by setting of ADMOD0<ITM0>.
When <ITM0> is set to "0" , interrupt request occurs after every conversion.
When <ITM0> is set to "1" , interrupt request occurs after every fourth conversion.

④ Channel Scan Repeat Conversion Mode

Channel scan repeat conversion mode can be specified by setting ADMOD0 <REPET, SCAN> to “11”. In this mode, specified channels are converted repeatedly. After every scan convert completion, ADMOD0<EOCF> is set to “1” and INTAD interrupt request occurs. ADMOD0<ADBF> remains “1”, not changed to “0”.

To stop the repeat conversion mode (③ and ④ modes), write “0” to ADMOD0<REPET>. After the current conversion is completed, repeat conversion mode is terminated, and ADMOD0<ADBF> is cleared to “0” .
If the device enters the IDLE2, IDLE1 or STOP modes during A/D conversion, the conversion halt immediately. After the halt mode is released, A/D conversion restarts from the beginning in repeat conversion mode (③ and ④), it does not restart in single conversion mode (① and ②).

Table 3.11 (2) shows the relations between A/D conversion modes and interrupt request.

Table 3.11 (2) Relation between A/D Conversion Modes and Interrupt Request

<table>
<tr><th rowspan="2">Mode</th><th rowspan="2">Interrupt Request Timing</th><th colspan="3">ADMOD0</th></tr>
<tr><th><ITM0></th><th><REPET></th><th><SCAN></th></tr>
<tr><td>Fixed channel single conversion mode</td><td>After conversion</td><td>X</td><td>0</td><td>0</td></tr>
<tr><td>Channel scan single conversion mode</td><td>After conversion</td><td>X</td><td>0</td><td>1</td></tr>
<tr><td>Fixed channel repeat conversion mode (Every conversion)</td><td>After every conversion</td><td>0</td><td rowspan="2">1</td><td rowspan="2">0</td></tr>
<tr><td>Fixed channel repeat conversion mode (Every fourth conversion)</td><td>After every fourth conversion</td><td>1</td></tr>
<tr><td>Channel scan repeat conversion mode</td><td>After every scan conversion</td><td>X</td><td>1</td><td>1</td></tr>
</table>

X : Don’t Care

(5) A/D Conversion Time

140 states (14 μs at fc = 20 MHz) are required for A/D conversion of one channel.

(6) Storing and Reading the A/D Conversion Result

A/D conversion results are stored in A/D conversion result registers high/low (ADREG04H/L to ADREG37H/L). These registers are read only.
In fixed channel repeat conversion mode, A/D conversion results are stored in order from ADREG04H/L to ADREG37H/L. Except in this mode, A/D conversion results for channel AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7 are stored severally ADREG04H/L, ADREG15H/L, ADREG26H/L, ADREG37H/L.
Table 3.11 (3) shows correspondence between analog input channels and A/D conversion result registers.

Table 3.11 (3) Correspondence Between Analog Input Channels and A/D Conversion Result Registers

Analog Input Channel (port 5)	A/D Conversion Result Registers	
	Conversion Modes Except Right	Fixed Channel Repeat Conversion Mode (Every fourth conversion)
AN0	ADREG04H/L	ADREG04H/L ↓ ADREG15H/L ↓ ADREG26H/L ↓ ADREG37H/L
AN1	ADREG15H/L	
AN2	ADREG26H/L	
AN3	ADREG37H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

A/D conversion result registers bit "0" is A/D conversion result stored flag <ADRxRF>. The flag shows that whether those registers are read or not. When A/D conversion results are stored in those registers (ADREGxH or ADREGxL), this flag is set to "1". When each register is read, this flag is cleared to "0", and A/D conversion end flag ADMOD0<EOCF> is also cleared to "0".

Setting example :

① This example converts the analog input voltage at the AN3 pin. The INTAD interrupt routine writes the result to memory address 0800H.

Main routine setting:

	7 6 5 4 3 2 1 0	
INTE0AD ←	0 0 0 0 1 1 0 0	Enables INTAD and sets level 4.
ADMOD1 ←	1 X X X 0 0 1 1	Sets analog input channel to AN3.
ADMOD0 ←	X X 0 0 0 0 0 1	Starts A/D conversion in fixed channel single conversion mode.

Example of interrupt routine processing:

WA	← ADREG37	Reads ADREG37L and ADREG37H values and writes them to WA (16 bits).
WA	>> 6	Shifts right WA six times and zero-fills the upper bits.
(0800H)	← WA	Writes contents of WA to memory address 0800H.

② This example repeatedly converts the analog input voltages at pins AN0 to AN2, using channel scan repeat conversion mode.

INTE0AD ←	0 0 0 0 1 0 0 0	Disables INTAD.
ADMOD1 ←	1 X X X 0 0 1 0	Sets AN0 to AN2 as analog input channels.
ADMOD0 ←	X X 0 0 0 1 1 1	Starts A/D conversion in channel scan repeat conversion mode.

(Note) X : Don’t care - : no change

3.12 Watchdog Timer (Runaway Detecting Timer)

TMP93CS44 / S45 contain a watchdog timer of Runaway detecting.
The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.
Connecting the watchdog timer output to the reset pin internally forces a reset.
This watchdog timer consists of 7-stage and 15-stage binary counters.
These binary counters are also used as a warm-up timer for the internal oscillator stabilization. This is used for releasing the STOP and before changing system clock.

3.12.1 Configuration

Figure 3.12 (1) shows the block diagram of the watchdog timer (WDT).

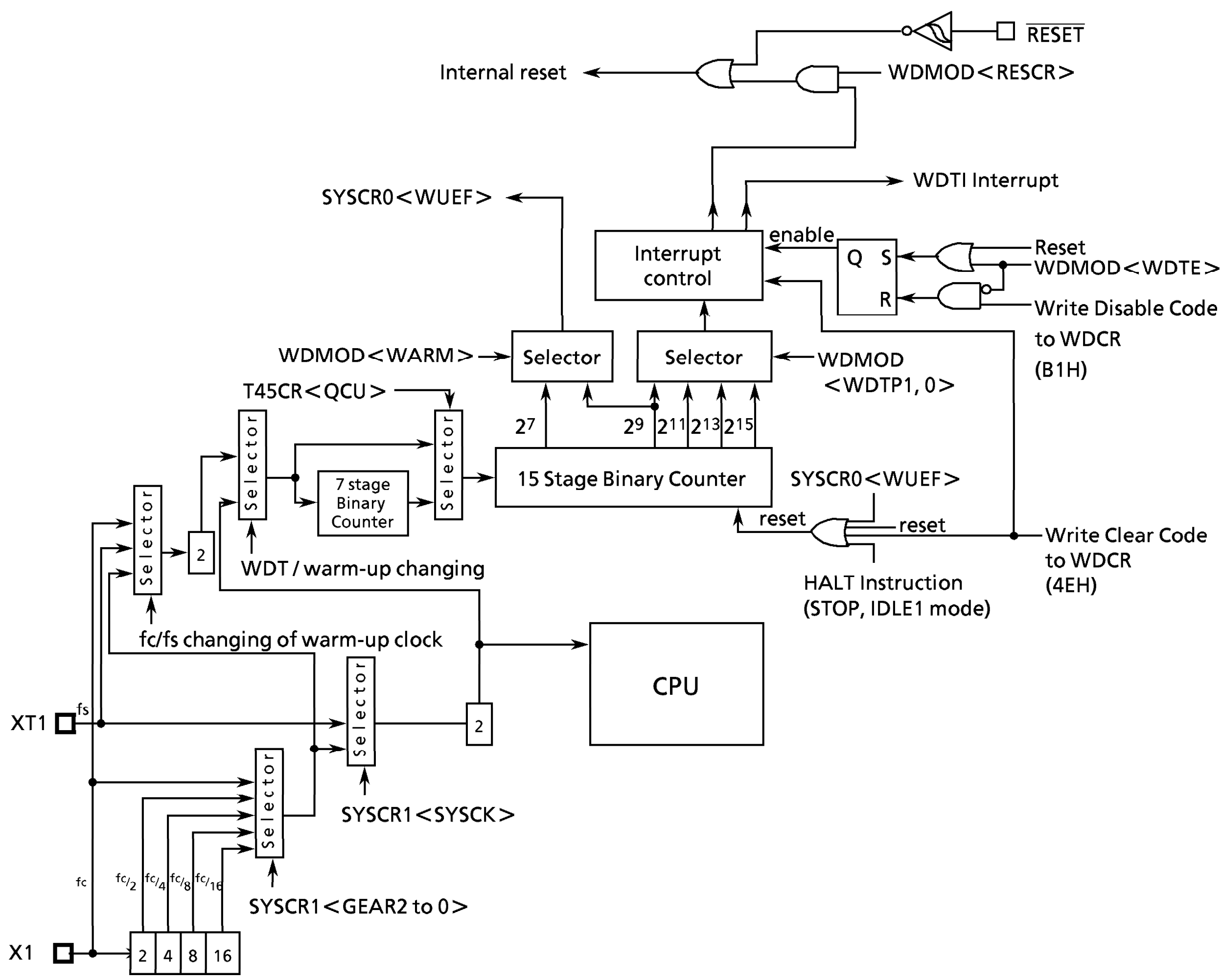


Figure 3.12 (1) Block Diagram of Watchdog Timer / Warm-up Timer

The watchdog timer consists of 7-stage and 15-stage binary counters which use System clock (f_{SYS}) as the input clock. The 15-stage binary counter has $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$ output. Selecting one of the outputs with the WDMOD<WDTP1, 0> register generates a watchdog interrupt and outputs watchdog timer out when an overflow occurs. The binary counter for the watchdog timer should be cleared to "0" with runaway detecting result software (instruction) before an interrupt occurs.

(Example)

```
    LDW     (WDMOD), B100H     ;   disable
    LD      (WDCR), 4EH        ;   write clear code
    SET     7, (WDMOD)         ;   enable again
```

The runaway detecting result can also be connected to the reset pin internally. In this case, the watchdog timer resets itself.

