CMOS 16-bit Microcontrollers

TMP93CS42AF

1. Outline and Device Characteristics

The TMP93CS42A is high-speed advanced 16-bit microcontroller developed for controlling medium to large-scale equipment. The TMP93CS42A has a built-in ROM.
The TMP93CS42AF is housed in 100-pin flat package.
The device characteristics are as follows:

(1) Original 16-bit CPU (900/L CPU)

- TLCS-90 instruction mnemonic upward compatible
- 16 M-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 / 2 bytes at 20 MHz)

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

(3) Internal RAM : 2 Kbytes
Internal ROM : 64 Kbytes

(4) External memory expansion

- Can be expanded up to 16 M-bytes (for both programs and data).
- Can mix 8- and 16-bit external data buses.
  ···Dynamic data bus sizing

(5) 8-bit timer : 2 channels

(6) 8-bit PWM timer : 2 channels

(7) 16-bit timer : 2 channels

(8) Serial interface : 2 channels

(9) 10-bit AD converter : 5 channels

(10) Watchdog timer

(11) Chip select/wait controller : 3 blocks

(12) Interrupt functions : 29

- 9 CPU interrupts ……SWI instruction, and Illegal instruction
- 12 internal interrupts
- 8 external interrupts

} 7-level priority can be set.

(13) I/O ports :

80 pins

(14) Standby function :

4 halt modes (RUN, IDLE2, IDLE1, STOP)

(15) Clock gear function

- Clock can be changed fc to fc/16.