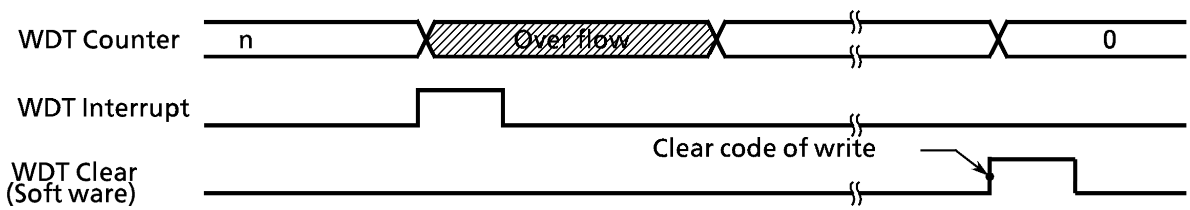


Figure 3.12 (2) Normal Mode

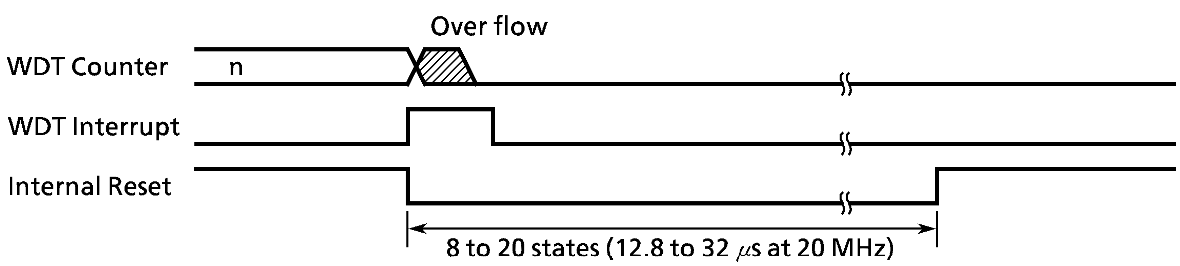


Figure 3.12 (3) Reset Mode

For warm-up counter, 2^7 and 2^9 output of 15-stage binary counter can be selected using WDMOD<WARM> register. When a stable-external oscillator is used, shorter warm-up time is available using T45CR<QCU> register. When <QCU> = 1, counting value 2^7 is selected.
When the watchdog timer is in operation, this shorter warm-up time function cannot be available. This function can be available by setting <QCU> = 0.

3.12.2 Control Registers

Watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog Timer Mode Register (WDMOD)

① Setting the detecting time of watchdog timer <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD<WDTP1, 0> = 00 when reset.
The defecting time of WDT is shown Table 3.12. (1).

② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD<WDTE> is initialized to "1" enable the watchdog timer.
To disable, it is necessary to set this bit to "0" and write the disable code (B1H) in the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.
However, it is possible to return from the disable state to enable state by merely setting <WDTE> to "1".

③ Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with $\overline{\text{RESET}}$ terminal, internally. Since WDMOD<RESCR>is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog Timer Control Register (WDCR)

This register is used to disable and clear of binary counter the watchdog timer function.

- Disable control

By writting the disable code (B1H) in this WDCR register after clearing WDMOD<WDTE> to "0", the watchdog timer can be disabled.

```
WDMOD ← 0 - - - - - X X    Clear WDMOD<WDTE>to "0".
WDCR  ← 1 0 1 1 0 0 0 1    Write the disable code (B1H).
```

- Enable control

Set WDMOD<WDTE>to "1".

- Watchdog timer clear control

The binary counter can be cleared and resume counting by writing clear code (4EH) into the WDCR register.

```
WDCR  ← 0 1 0 0 1 1 1 0    Write the clear code (4EH).
```

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: Disable 1: Enable	Select detecting time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$ See Table 3.12. (1)		Warm-up time	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

「3.3 Standby function」

Watchdog timer Enable / Disable control

0	Disable
1	Enable

Watchdog timer out control

0	Don't care
1	Connects WDT out to a reset

Select the Warm-up time at fc = 20 MHz, fs = 32.768 kHz

System clock selection <SYSCK>	Gear value <GEAR2 to 0>	Warm-up time		
		T45CR<QCU> = 0		T45CR<QCU> = 1
		<WARM = 0>	<WARM = 1>	<WARM> = X
1 (fs)	XXX (Don't care)	0.500 s	2.000 s	3.9 ms
0 (fc)	000 (fc)	0.8192 ms	3.277 ms	6.4 μs
	001 (fc/2)	1.638 ms	6.554 ms	12.8 μs
	010 (fc/4)	3.277 ms	13.107 ms	25.6 μs
	011 (fc/8)	6.554 ms	26.214 ms	51.2 μs
	100 (fc/16)	13.107 ms	52.429 ms	102.4 μs

Note : When the watchdog timer is in opelation, T45CR<QCU> set to "0".

Figure 3.12 (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	Don't care

(Note) When the watchdog timer is in operation, T45CR<QCU> is set to "0".

Figure 3.12 (5) Watchdog Timer Control Register

Table 3.12 (1) Watchdog Timer Detecting Time

at fc = 20 MHz, fs = 32.768 kHz

System clock selection <SYSCK>	Gear value <GEAR2 to 0>	Watchdog Timer Detecting Time WDMOD<WDTP1, 0> 00	01	10	11
1 (fs)	XXX (Don't care)	2.000 s	8.000 s	32.000 s	128.000 s
0 (fc)	000 (fc)	3.277 ms	13.107 ms	52.429 ms	209.715 ms
	001 ($fc/2$)	6.554 ms	26.214 ms	104.858 ms	419.430 ms
	010 ($fc/4$)	13.107 ms	52.429 ms	209.715 ms	838.861 ms
	011 ($fc/8$)	26.214 ms	104.858 ms	419.430 ms	1.678 s
	100 ($fc/16$)	52.429 ms	209.715 ms	838.861 ms	3.355 s

3.12.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1, 0>. The watchdog timer must be zero-cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (runaway) due to the INTWD Interrupt and it is possible to return to normal operation by an anti-mulfunction program. 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.
However, the function can be disabled when entering the RUN, IDLE2 mode.

Example : ① Clear the binary counter

```
WDCR  ← 0 1 0 0 1 1 1 0        Write clear code (4EH).
```

② Set the watchdog timer detecting time to 2^{17} / f_{SYS}

```
WDMOD ← 1 0 1 - - - X X
```

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 HALT 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 HALT mode.
```

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

4. ELECTRICAL CHARACTERISTICS

"X" used in an expression shows a cycle of clock f_{FPH} selected by SYSCR1<SYSCK>. If a clock gear or a low speed oscillator is selected, a value of "X" is different. The value as an example is calculated at fc, gear = 1/fc (SYSCR1<SYSCK, GEAR 2 to 0> = "0000").

4.1 Absolute Maximum Ratings (TMP93CS44F, TMP93CS45F)

Symbol	Parameter	Rating	Unit
Vcc	Power Supply Voltage	– 0.5 to 6.5	V
V_{IN}	Input Voltage	– 0.5 to Vcc + 0.5	V
I_{OL1}	Output current (Per 1 pin) P7	20	mA
I_{OL2}	Output current (Per 1 pin) except P7	2	mA
ΣI_{OL1}	Output Current (P7 total)	80	mA
ΣI_{OL}	Output Current (total)	120	mA
ΣI_{OH}	Output Current (total)	– 80	mA
P_D	Power Dissipation (Ta = 85 ℃)	350	mW
T_{SOLDER}	Soldering Temperature (10 s)	260	℃
T_{STG}	Storage Temperature	– 65 to 150	℃
T_{OPR}	Operating Temperature	– 40 to 85	℃

4.2 DC Characteristics (1/2)

Ta = – 40 to 85°C

Symbol	Parameter		Min.	Typ. (Note1)	Max.	Unit	Condition
Vcc	Power Supply Voltage		4.5		5.5	V	fc = 4 to 20 MHz; fs = 30 to 34 kHz
			2.7				fc = 4 to 12.5 MHz
V_{IL}	Input Low Voltage	AD0 to 15	– 0.3		0.8		Vcc ≧ 4.5V
					0.6		Vcc < 4.5V
V_{IL1}		Port2 to 7 (except P35)			0.3Vcc	V	Vcc = 2.7 to 5.5V
V_{IL2}		$\overline{RESET}$, $\overline{NMI}$, INT0			0.25Vcc		
V_{IL3}		$\overline{EA}$, AM8/$\overline{16}$			0.3		
V_{IL4}		X1			0.2Vcc		
V_{IH}	Input High Voltage	AD0 to 15	2.2		Vcc + 0.3		Vcc≧4.5V
			2.0				Vcc<4.5V
V_{IH1}		Port2 to 7 (except P35)	0.7Vcc				Vcc = 2.7 to 5.5V
V_{IH2}		$\overline{RESET}$, $\overline{NMI}$, INT0	0.75Vcc				
V_{IH3}		$\overline{EA}$, AM8/$\overline{16}$	Vcc – 0.3				
V_{IH4}		X1	0.8Vcc				
V_{OL}	Output Low Voltage				0.45		I_{OL} = 1.6 mA (Vcc = 2.7 to 5.5V)
I_{OL7}	Output Low current (P7)		16			mA	V_{OL} = 1.0V (Vcc = 5V ± 10 %)
			7				V_{OL} = 1.0V (Vcc = 3V ± 10 %)
V_{OH1}	Output High Voltage		2.4			V	I_{OH} = – 400 μA (Vcc = 3V ± 10%)
V_{OH2}			4.2			V	I_{OH} = – 400 μA (Vcc = 5V ± 10%)

Note 1: Typical values are for Ta = 25 °C and V_{CC} = 5 V unless otherwise noted.

4.2 DC Characteristics (2/2)

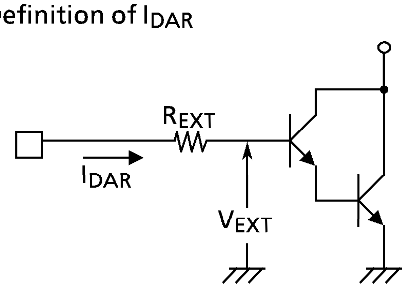
Symbol	Parameter	Min	Typ.(Note1)	Max	Unit	Condition
I_{DAR} (Note2)	Darlington Drive Current (8 Output Pins max.)	−1.0		−3.5	mA	V_{EXT} = 1.5 V R_{EXT} = 1.1 kΩ (Vcc = 5 V ± 10 % only)
I_{LI}	Input Leakage Current		0.02	± 5	μA	0.0 ≦ V_{IN} ≦ V_{CC}
I_{LO}	Output Leakage Current		0.05	± 10	μA	0.2 ≦ V_{IN} ≦ V_{CC} − 0.2
V_{STOP}	Power Down Voltage (at STOP, RAM Back up)	2.0		6.0	V	V_{IL2} = 0.2 Vcc, V_{IH2} = 0.8 Vcc
R_{PU}	Pull Up Resistance	45		130	kΩ	Vcc = 5.5 V
		50		160		Vcc = 4.5 V
		70		280		Vcc = 3.3 V
		90		400		Vcc = 2.7 V
C_{IO}	Pin Capacitance			10	pF	fc = 1 MHz
V_{TH}	Schmitt Width $\overline{RESET}$, $\overline{NMI}$, INT0	0.4	1.0		V	
Icc	NORMAL (Note3)		19	25	mA	Vcc = 5 V ± 10 % fc = 20 MHz
	RUN		17	25		
	IDLE2		12	17		
	IDLE1		3.5	5		
	NORMAL (Note3)		6.5	10	mA	Vcc = 3 V ± 10 % fc = 12.5 MHz (Typ. : Vcc = 3.0 V)
	RUN		5.0	9		
	IDLE2		4.5	6.5		
	IDLE1		0.8	1.5		
	SLOW (Note3)		20	45	μA	Vcc = 3 V ± 10 % fs = 32.768 kHz (Typ. : Vcc = 3.0 V)
	RUN		16	40		
	IDLE2		15	25		
	IDLE1		5	15		
	STOP		0.2	10	μA	Ta ≦ 50 °C (Vcc = 2.7 V to 5.5 V)
				20		Ta ≦ 70 °C
				50		Ta ≦ 85 °C

Note 1: Typical values are for Ta = 25 °C and V_{CC} = 5 V unless otherwise noted.

Note 2: I_{DAR} is guranteed for total of up to 8 ports.

Note 3: I_{CC} measurement conditions (NORMAL, SLOW).
Only CPU is operational; output pins are open and input pins are fixed.

(Reference) Definition of I_{DAR}



4.3 AC Characteristics

(1) Vcc = 5 V ± 10 %

No.	Symbol	Parameter	Variable		16 MHz		20 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{OSC}	Osc. Period (= x)	50	31250	62.5		50		ns
2	t_{CLK}	CLK width	2x – 40		85		60		ns
3	t_{AK}	A0 to 23 Valid→CLK Hold	0.5x – 20		11		5		ns
4	t_{KA}	CLK Valid→A0 to 23 Hold	1.5x – 70		24		5		ns
5	t_{AL}	A0 to 15 Valid→ALE fall	0.5x – 15		16		10		ns
6	t_{LA}	ALE fall→A0 to 15 Hold	0.5x – 20		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 – 25		6		0		ns
9	t_{CL}	$\overline{RD}$/$\overline{WR}$ rise→ALE rise	0.5x – 20		11		5		ns
10	t_{ACL}	A0 to 15 Valid→$\overline{RD}$/$\overline{WR}$ fall	x – 25		38		25		ns
11	t_{ACH}	A0 to 23 Valid→$\overline{RD}$/$\overline{WR}$ fall	1.5x – 50		44		25		ns
12	t_{CA}	$\overline{RD}$/$\overline{WR}$ rise→A0 to 23 Hold	0.5x – 25		6		0		ns
13	t_{ADL}	A0 to 15 Valid→D0 to 15 input		3.0x – 55		133		95	ns
14	t_{ADH}	A0 to 23 Valid→D0 to 15 input		3.5x – 65		154		110	ns
15	t_{RD}	$\overline{RD}$fall →D0 to 15 input		2.0x – 60		65		40	ns
16	t_{RR}	$\overline{RD}$ Low pulse width	2.0x – 40		85		60		ns
17	t_{HR}	$\overline{RD}$rise→D0 to 15 Hold	0		0		0		ns
18	t_{RAE}	$\overline{RD}$rise→A0 to 15output	x – 15		48		35		ns
19	t_{WW}	$\overline{WR}$ Low pulse width	2.0x – 40		85		60		ns
20	t_{DW}	D0 to 15 Valid→$\overline{WR}$ rise	2.0x – 55		70		45		ns
21	t_{WD}	$\overline{WR}$ rise →D0 to 15 Hold	0.5x – 15		16		10		ns
22	t_{AWH}	A0 to 23 Valid→$\overline{WAIT}$ input (1 WAIT + n mode)		3.5x – 90		129		85	ns
23	t_{AWL}	A0 to 15 Valid→$\overline{WAIT}$ input (1 WAIT + n mode)		3.0x – 80		108		70	ns
24	t_{CW}	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ Hold (1 WAIT + n mode)	2.0x + 0		125		100		ns
25	t_{APH}	A0 to 23 Valid→PORT input		2.5x – 120		36		5	ns
26	t_{APH2}	A0 to 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t_{CP}	$\overline{WR}$ rise→PORT Valid		200		200		200	ns

AC Measuring Conditions

- Output Level : High 2.2 V / Low 0.8 V , CL = 50 pF
  (However CL = 100 pF for AD0 to AD15, A0 to A23, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, CLK)