(16) Operating voltage

- Vcc = 4.5 to 5.5 V

(17) Package

- P-QFP100-1414-0.50

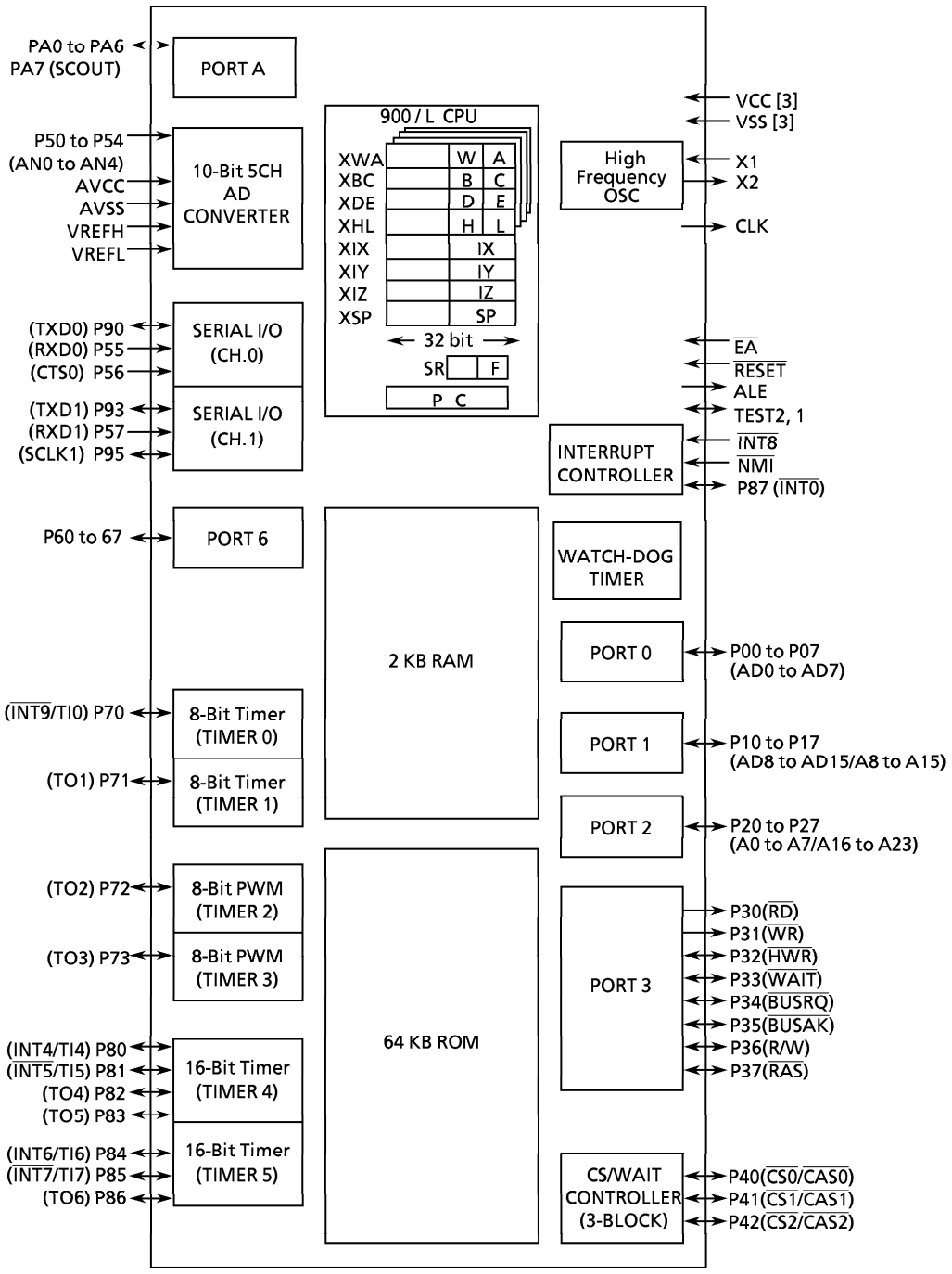


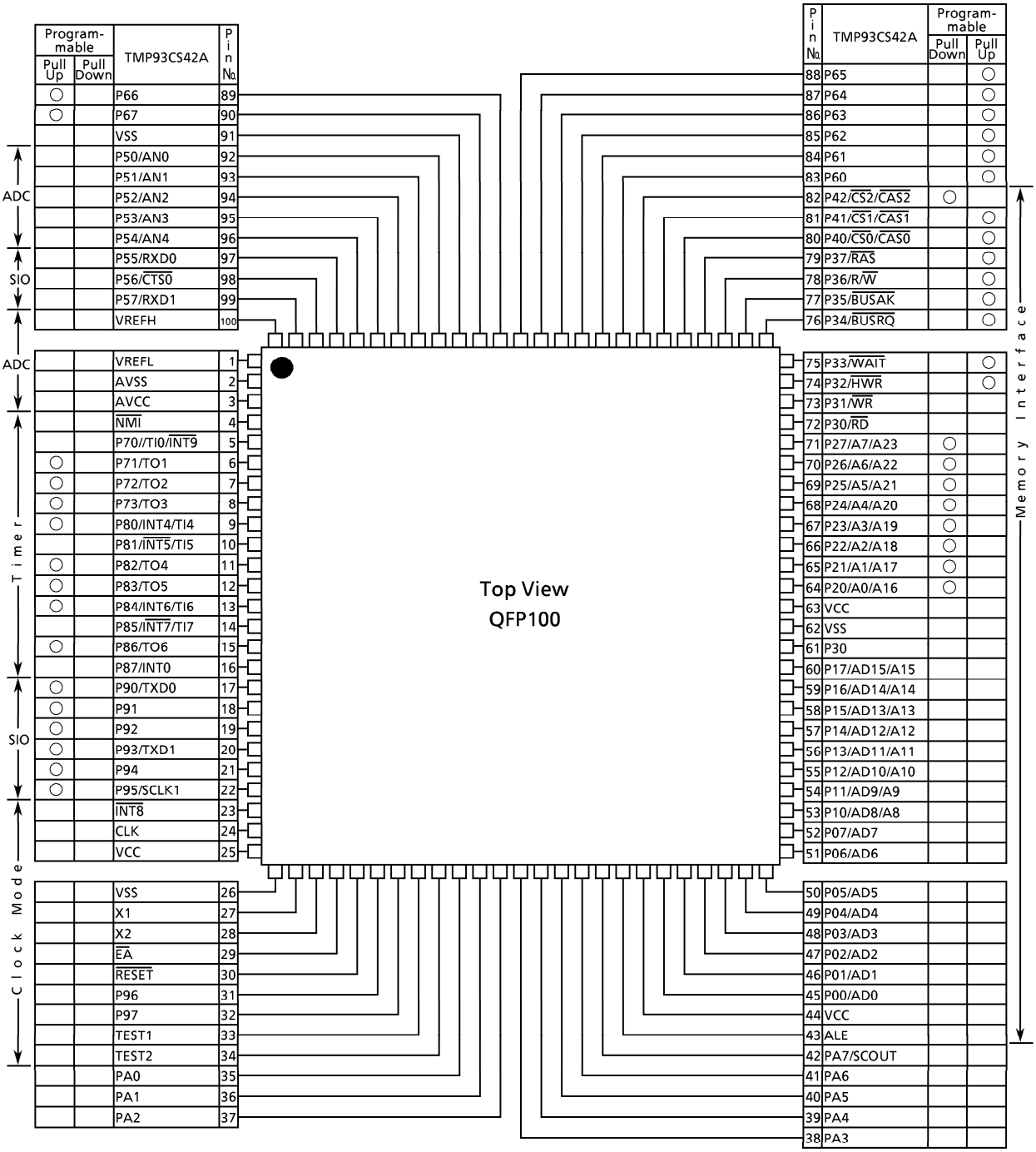
Figure 1.1 TMP93CS42A Block Diagram

2. Pin Assignment and Function

The assignment of input/output pins for the TMP93CS42A, their names and outline functions are described below.

2.1 Pin Assignment

Figure 2.1.1 shows pin assignment of the TMP93CS42AF.



Programmable Pull Up	Programmable Pull Down	TMP93CS42A	Pin No.
○		P66	89
○		P67	90
		VSS	91
		P50/AN0	92
		P51/AN1	93
		P52/AN2	94
		P53/AN3	95
		P54/AN4	96
		P55/RXD0	97
		P56/$\overline{CTS0}$	98
		P57/RXD1	99
		VREFH	100
		VREFL	1
		AVSS	2
		AVCC	3
		$\overline{NMI}$	4
		P70//TI0/$\overline{INT9}$	5
○		P71/TO1	6
○		P72/TO2	7
○		P73/TO3	8
○		P80/INT4/TI4	9
		P81/$\overline{INT5}$/TI5	10
○		P82/TO4	11
○		P83/TO5	12
○		P84/INT6/TI6	13
		P85/$\overline{INT7}$/TI7	14
○		P86/TO6	15
		P87/INT0	16
○		P90/TXD0	17
○		P91	18
○		P92	19
○		P93/TXD1	20
○		P94	21
○		P95/SCLK1	22
		$\overline{INT8}$	23
		CLK	24
		VCC	25
		VSS	26
		X1	27
		X2	28
		$\overline{EA}$	29
		$\overline{RESET}$	30
		P96	31
		P97	32