- Input Level : High 2.4 V / Low 0.45 V (AD0 to AD15)
  High 0.8 Vcc / Low 0.2 Vcc (Except for AD0 to AD15)

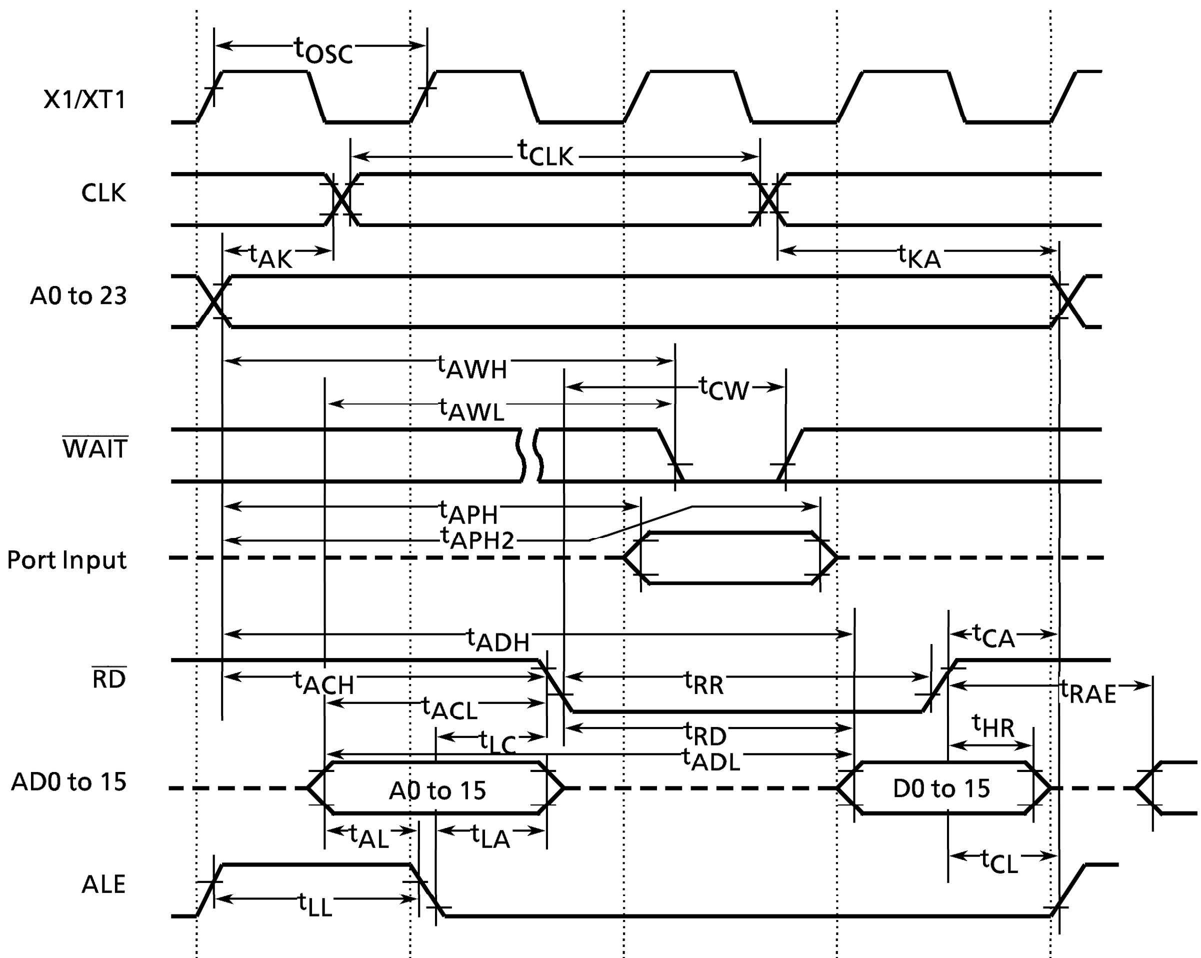
(2) Vcc = 3 V ± 10 %

No.	Symbol	Parameter	Variable		12.5 MHz		Unit
			Min	Max	Min	Max	
1	t_{OSC}	Osc. Period (= x)	80	31250	80		ns
2	t_{CLK}	CLK width	2x – 40		120		ns
3	t_{AK}	A0 to 23 Valid→ CLK Hold	0.5x – 30		10		ns
4	t_{KA}	CLK Valid→ A0 to 23 Hold	1.5x – 80		40		ns
5	t_{AL}	A0 to 15 Valid→ ALE fall	0.5x – 35		5		ns
6	t_{LA}	ALE fall→ A0 to 15 Hold	0.5x – 35		5		ns
7	t_{LL}	ALE High width	x – 60		20		ns
8	t_{LC}	ALE fall→ $\overline{RD}/\overline{WR}$ fall	0.5x – 35		5		ns
9	t_{CL}	$\overline{RD}/\overline{WR}$ rise→ ALE rise	0.5x – 40		0		ns
10	t_{ACL}	A0 to 15 Valid→ $\overline{RD}/\overline{WR}$ fall	x – 50		30		ns
11	t_{ACH}	A0 to 23 Valid→ $\overline{RD}/\overline{WR}$ fall	1.5x – 50		70		ns
12	t_{CA}	$\overline{RD}/\overline{WR}$ rise→ A0 to 23 Hold	0.5x – 40		0		ns
13	t_{ADL}	A0 to 15 Valid→ D0 to 15 input		3.0x – 110		130	ns
14	t_{ADH}	A0 to 23 Valid→ D0 to 15 input		3.5x – 125		155	ns
15	t_{RD}	$\overline{RD}$fall → D0 to 15 input		2.0x – 115		45	ns
16	t_{RR}	$\overline{RD}$ Low pulse width	2.0x – 40		120		ns
17	t_{HR}	$\overline{RD}$rise→ D0 to 15 Hold	0		0		ns
18	t_{RAE}	$\overline{RD}$rise→ A0 to 15output	x – 25		55		ns
19	t_{WW}	$\overline{WR}$ Low pulse width	2.0x – 40		120		ns
20	t_{DW}	D0 to 15 Valid→ $\overline{WR}$rise	2.0x – 120		40		ns
21	t_{WD}	$\overline{WR}$ rise →D0 to 15 Hold	0.5x – 40		0		ns
22	t_{AWH}	A0 to 23 Valid→ $\overline{WAIT}$ input (1 WAIT + n mode)		3.5x – 130		150	ns
23	t_{AWL}	A0 to 15 Valid→ $\overline{WAIT}$ input (1 WAIT + n mode)		3.0x – 100		140	ns
24	t_{CW}	$\overline{RD}/\overline{WR}$ fall→$\overline{WAIT}$ Hold (1 WAIT + n mode)	2.0x + 0		160		ns
25	t_{APH}	A0 to 23 Valid→ PORT input		2.5x – 120		80	ns
26	t_{APH2}	A0 to 23 Valid→ PORT Hold	2.5x + 50		250		ns
27	t_{CP}	$\overline{WR}$ rise→ PORT Valid		200		200	ns

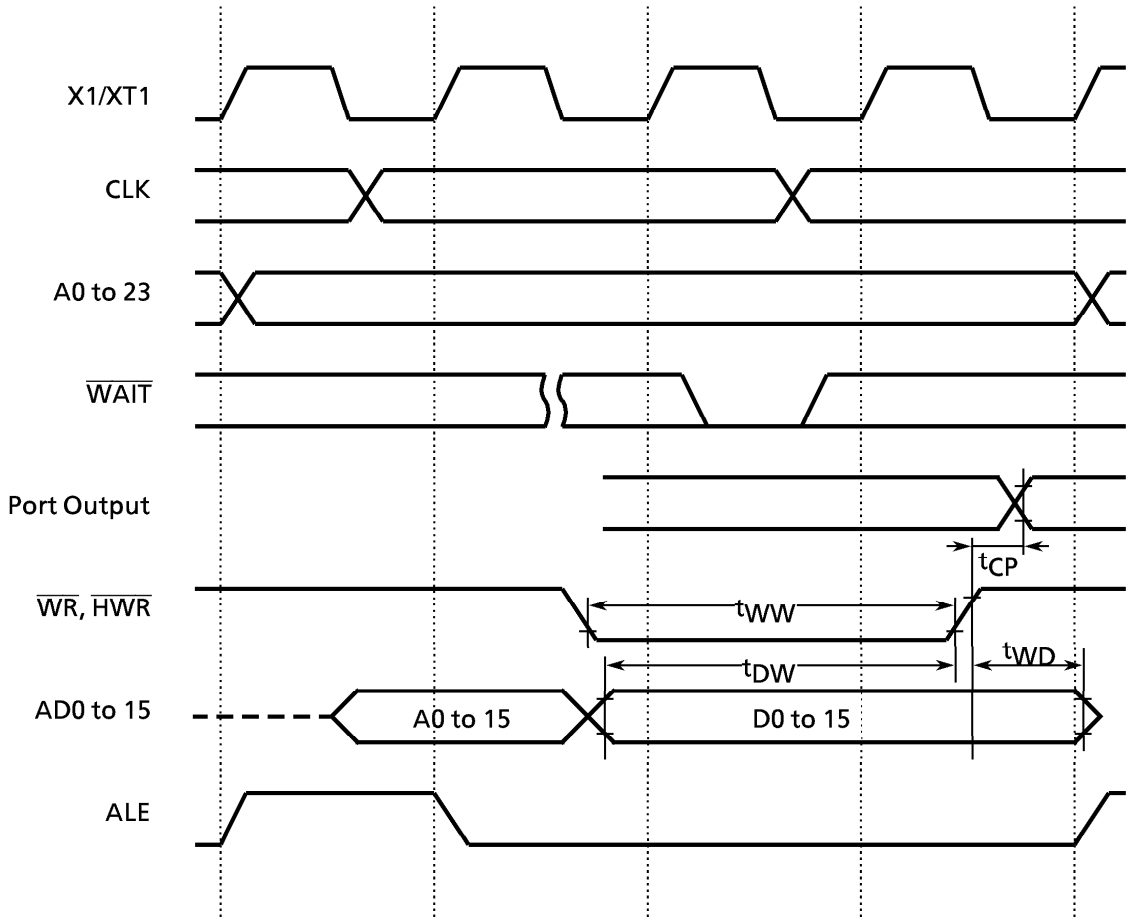
AC Measuring Conditions

- Output Level : High $0.7 \times V_{CC}$ / Low $0.3 \times V_{CC}$, CL = 50 pF
- Input Level : High $0.9 \times V_{CC}$ / Low $0.1 \times V_{CC}$

(1) Read Cycle


X1/XT1
CLK
A0 to 23
WAIT
Port Input
RD
AD0 to 15
ALE
tOSC
tCLK
tAK
tKA
tAWH
tAWL
tCW
tAPH
tAPH2
tADH
tCA
tACH
tRR
tRAE
tACL
tRD
tHR
tLC
tADL
A0 to 15
D0 to 15
tAL
tLA
tCL
tLL

(2) Write Cycle



4.4 Serial Channel Timing

(1) I/O Interface Mode

① SCLK Input Mode

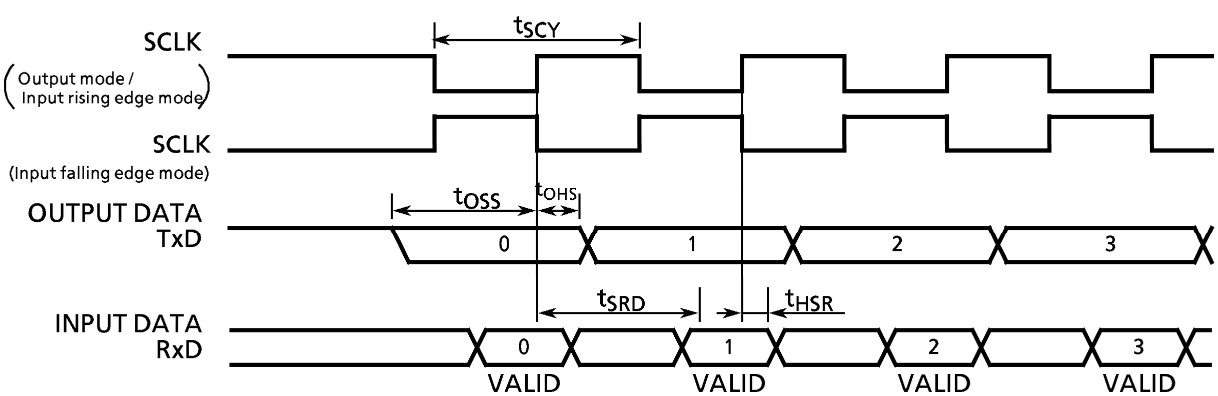
Symbol	Parameter	Variable		32.768 MHz (Note)		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	16X		488 μs		1.28		0.8		μs
t_{OSS}	Output Data → falling edge of SCLK	t_{SCY}/2 – 5X – 50		91.5 μs		190		100		ns
t_{OHS}	SCLK rising / falling edge → Output Data hold	5X – 100		152 μs		300		150		ns
t_{HSR}	SCLK rising / falling edge → Input Data hold	0		0		0		0		ns
t_{SRD}	SCLK rising / falling edge → effective data input		t_{SCY} – 5X – 100		336 μs		780		450	ns

Note 1: When fs is used as system clock or fs divided by 4 is used as input clock to prescaler.
Note 2: SCLK rising/falling timing; SCLK rising in the rising mode of SCLK, SCLK falling in the falling mode of SCLK.

② SCLK Output Mode

Symbol	Parameter	Variable		32.768 MHz (Note)		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (Programmable)	16X	8192X	488 μs	250 ms	1.28	865.36	0.8	409.6	μs
t_{OSS}	Output Data → SCLK rising edge	t_{SCY} – 2X – 150		427 μs		970		550		ns
t_{OHS}	SCLK rising edge → Output Data hold	2X – 80		60 μs		80		20		ns
t_{HSR}	SCLK rising edge → Input Data hold	0		0		0		0		ns
t_{SRD}	SCLK rising edge → effective Data input		t_{SCY} – 2X – 150		428 μs		970		550	ns

Note: When fs is used as system clock or fs divided by 4 is used as input clock to prescaler.



(2) UART Mode (SCLK0, 1 are external input)

Symbol	Parameter	Variable		32.768 kHz (Note)		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	4x + 20		122 μs		340		220		ns
t_{SCYL}	SCLK Low level pulse width	2x + 5		6 μs		165		105		ns
t_{SCYH}	SCLK High level pulse width	2x + 5		6 μs		165		105		ns

Note: When fs is used as system clock or fs divided by 4 is used as input clock to prescaler.

4.5 A/D Conversion Characteristics

$AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$

Symbol	Parameter	Power Supply	Min	Typ	Max	Unit
VREFH	Analog reference voltage (+)	V_{CC} = 5V ± 10 %	V_{CC} – 0.2 V	V_{CC}	V_{CC}	V
		V_{CC} = 3V ± 10 %	V_{CC} – 0.2 V	V_{CC}	V_{CC}	
VREFL	Analog reference voltage (–)	V_{CC} = 5V ± 10 %	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
		V_{CC} = 3V ± 10 %	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
V_{AIN}	Analog input voltage range		VREFL		VREFH	
I_{REF} (V_{REFL} = 0 V)	Analog current for analog reference voltage <VREFON> = 1	V_{CC} = 5V ± 10 %		0.5	1.5	mA
		V_{CC} = 3V ± 10 %		0.3	0.9	
	<VREFON> = 0	V_{CC} = 2.7 to 5.5V		0.02	5.0	μA
–	Error (except quantization errors)	V_{CC} = 5V ± 10 %		± 1.0	± 3.0	LSB
		V_{CC} = 3V ± 10 %		± 1.0	± 5.0	

Note 1: $1LSB = (VREFH - VREFL) / 2^{10}$ [V]
Note 2: The operation above is guaranteed for $f_{FPH} \geqq 4$ MHz.
Note 3: The value I_{CC} includes the current which flows through the AVCC pin.

4.6 Event Counter Input Clock (external input clock : TI0, TI4, TI5, TI6, TI7)

Symbol	Parameter	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock Cycle	8X + 100		740		500		ns
t_{VCKL}	Low level clock Pulse width	4X + 40		360		240		ns
t_{VCKH}	High level clock Pulse width	4X + 40		360		240		ns

4.7 Interrupt and Capture Operation

(1) $\overline{\text{NMI}}$, INT0 Interrupts

Symbol	Parameter	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{\text{NMI}}$, INT0 Low level Pulse width	4X		320		200		ns
t_{INTAH}	$\overline{\text{NMI}}$, INT0 High level Pulse width	4X		320		200		ns

(2) INT1, 4 to 7 Interrupts and Capture

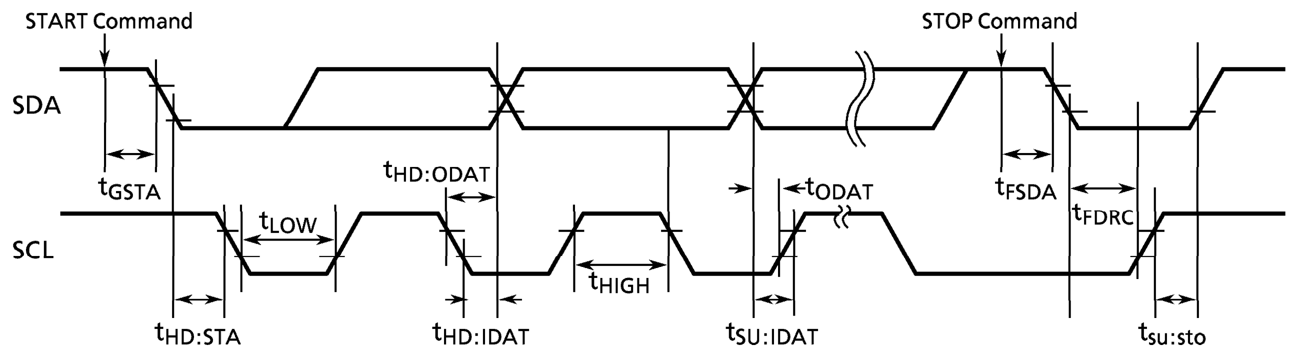
Symbol	Parameter	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTBL}	INT1, INT4 to INT7 Low level Pulse width	4X + 100		420		300		ns
t_{INTBH}	INT1, INT4 to INT7 High level Pulse width	4X + 100		420		300		ns

4.8 Serial Bus Interface Timing

(1) I^2C Bus Mode

Symbol	Parameter	Variable			Unit
		Min	Typ	Max	
t_{GSTA}	START command → SDA fall	3X			s
$t_{HD:STA}$	Hold time START condition	2^nX			s
t_{LOW}	SCL Low level pulse width	2^nX			s
t_{HIGH}	SCL High level pulse width	$2^nX + 12X$			s
$t_{HD:IDAT}$	Data hold time (input)	0			ns
$t_{SU:IDAT}$	Data set-up time (input)	250			ns
$t_{HD:ODAT}$	Data hold time (output)	7X		11X	s
t_{ODAT}	Data output → SCL Rising edge		$2^nX - t_{HD:ODAT}$		s
t_{FSDA}	STOP command → SDA fall	3X			s
t_{FDRC}	SDA Falling edge → SCL Rising edge	2^nX			s
$t_{SU:STO}$	Set-up time STOP condition	$2^nX + 16X$			s

Note : "n" value is set by SBICR1 <SCK2 to 0>



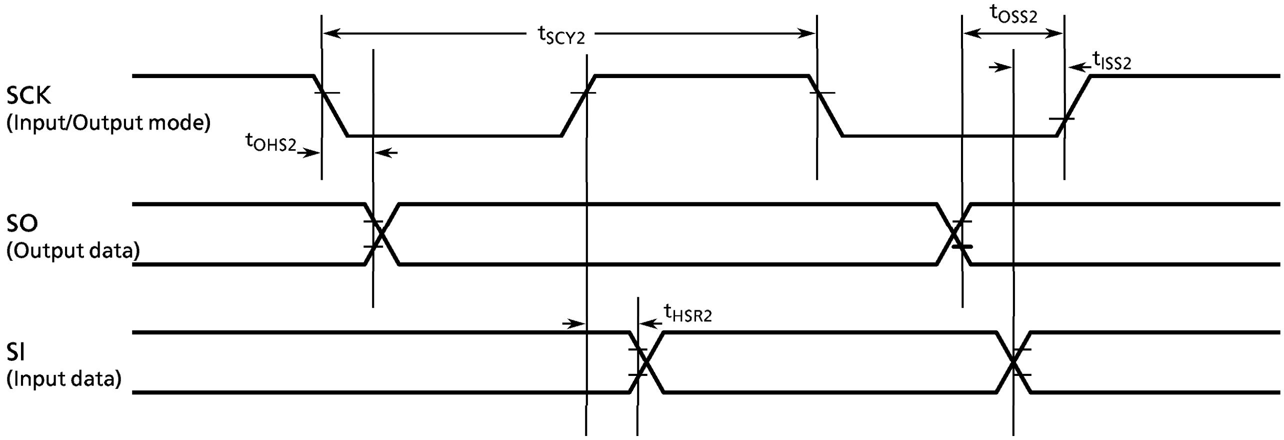
(2) Clocked-synchronous 8-bit SIO Mode

① SCK Input Mode

Symbol	Parameter	Variable		Unit
		Min	Max	
t_{SCY2}	SCK cycle	2^5X		s
t_{OHS2}	SCK falling edge→ Output data hold	6X		s
t_{OSS2}	Output data → SCK rising edge	$t_{SCY2} - 6X$		s
t_{HSR2}	SCK rising edge→ Input data hold	6X		ns
t_{ISS2}	Input data→ SCK rising edge	0		ns

② SCK Output Mode

Symbol	Parameter	Variable		Unit
		Min	Max	
t_{SCY2}	SCK cycle	2^5X	$2^{11}X$	s
t_{OHS2}	SCK falling edge → Output data hold	2X		s
t_{OSS2}	Output data→ SCK rising edge	$t_{SCY2} - 2X$		s
t_{HSR2}	SCK rising edge→ Input data hold	2X		s
t_{ISS2}	Input data→ SCK rising edge	0		ns



5. TABLE OF SPECIAL FUNCTION REGISTERS

(SFR ; Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 128-bytes addresses from 000000H to 00007FH.

(1) I/O Port

(2) I/O Port Control

(3) Clock Control

(4) Interrupt Control

(5) Chip Select / Wait Control

(6) Timer Control

(7) Serial Channel Control

(8) Serial Bus Interface Control

(9) Watchdog Timer Control

(10) A/D Converter Control

Configuration of the table

Symbol	Name	Address	7	6			1	0	
									→ bit Symbol
									→ Read / Write
									→ Initial value after reset
									→ Remarks

Note : "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.

(Example) When setting only bit 0 of register P0CR, "SET 0, (0002H)" cannot be used. The LD (transfer) instruction must be used to write all eight bits.

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	(Reserved)	41H	TREG6H	61H	ADREG04H
2H	P0CR	22H	TREG0	42H	TREG7L	62H	ADREG15L
3H	(Reserved)	23H	TREG1	43H	TREG7H	63H	ADREG15H
4H	P1CR	24H	T10MOD	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	T32MOD	48H	T5MOD	68H	WAITC0
9H	P2FC	29H	TRDC	49H	T5FFCR	69H	WAITC1
AH	P3CR	2AH		4AH	(Reserved)	6AH	WAITC2
BH	P3FC	2BH		4BH	SBICR1	6BH	(Reserved)
CH	P4	2CH	(Reserved)	4CH	SBIDBR	6CH	(Reserved)
DH	P5	2DH		4DH	I2CAR	6DH	CKOCR
EH	P4CR	2EH		4EH	SBICR2	6EH	SYSCR0
FH	(Reserved)	2FH		4FH	SBICR3	6FH	SYSCR1
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H	(Reserved)	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	INTET32
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H		37H	CAP2H	57H	BR1CR	77H	INTEO54
18H		38H	T4MOD	58H	ODE	78H	INTES0
19H		39H	T4FFCR	59H		79H	INTES1
1AH		3AH	T45CR	5AH	(Reserved)	7AH	INTE1S2
1BH	(Reserved)	3BH		5BH		7BH	IIMC
1CH		3CH		5CH	WDMOD	7CH	DMA0V
1DH		3DH	(Reserved)	5DH	WDCR	7DH	DMA1V
1EH		3EH		5EH	ADMOD0	7EH	DMA2V
1FH		3FH		5FH	ADMOD1	7FH	DMA3V

Note : Do not access addresses which do not have register names allocated.

(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							
			Underfined							
			Input mode							
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input mode							
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
		(Prohibit	R/W							
		RMW*)	1	1	1	1	1	1	1	1
			Input mode							
P3	PORT3	07H			P35	P34	P33	P32	P31	P30(note1)
		(Prohibit			R/W					
		RMW*)			1	1	1	1	1	1
			Input mode						Output mode	
P4	PORT4	0CH	P47	P46	P45	P44	P43	P42	P41	P40
		(Prohibit	R/W							
		RMW*)	1	1	1	1	1	1	1	1
			Input mode							
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
		(Prohibit	R/W							
		RMW*)	1	1	1	1	1	1	1	1
			Output mode		Input mode					
P7	PORT7	13H	P77	P76	P75	P74	P73	P72	P71	P70
		(Prohibit	R/W							
		RMW*)	1	1	1	1	1	1	1	1
			Input mode							

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 : Port66, 67 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)

Prohibit RMW* ; Read-modify-write is prohibited when controlling the PU resistor.

(2) I/O Port Control

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)			P35C	P34C	P33C	P32C		
					W					
					0	0	0	0		
					0 : IN 1 : OUT					
P3FC	PORT3 Function	0BH (Prohibit RMW)		P32M		P34F	P33F	P32F	P31F	P30F
				W		W				
				0		0	0	0	0	0
				0 : $\overline{\text{HWR}}$ 1 : SCK		0 : PORT 1 : SCL/SI	0: PORT 1: SDA/SO	0 : PORT 1 : P32M	0 : PORT 1 : $\overline{\text{WR}}$	0 : PORT 1 : $\overline{\text{RD}}$
P4CR	PORT4 Control	0EH (Prohibit RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0 : IN 1 : OUT							
P4FC	PORT4 Function	10H (Prohibit RMW)	P47F			P44F			P41F	
			W			W			W	
			0			0			0	
			0 : PORT 1 : TO6			0 : PORT 1 : TO4			0 : PORT 1 : TO3	
P6CR	PORT6 Control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			1	1	0	0	0	0	0	0
			0 : IN 1 : OUT							
P7CR	PORT7 Control	15H (Prohibit RMW)	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0	0	0	0	0	0	0	0
			0 : IN 1 : OUT							
P6FC	PORT6 Function	16H (Prohibit RMW)			P65F		P63F	P62F		P60F
					W		W	W		W
					0		0	0		0
					0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0		0 : PORT 1 : TxD0

Note : With the TMP93CS45, which requires an external ROM, PORT0 functions as AD0 to AD7; PORT1, AD8 to AD15 or A8 to A15; P30, the $\overline{\text{RD}}$ signal; P31, the $\overline{\text{WR}}$ signal, regardless of the values set in P0CR, P1CR, P1FC, P30F and P31F.