		TEST1	33
		TEST2	34
		PA0	35
		PA1	36
		PA2	37

Pin No.	TMP93CS42A	Programmable Pull Down	Programmable Pull Up
88	P65		○
87	P64		○
86	P63		○
85	P62		○
84	P61		○
83	P60		○
82	P42/$\overline{CS2}$/$\overline{CAS2}$	○	
81	P41/$\overline{CS1}$/$\overline{CAS1}$		○
80	P40/$\overline{CS0}$/$\overline{CAS0}$		○
79	P37/$\overline{RAS}$		○
78	P36/R/$\overline{W}$		○
77	P35/$\overline{BUSAK}$		○
76	P34/$\overline{BUSRQ}$		○
75	P33/$\overline{WAIT}$		○
74	P32/$\overline{HWR}$		○
73	P31/$\overline{WR}$		
72	P30/$\overline{RD}$		
71	P27/A7/A23	○	
70	P26/A6/A22	○	
69	P25/A5/A21	○	
68	P24/A4/A20	○	
67	P23/A3/A19	○	
66	P22/A2/A18	○	
65	P21/A1/A17	○	
64	P20/A0/A16	○	
63	VCC		
62	VSS		
61	P30		
60	P17/AD15/A15		
59	P16/AD14/A14		
58	P15/AD13/A13		
57	P14/AD12/A12		
56	P13/AD11/A11		
55	P12/AD10/A10		
54	P11/AD9/A9		
53	P10/AD8/A8		
52	P07/AD7		
51	P06/AD6		
50	P05/AD5		
49	P04/AD4		
48	P03/AD3		
47	P02/AD2		
46	P01/AD1		
45	P00/AD0		
44	VCC		
43	ALE		
42	PA7/SCOUT		
41	PA6		
40	PA5		
39	PA4		
38	PA3		

Figure 2.1.1 shows pin assignment of TMP93CS42AF.

2.2 Pin Names and Functions

Table 2.2.1 Pin Names and Functions.

Table 2.2.1 Pin Names and Functions (1/4)

Pin name	Number of pins	I/O	Function
P00 to P07 AD0 to AD7	8	I/O Tri-state	Port 0: I/O port that allows selection of I/O on a bit basis Address/data (lower): Bits 0 to 7 of address/data bus
P10 to P17 AD8 to AD15 A8 to A15	8	I/O Tri-state Output	Port 1: I/O port that allows selection of I/O on a bit basis Address data (upper): Bits 8 to 15 of address/data bus Address: Bits 8 to 15 of address bus
P20 to P27 A0 to A7 A16 to A23	8	I/O Output Output	Port 2: I/O port that allows selection of I/O on a bit basis (with pull-down resistor) Address: Bits 0 to 7 of address bus Address: Bits 16 to 23 of address bus
P30 $\overline{RD}$	2 1	Output Output	Port 30: Output port. 72 pin is also used as $\overline{RD}$, 61 pin is used only for P30. Read: Strobe signal for reading external memory
P31 $\overline{WR}$	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 to 7
P32 $\overline{HWR}$	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 to 15
P33 $\overline{WAIT}$	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 $\overline{BUSRQ}$	1	I/O Input	Port34: I/O port (with pull-up resistor) Bus request: Pin used to request bus release
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O port (with pull-up resistor) Bus acknowledge: Pin used to acknowledge bus release
P36 R/$\overline{W}$	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0, write cycle.
P37 $\overline{RAS}$	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs $\overline{RAS}$ strobe for DRAM.
P40 $\overline{CS0}$ $\overline{CAS0}$	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs $\overline{CAS}$ strobe for DRAM when address is within specified address area.

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

Table 2.2.1 Pin Names and Functions (2/4)

Pin name	Number of pins	I/O	Function
P41 $\overline{CS1}$ $\overline{CAS1}$	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P42 $\overline{CS2}$ $\overline{CAS2}$	1	I/O Output Output	Port 42: I/O port (with pull-down resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P50 to P57 AN0 to AN4 RXD0 $\overline{CTS0}$ RXD1	8	Input Input Input Input Input	Port 5: Input port Analog input: Analog signal input for AD converter Serial receive data 0 Serial data send enable 0 (Clear to Send) Serial receive data 1
VREFH	1	Input	Pin for high level reference voltage input to AD converter
VREFL	1	Input	Pin for low level reference voltage input to AD converter
P60 to P67	8	I/O	Ports 60 to 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor)
P70 TI0 $\overline{INT9}$	1	I/O Input Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input Interrupt request pin 9: Interrupt request pin with falling edge. (TTL level)
P71 TO1	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 TO2	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 TO3	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 TI4 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count / capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising / falling edge.
P81 TI5 $\overline{INT5}$	1	I/O Input Input	Port 81: I/O port Timer input 5: Timer 4 count / capture trigger signal input Interrupt request pin 5: Interrupt request pin with falling edge. (TTL level)
P82 TO4	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 TO5	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Table 2.2.1 Pin Names and Functions (3/4)

Pin name	Number of pins	I/O	Function
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count / capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising / falling edge
P85 TI7 $\overline{INT7}$	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count / capture trigger signal input Interrupt request pin 7: Interrupt request pin with falling edge (TTL level)
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 $\overline{INT0}$	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0 : Interrupt request pin with programmable level/falling edge (TTL level)
P90 TXD0	1	I/O Input	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91	1	I/O	Port 91: I/O port (with pull-up resistor)
P92	1	I/O	Port 92: I/O port (with pull-up resistor)
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) Serial send data 1
P94	1	I/O	Port 94: I/O port (with pull-up resistor)
P95 SCLK1	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O 1
P96	1	I/O	Port 96: I/O port (Open Drain Output)
P97	1	I/O	Port 97: I/O port (Open Drain Output)
PA0 to PA6	7	I/O	Port A0 to A6 : I/O ports
PA7 SCOUT	1	I/O Output	Port A7: I/O port System Clock Output: Outputs f_{FPH} or f_{SYS} clock
$\overline{INT8}$	1	Input	Interrupt request pin 8 : Interrupt request pin with falling edge. (TTL level)
$\overline{NMI}$	1	Input	Non-maskable interrupt request pin : Interrupt request pin with programmable falling/rising edge
CLK	1	Output	Clock output: Outputs [f_{SYS}] Clock. Pulled-up during reset. can be disabled for reducing noise.
$\overline{EA}$	1	Input	External access: "1" should be inputted. (See Note on usage ⑭ in section 7.)
ALE	1	Output	Address Latch Enable (Can be disabled for reducing noise.)
$\overline{RESET}$	1	Input	Reset: Initializes TMP93CS42A. (with pull-up resistor)
X1/X2	2	I/O	High Frequency Oscillator connecting pin
TEST1 / TEST2	2	Output / Input	TEST1 Should be connected with TEST2 pin.