(3) Clock Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CKOCR	Clock Output Control Register	006DH	–	–					ALEEN	CLKEN
			R/W						R/W	
			0	0						
			(Note) Always write "0"						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	Warm-up Timer 0 write : don't care 1 write : start timer 0 read : end warm-up 1 read : not end warm-up	select prescaler clock 00: f_{FPH} 01: 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 :

TMP93CS44 : "0" (High impedance output)

TMP93CS45 : "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 clear 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".

(4) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT 0 / AD Enable Register	0070H (Prohibit PMW)	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	INT 4 / 5 Enable Register	0071H (Prohibit PMW)	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	INT 6 / 7 Enable Register	0072H (Prohibit PMW)	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	INTT 1 / 0 Enable Register	0073H (Prohibit PMW)	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
INTET32	INTT 3 / 2 Enable Register	0074H (Prohibit PMW)	INTT3 (Timer3)				INTT2 (Timer2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET54	INTT 5 / 4 Enable Register	0075H (Prohibit PMW)	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	INTT 7 / 6 Enable Register	0076H (Prohibit PMW)	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
INTEO54	INTTO 5 / 4 Enable Register	0077H (Prohibit PMW)	INTTO5				INTTO4			
			ITO5C	ITO5M2	ITO5M1	ITO5M0	ITO4C	ITO4M2	ITO4M1	ITO4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES0	INT RX0 / TX0 Enable Register	0078H (Prohibit PMW)	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	INT RX1 / TX1 Enable Register	0079H (Prohibit PMW)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE1S2	INT1 / INTS2 Enable Register	007AH (Prohibit PMW)	INT1				INTS2			
			I1C	I1M2	I1M1	I1M0	IS2C	IS2M2	IS2M1	IS2M0
			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 - - - - -

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 Control	7BH (Prohibit RMW)			–			I0IE	I0LE	NMIREE
					W			W	W	W
								0	0	0
					(Note) Always write "0"			1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Operation even at $\overline{NMI}$ rising edge

(5) Bus-width / Wait Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
WAITC0	Block 0 WAIT control register	68H (Prohibit RMW)				B0BUS	B0W1	B0W0	B0C1	B0C0
						W				
						0	0	0	0	0
						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	
WAITC1	Block 1 WAIT control register	69H (Prohibit RMW)				B1BUS	B1W1	B1W0	B1C1	B1C0
						W				
						0	0	0	0	0
						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	
WAITC2	Block 2 WAIT control register	6AH (Prohibit RMW)				B2BUS	B2W1	B2W0	B2C1	B2C0
						W				
						0	0	0	1	1
						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	

(6) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	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							
T10 MOD	8 bit Timer 0,1 Source CLK & MODE	24H	T01M1	T01M0			T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W				R/W			
			0	0			0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : – 11 : –				00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 INPUT 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TFFCR	8 bit Timer Flip-Flop Control	25H	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00 : Invert TFF3 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		1: TFF3 Invert Enable	1: Inversion of Timer 3	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1: TFF1 Invert Enable	1: Inversion of Timer 1
TREG2	8 bit Timer Register 2	26H (Prohibit RMW)	–							
			W							
			Undefined							
TREG3	8 bit Timer Register 3	27H (Prohibit RMW)	–							
			W							
			Undefined							
T32 MOD	8 bit Timer 2,3 Source CLK & MODE	28H	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : $2^6 - 1$ PWM 10 : $2^7 - 1$ Cycle 11 : $2^8 - 1$		00 : TO2TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : – 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TRDC	Timer Reg. Double Buffer Control Reg.	29H							TR2DE	–
									R/W	
									0	0
									0 : Double Buffer Disable 1 : Double Buffer Enable	(Note) Always write "0"

Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
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			CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
					W	R/W				
					1	0	0	0	0	0
					0 : Soft-Capture 1 : Don't care	Capture Timing 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			CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
					R/W				W	
					0	0	0	0	1	1
					TFF4 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF4 01 : Set TFF4 10 : Clear TFF4 11 : Don't care	

Timer Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T45CR	T4, T5 Control	3AH	QCU						DB6EN	DB4EN
			R/W						R/W	
			0						0	0
			Warm-up Timer control						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				
					1	0	0	0	0	0
					0 : Soft-Capture 1 : Don't care	Capture Timing 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	1	1
					TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(7) Serial Channel Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	50H (Prohibit RMW)	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	CTSE0	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			undefined	0	0	0	0	0	0	0
			Transmisson data bit 8	1: CTS0 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: TO2 Trigger 01: Baud rate generator 10: Internal clock $\phi 1$ 11: External clock SCLK0	
BR0CR	Baud Rate 0 Control	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: $\phi T0$ 01: $\phi T2$ 10: $\phi T8$ 11: $\phi T32$		Set frequency divisor 0000 : 16 Divisions 0001 to 1111 : 1 to 15 Divisions			