Table 2.2.1 Pin Names and Functions (4/4)

Pin name	Number of pins	I/O	Function
VCC	3		Power supply pin (All VCC pins are connected to the power supply source.)
VSS	3		GND pin (All VSS pins are connected with GND (0 V).)
AVCC	1		Power supply pin for AD converter
AVSS	1		GND pin for AD converter (0 V)

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

3. Operation

This section describes the functions and basic operational blocks of the TMP93CS42A devices.
See the「7. Points of Concern and Restrictions」for the using notice and restrictions for each block.

3.1 CPU

The TMP93CS42A device has 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 TMP93CS42A that are not described in the previous section.

3.1.1 Reset

To reset the TMP93CS42A, 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 8000H to 8002H.
  PC (7 - 0) ← stored data to 8000H
  PC (15 - 8) ← stored data to 8001H
  PC (23 - 16) ← stored data to 8002H
- Stack pointer (XSP) for system mode to 100H.
- Status register <IFF2-0> to 111. (Sets mask register to interrupt level 7.)
- Status register <MAX> to 1. (Sets to maximum mode)
- Status register <REP2-0> 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 are 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.
- Watchdog timer is set to enable.
- Pulls up the CLK pin to 1.
- Sets the ALE pin to High Impeadance (High-z)

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 during reset. When the voltage is externally put down, there is possible to cause malfunctions.

Figure 3.1.1 shows the reset timing chart of TMP93CS42A.

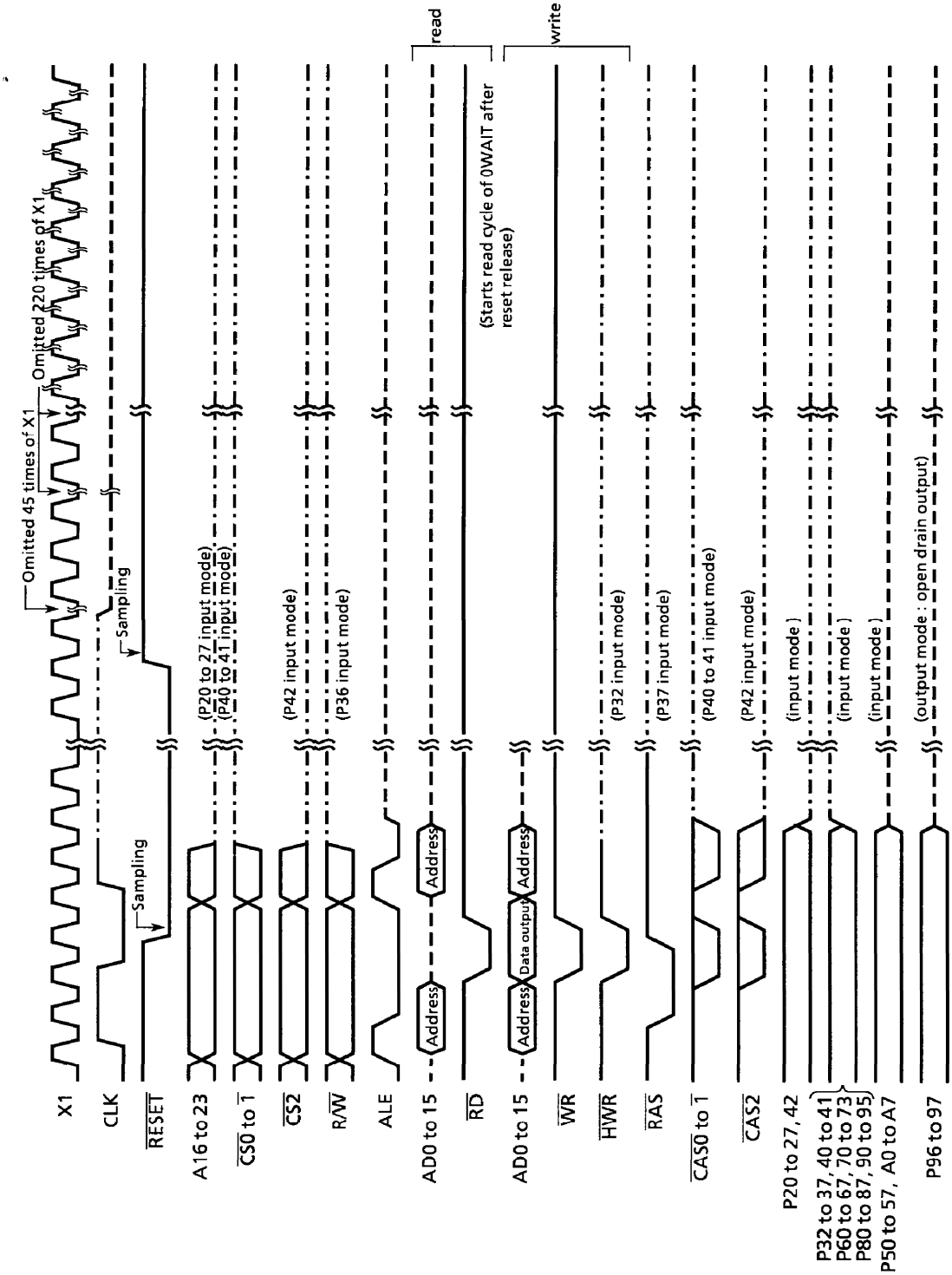
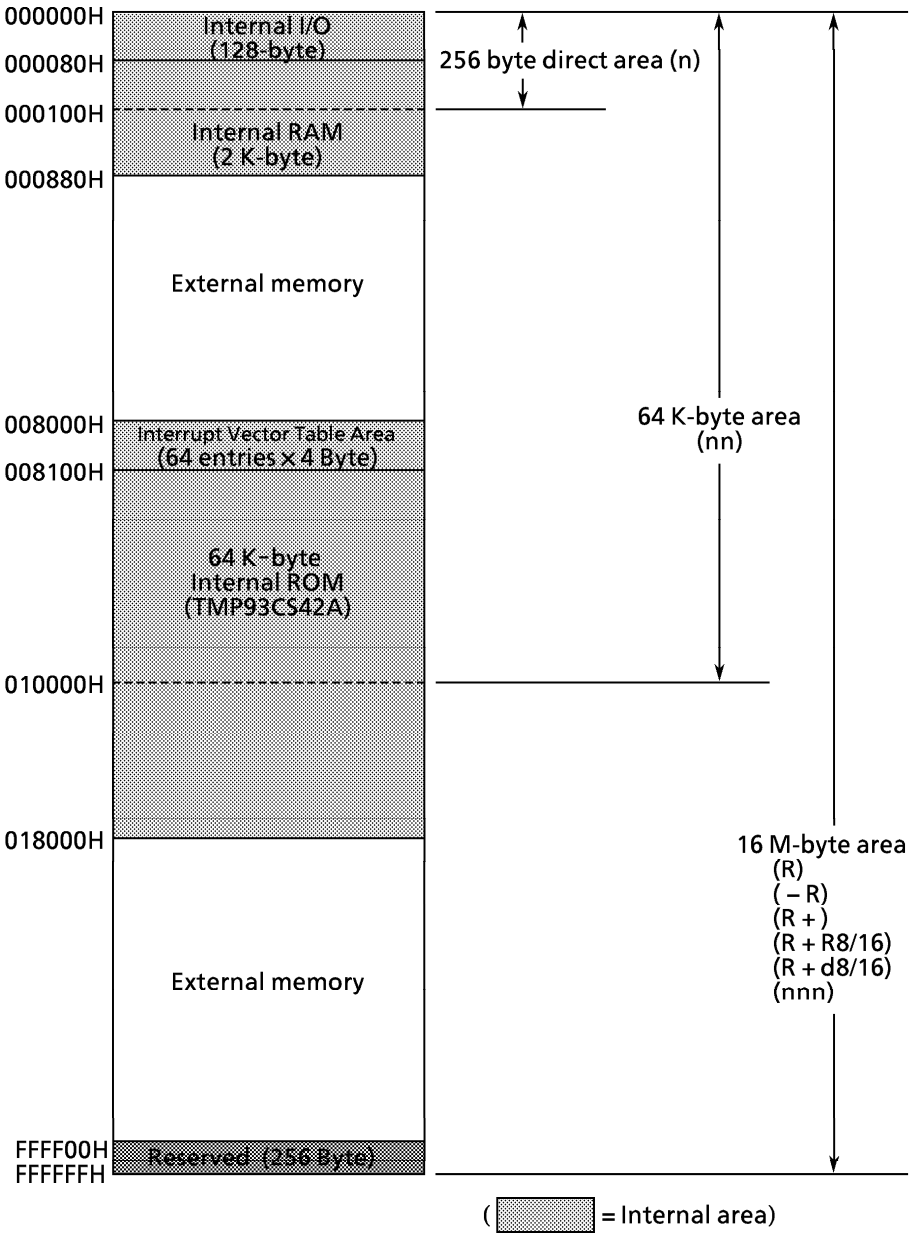


Figure 3.1.1 TMP93CS42A Reset Timing Chart

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP93CS42A.