SC1BUF	Serial Channel 1 Buffer	54H (Prohibit RMW)	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	CTSE1	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			undefined	0	(0→1)	0	0	0	0	0
			Transmisson data bit 8	1: CTS1 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: TO2 Trigger 01: Baud rate generator 10: Internal clock $\phi 1$ 11: External clock SCLK1	
BR1CR	Baud Rate 1 Control	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: $\phi T0$ 01: $\phi T2$ 10: $\phi T8$ 11: $\phi T32$		Set frequency divisor 0000 : 16 Divisions 0001 to 1111 : 1 to 15 Divisions			
ODE	Serial Open Drain Enable	58H					ODE34	ODE33	ODE63	ODE60
							R/W			
							0	0	0	0
							1: P34 Open-drain	1: P33 Open-drain	1: P63 Open-drain	1: P60 Open-drain

(8) Serial Bus Interface Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBICR1	Serial Bus Interface Control Register 1	4BH (I²C Bus Mode) (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0
			W			R/W		W		
			0	0	0	0		0	0	0
			Number of transfer bits 000: 8 001: 1 010: 2 011: 3	100: 4 101: 5 110: 6 111: 7		Acknowledge mode specification 0: Disable 1: Enable		Setting of the divide value "n" 000: 4 001: 5 010: 6 011: 7	100: 8 101: 9 110: 10 111: (reserved)	
		4BH (SIO Mode) (Prohibit RMW)	SIOS	SIOINH	SIONH	SIOM0		SCK2	SCK1	SCK0
			W					W		
			0	0	0	0		0	0	0
			Indicate transfer start / stop 0: Stop 1: Start	Continue / Abort transfer 0: Continue 1: Abort	Transfer mode Select 00: 8-bit transmit 01: (reserved) 10: 8-bit transmit / receive 11: 8-bit receive			Serial clock selection 000: $f_{FPH}/2^5$ 001: $f_{FPH}/2^6$ 010: $f_{FPH}/2^7$ 011: $f_{FPH}/2^8$	100: $f_{FPH}/2^9$ 101: $f_{FPH}/2^{10}$ 110: $f_{FPH}/2^{11}$ 111: External clock (SCK pin)	
SBICR2	Serial Bus Interface Control Register 2	4EH (I²C Bus Mode) (Prohibit RMW)	MST	TRX	BB	PIN	SBIM1	SBIM0	–	–
			W							
			0	0	0	1	0	0	0	0
			Master / Slave selection 0: Slave 1: Master	Transmitter / Receiver selection 0: Receiver 1: Transmitter	Start / Stop generation (when the MST, TRX, PIN are "1") 0: Stop 1: Start	Cancel INTS2 request 0: Don't care 1: Cancel	Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)			
		4EH (SIO Mode) (Prohibit RMW)	–	–	–	–	SBIM1	SBIM0	–	–
							W			
			0	0	0	1	0	0	0	0
							Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)			
SBISR	Serial Bus Interface Control Register 2	4EH (I²C Bus Mode) (Prohibit RMW)	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			R							
			0	0	0	1	0	0	0	0
			Master / Slave selection status monitor 0: Slave 1: Master	Transmitter / Receiver selection status monitor 0: Receiver 1: Transmitter	I²C bus status monitor 0: Bus free 1: Bus busy	INTS2 request status monitor 0: Request 1: Cancel	Arbitration lost detection monitor 1: detect	Slave address much detection monitor 1: detect	GENERAL CALL Detection monitor 1: detect	Last received bit monitor 0: "0" 1: "1"
		4EH (SIO Mode) (Prohibit RMW)	–	–	–	–	SIOF	SEF	–	–
							R			
			0	0	0	1	0	0	0	0
							Serial transfer operating status monitor 0: Terminated 1: In process	Sift operating status monitor 0: Terminated 1: In process		

Serial Bus Interface Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBICR3	Serial Bus Interface Status Register 3	4FH								SWRST
										R/W
										0
										Software reset 0: – 1: Initialize SBI
SBIDBR	Serial Bus Interface Data Buffer Register (Prohibit RMW)	4CH	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
			R (receive) / W (send)							
			Undefined							
I2CAR	I^2C Bus Address Register	4DH (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W							
			0	0	0	0	0	0	0	0
			Slave address selection							Address recognition mode 0: Enable 1: Disable

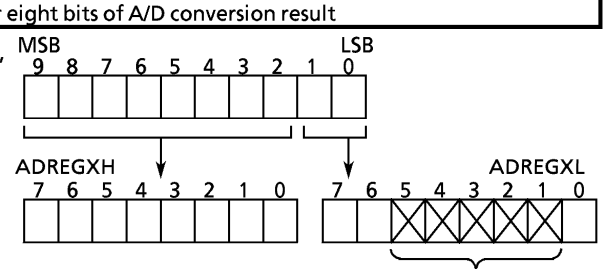
(9) Watch Dog Timer

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: $2^{21}/f_{SYS}$		Warm-up Time 0: 2^{14}/inputted frequency 1: 2^{16}/inputted frequency	HALT 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							

(10) A/D Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD 0	A/D Mode Control Register 0	5EH	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
			R		R/W					
			0	0	0	0	0	0	0	0
			1: End	1: Busy	(Note) Always write "0"		0: Every conversion 1: Every four conversion	0: Single 1: Repeat	0: fixed-channel 1: Scan	1: START
ADMOD 1	A/D Mode Control Register 1	5FH	VREFON			–	ADTRG	ADCH2	ADCH1	ADCH0
			R/W			R/W				
			1			0	0	0	0	0
			0: OFF 1: ON			(Note) Always write "0"	External trigger start control 0: Disable 1: Enable	Analog Input Channel Selection		
AD REG04L	*1) A/D Conversion Result Register 0/4 Low	60H	ADR01	ADR00						ADD0RF
			R							R
			Undefined							0
			Stores lower two bits of A/D conversion result							Conversion result stored flag
AD REG04H	A/D Conversion Result Register 0/4 High	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
			Stores upper eight bits of A/D conversion result							
AD REG15L	*1) A/D Conversion Result Register 1/5 Low	62H	ADR11	ADR10						ADD1RF
			R							R
			Undefined							0
			Stores lower two bits of A/D conversion result							Conversion result stored flag
AD REG15H	A/D Conversion Result Register 1/5 High	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
			Stores upper eight bits of A/D conversion result							
AD REG26L	*1) A/D Conversion Result Register 2/6 Low	64H	ADR21	ADR20						ADD2RF
			R							R
			Undefined							0
			Stores lower two bits of A/D conversion result							Conversion result stored flag
AD REG26H	A/D Conversion Result Register 2/6 High	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
			Stores upper eight bits of A/D conversion result							
AD REG37L	*1) A/D Conversion Result Register 3/7 Low	66H	ADR31	ADR30						ADD3RF