Note : The 256 Byte Area from FFFF00H to FFFFFFH can not be used.

Figure 3.2.1 Memory map

3.3 Standby Function

Standby Control Circuits consists of System clock Controller, Prescaler Clock Controller, Internal clock pin output function and Standby Controller.
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 internal operation and system clock.

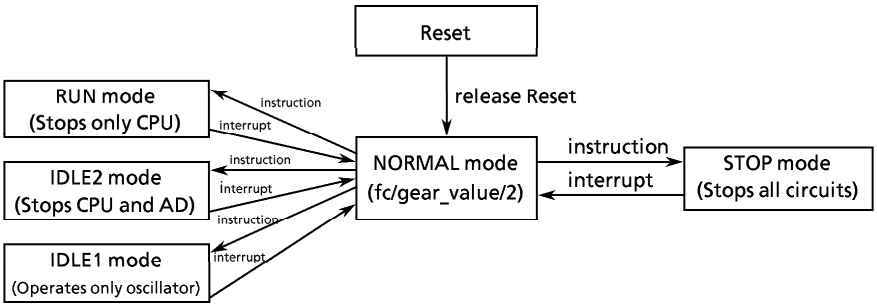


Figure 3.3.1 Clock mode transition figure

The Clock Frequency input from X1, X2 pin is called fc, and the clock frequency selected by SYSCR1<SYSCK>, <GEAR2-0> 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 defined as 1 state.

Table 3.3.1 Internal operation and system clock

Operating Mode	Oscillator (fc)	CPU	internal I/O	System clock f_{SYS}
RESET	oscillation	reset	reset	$fc/32$
NORMAL		operate	operate	programmable ($fc/2$, $fc/4$, $fc/8$, $fc/16$, $fc/32$)
RUN		stop		
IDLE2			stop only AD	
IDLE1			stop	
STOP	stop			stop