			R							R
			Undefined							0
			Stores lower two bits of A/D conversion result							Conversion result stored flag
AD REG37H	A/D Conversion Result Register 3/7 High	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
			Stores upper eight bits of A/D conversion result							

Converted data of channel 'X'



*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 5 to 1 bits of this register are always read as "1".
Bit 0 conversion result stored flag bit <ADRXRF>
<ADRXRF> is set to "1" when the A/D conversion result is stored.
Reading either the ADREGXH or the ADREGXL registers clears <ADRXRF> to "0".

6. PORT SECTION EQUIVALENT CIRCUIT DIAGRAM

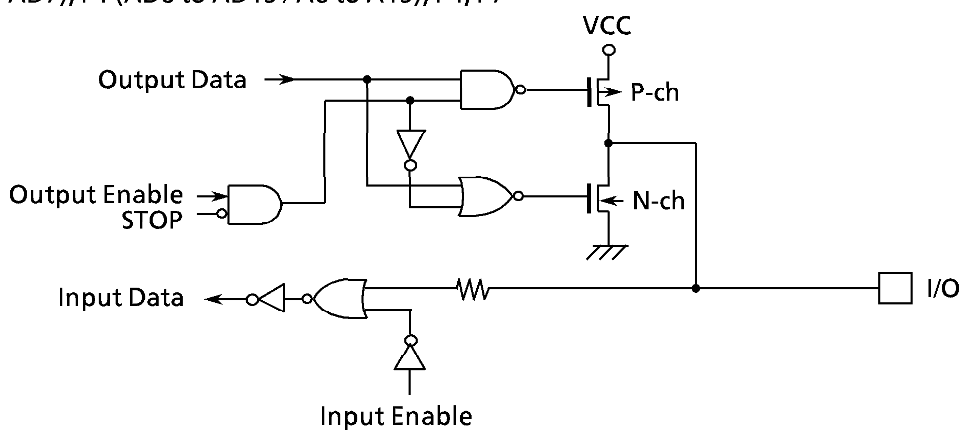
- Reading The Circuit Diagram

  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.
  The dedicated signal is described below.

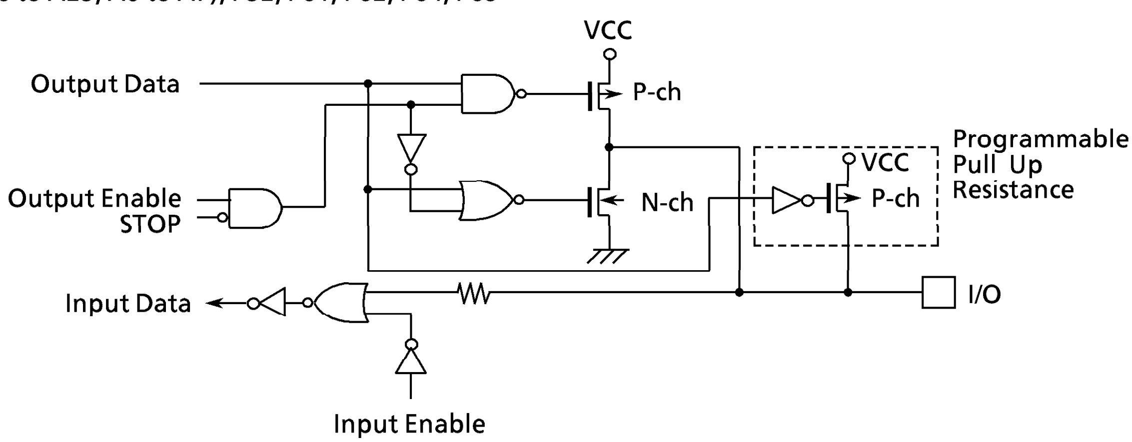
  STOP : This signal becomes active "1" when the halt mode setting register is set to the STOP mode (WDMOD<HALTM1, 0> =0, 1) and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however, STOP remains at "0".

- The input protection resistans ranges from several tens of ohms to several hundreds of ohms.

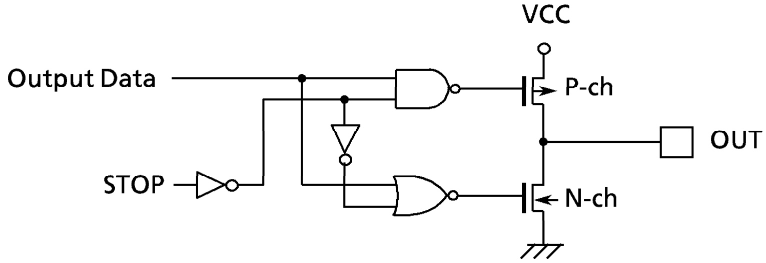
■ P0 (AD0 to AD7), P1 (AD8 to AD15 / A8 to A15), P4, P7



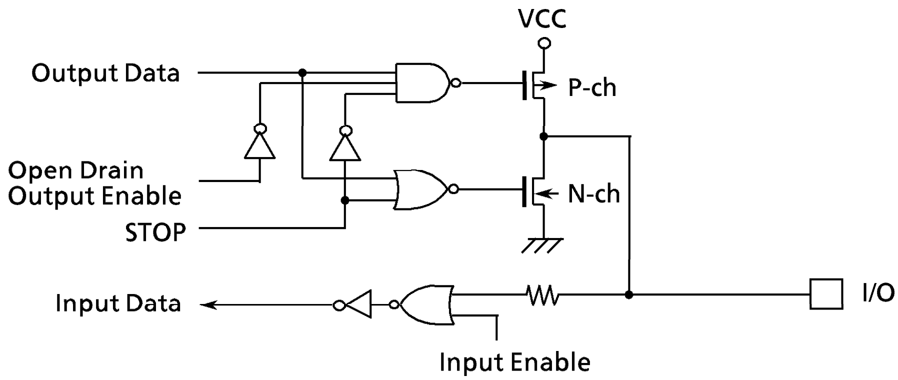
■ P2 (A16 to A23, A0 to A7), P32, P61, P62, P64, P65



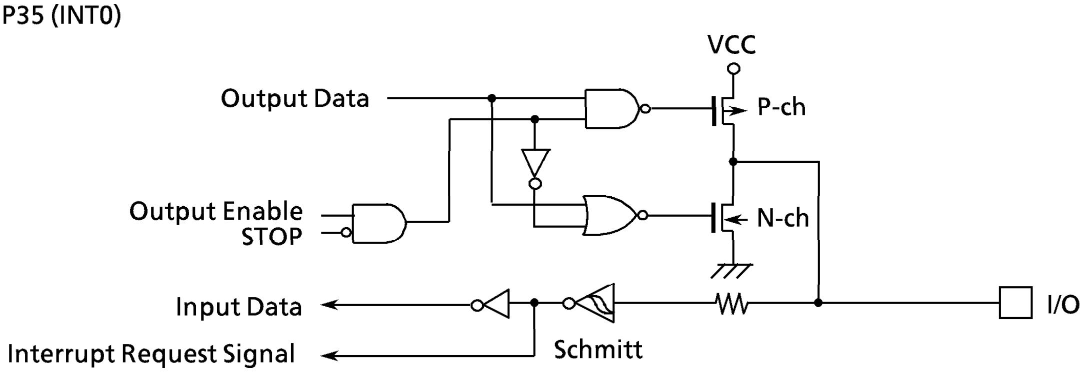
■ P30($\overline{RD}$), P31($\overline{WR}$)



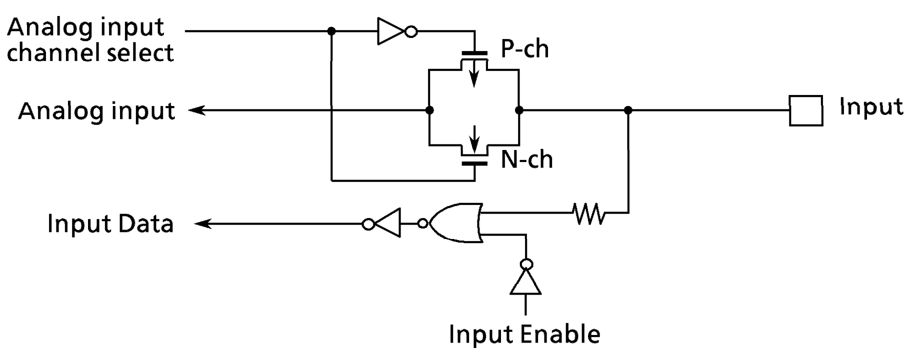
■ P33 (SO/SDA), P34 (SI/SCL)



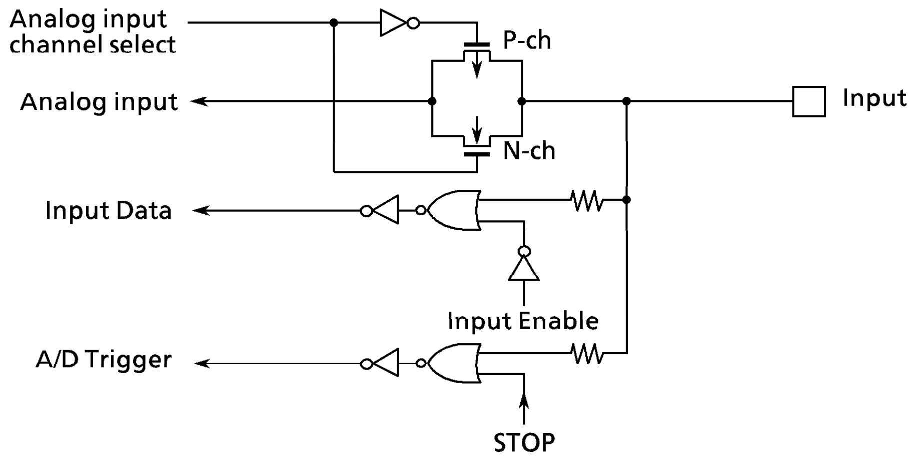
■ P35 (INT0)



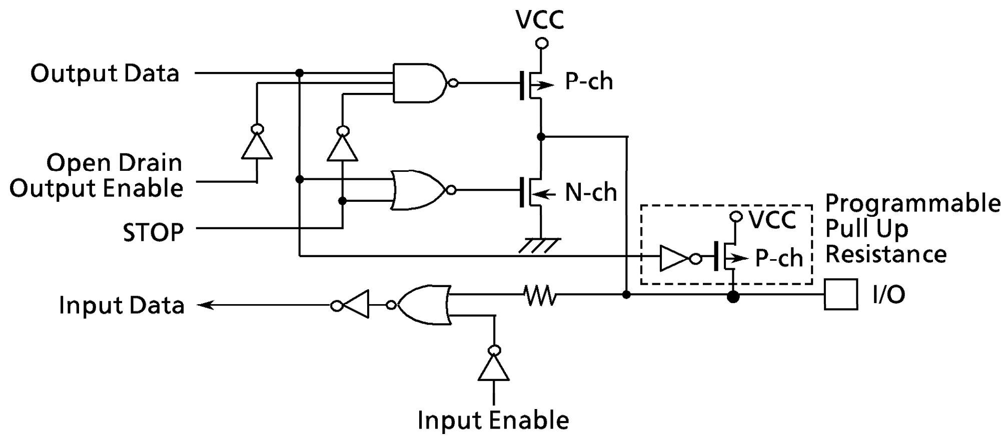
■ P50 to P52 (AN0 to AN2), P54 to P57 (AN4 to AN7)



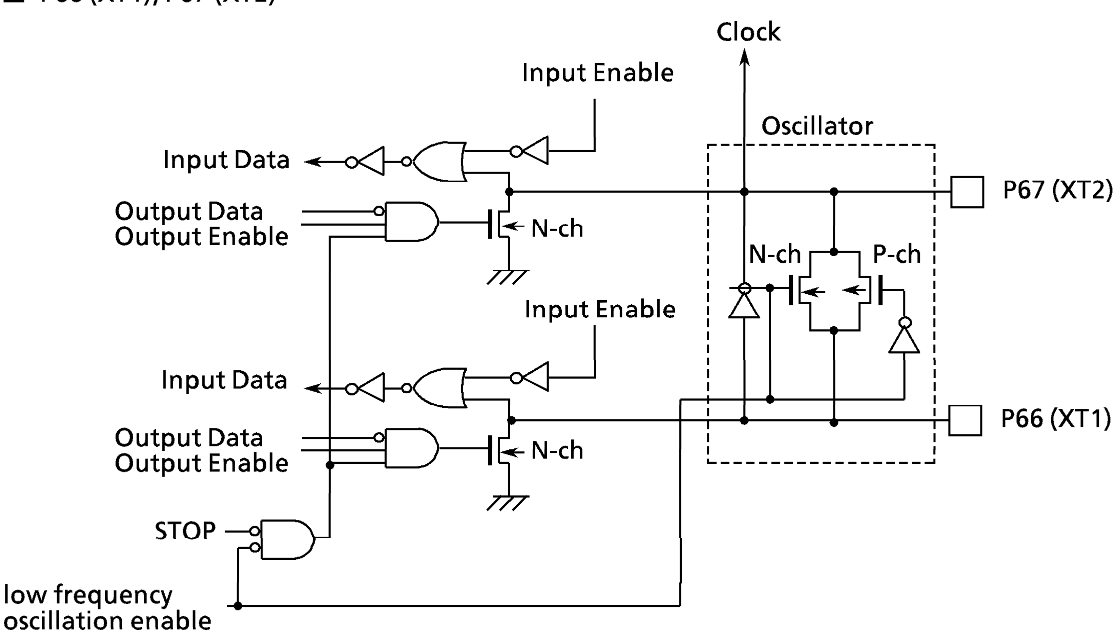
■ P53 (AN3 /$\overline{ADTRG}$)



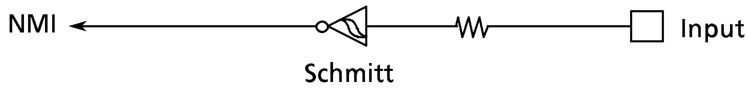
■ P60 (TXD0), P63 (TXD1)



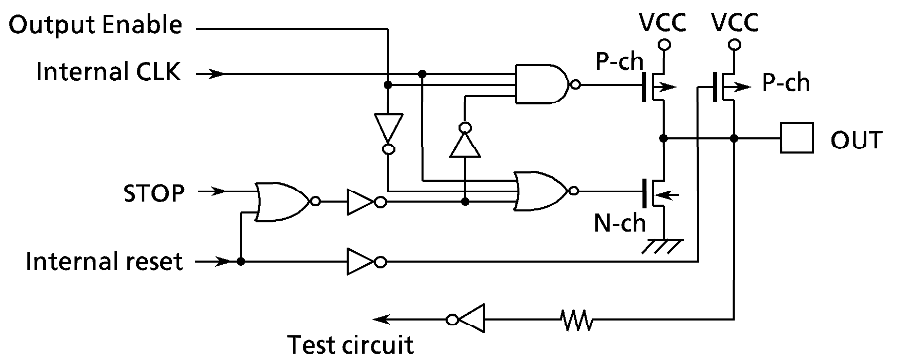
■ P66 (XT1), P67 (XT2)



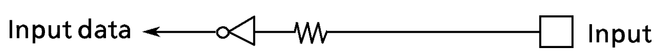
■ $\overline{\text{NMI}}$



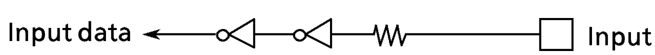
■ CLK



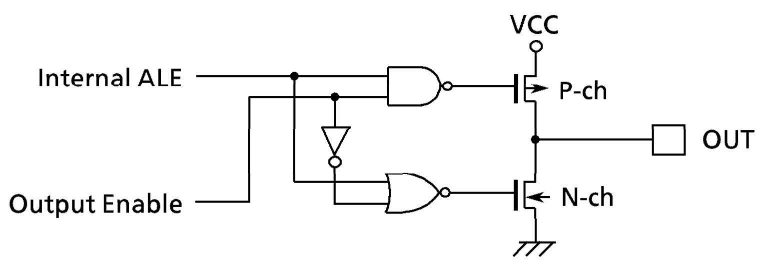
■ $\overline{\text{EA}}$



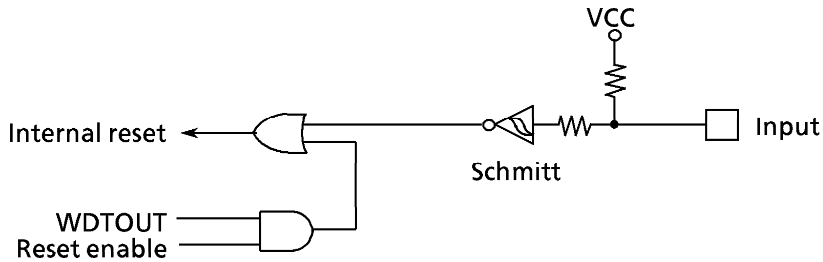
■ AM8 / $\overline{\text{16}}$



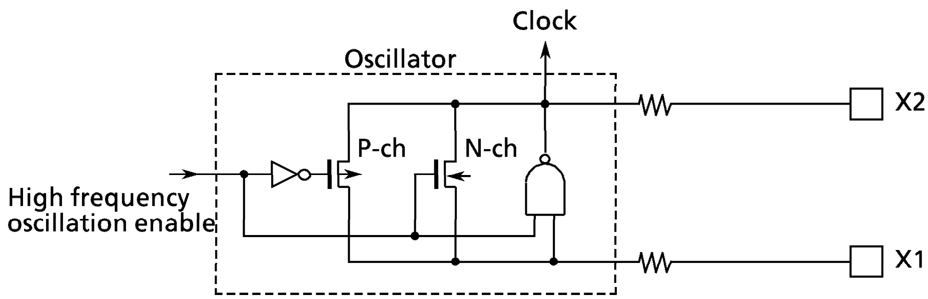
■ ALE



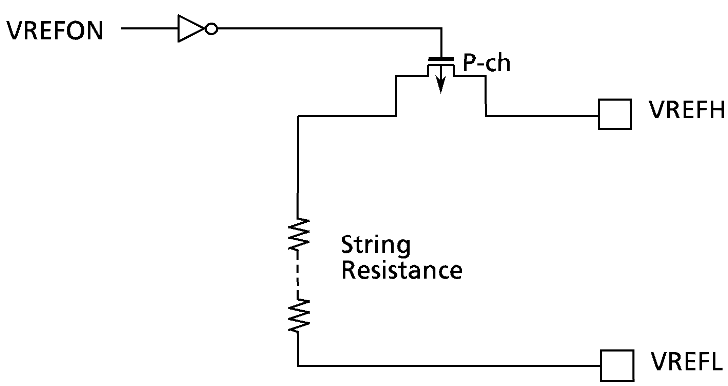
■ $\overline{\text{RESET}}$



■ X1, X2



■ VREFH, VREFL



7. POINTS OF CONCERN AND RESTRICTION

(1) Notation

① Explanation of a built-in I/O register : Register Symbol <Bit Symbol>

e.g.) TRUN<T0RUN> ··· Bit T0RUN of Register TRUN

② Read, Modify and Write Instruction

An instruction in which the 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 bit 3 of TRUN

ex2) INC 1, (100H) ··· increment the data of 100H

- A sample Read, Modify and Write instructions using the TLCS-900

Exchange

EX	(mem), R		

Arithmetic Operation

ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)

Logical Operation

AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		

Bit Manipulation

STCF	#3/A, (mem)	SET	#3, (mem)
RES	#3, (mem)	TEST	#3, (mem)
CHG	#3, (mem)		

Rotate and Shift

RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

③ fc, fs, f_{FPH}, f_{SYS}, 1 state

The clock frequency input from pins X1 and X2 pin is called fc, and the clock frequency input from XT1, XT2 pin is called fs. The clock frequency selected by SYSCR1<SYSCK, GEAR2 to 0> is called system clock f_{FPH}, and the clock frequency given by f_{FPH} divided by 2 is called f_{SYS}. One cycle of f_{SYS} is called 1 state.

(2) Care Points

① $\overline{EA}$, AM8 / $\overline{16}$ pin

Fix these pins Vcc or Vss unless changing voltage.

② TEST1, TEST2 pin

Connect TEST1 pin with TEST2 pin.

③ HALT Mode (IDLE1)

When IDLE1 mode (oscillator operation only) is used, set TRUN <PRRUN> to “0” to stop prescaler before “HALT” instruction is executed.

④ Warmingup Counter

The warm-up counter operates when STOP mode is released even if the system is using an external oscillator. As a result, it takes warm-up time from inputting the releasing request to outputting the system clock.

⑤ Programmable Pull Up Resistance

The programmable pull up resistances can be turned ON / OFF by the program when the ports are used as input ports. When the ports are used as the output ports, they can not be selected ON/OFF by the program.
The data registers (e.g. P6 register) are used for the pull-up resistors ON/OFF. Consequently, Read-modify-write instructions are prohibited.

⑥ WatchDog Timer

The watchdog timer starts operation immediately after the reset is released. When the watchdog timer is not used, disable it.

⑦ A/D Converter

The string register between VREFH and VREFL pins can be cut by a program to reduce power consumption. When the Standby mode is used, disable the resistor using the program before the “HALT” instruction is executed.

⑧ CPU (Micro DMA)

Only the “LDC cr, r”, “LDC r, cr” instructions can be used to access the control registers in the CPU (like the transfer source address register (DMASn)).

⑨ POP SR instruction

Please execute POP SR instruction during DI condition.

8. TMP93XX44/45 DIFFERENT POINTS

ITEM	93CS44	93CS45	93PS44	93CU44	93CW44	93PW44
Built-in ROM	64K byte Mask ROM (FF0000H to FFFFFFH)	None	64K byte OTP (FF0000H to FFFFFFH)	96K byte Mask ROM (FE8000H to FFFFFFH)	128K byte Mask ROM (FE0000H to FFFFFFH)	128K byte OTP (FE0000H to FFFFFFH)
Built-in RAM	2K byte (80H to 87FH)	2K byte (80H to 87FH)	2K byte (80H to 87FH)	3K byte (80H to C7FH)	4K byte (80H to 107FH)	4K byte (80H to 107FH)
CS1 Mapping Area (WAITC1<B1C1,0> = 00)	880H to 7FFFH	880H to 7FFFH	880H to 7FFFH	C80H to 7FFFH	1080H to 7FFFH	1080H to 7FFFH
CS2 Mapping Area (WAITC2<B2C1,0> = 11)	C00000H to FEFFFFH	C00000H to FFFFFFH	C00000H to FEFFFFH	C00000H to FE7FFFH	C00000H to FDFFFFH	C00000H to FDFFFFH