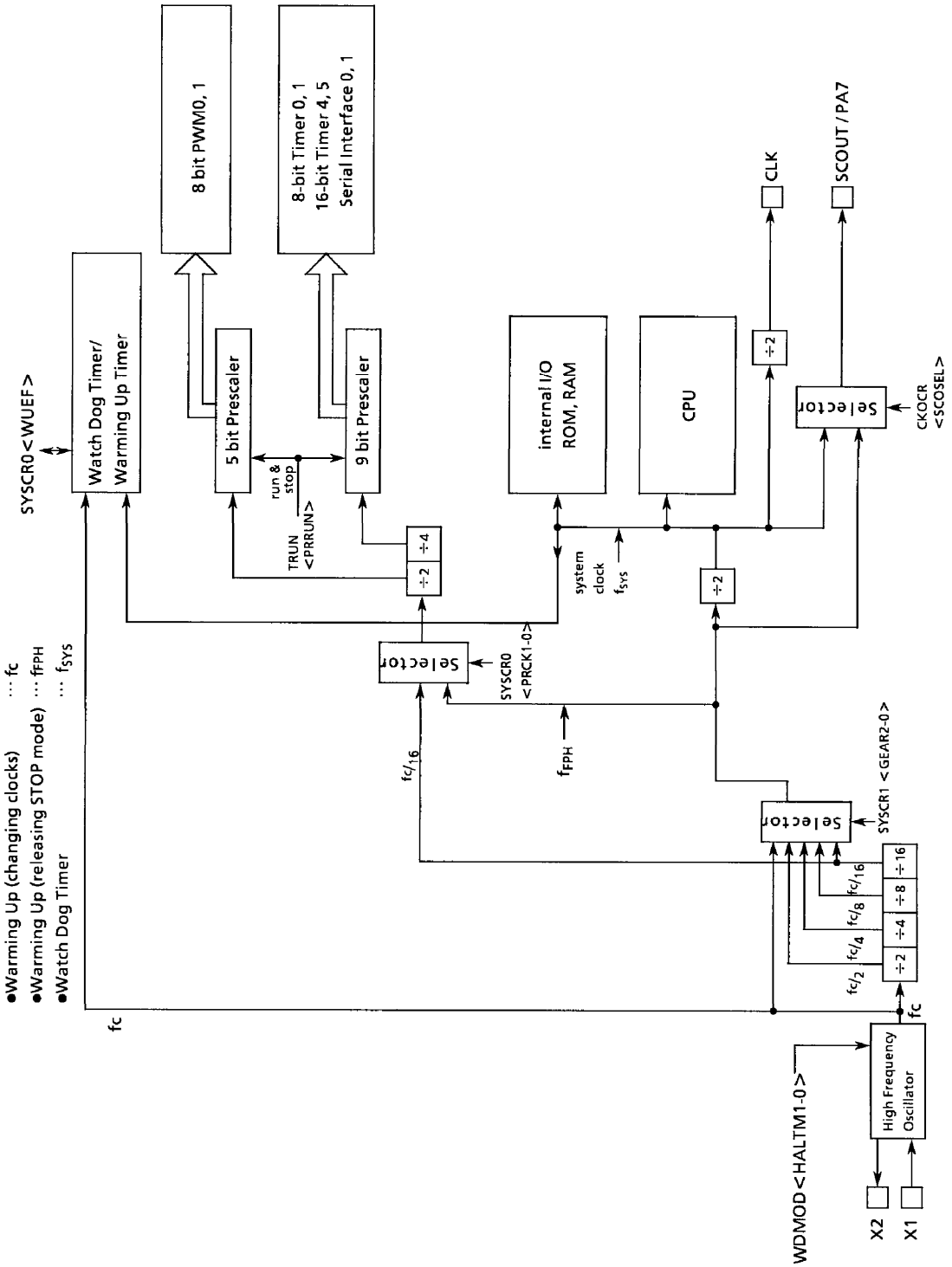


Figure 3.3.2 Block Diagram of Standby circuit

		7	6	5	4	3	2	1	0
SYSCR0 (006EH)	Bit symbol							PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	0	1	0	0	0	0	0
	Function	Set to 1	Set to 0	Set to 1	Set to 0	Set to 0	Set to 0	Prescaler clock select 00 : f_{FPH} 01 : (reserved) 10 : fc/16 11 : (reserved)	
SYSCR1 (006FH)	Bit symbol						GEAR2	GEAR1	GEAR0
	Read/Write					R/W			
	After reset					0	1	0	0
	Function					Set to 0	select gear value of high frequency 000 : fc 001 : fc/2 010 : fc/4 011 : fc/8 100 : fc/16 101 : (reserved) 110 : (reserved) 111 : (reserved)		
WDMOD (005CH)	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 1 : enable	WDT Detection Time 00 : $2^{15}/f_{SYS}$ 01 : $2^{17}/f_{SYS}$ 10 : $2^{19}/f_{SYS}$ 11 : $2^{21}/f_{SYS}$		Warming Up Timer 0 : 2^{14}/ inputted frequency 1 : 2^{16}/ inputted frequency	Standby mode 00 : RUN mode 01 : STOP mode 10 : IDLE1 mode 11 : IDLE2 mode		1 : Connects WDT output to $\overline{RESET}$ pin internally.	1 : Drives pin even in STOP mode
CKOCR (006DH)	Bit symbol					SCOSEL	SCOEN	ALEEN	CLKEN
	Read/Write	R/W							
	After reset					0	0	0	0 Note 2
	Function					SCOUT select 0 : f_{FPH} 1 : f_{SYS}	SCOUT output control 0 : I/O ports 1 : SCOUT output	ALE pin output control 0 : High-z 1 : ALE output	CLK pin output control 0 : High-z 1 : CLK output

Note 1 : SYSCR1<bit7 to 4> is always read as "1".
Note 2 : The CLK pin is internally pulled up during reset.

Figure 3.3.3 I/O register about Standby

3.3.1 System Clock Controller

The system clock controller generates system clock (f_{SYS}) for CPU core and internal I/O. 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), these function can reduce the power consumption.
The system clock (f_{SYS}) is set to fc/32 (fc/16×1/2) because of <GEAR2-0>="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 clock (fc) can be easily obtained by connecting a resonator to the X1, X2 pins. Clock input from an external oscillator is also possible.

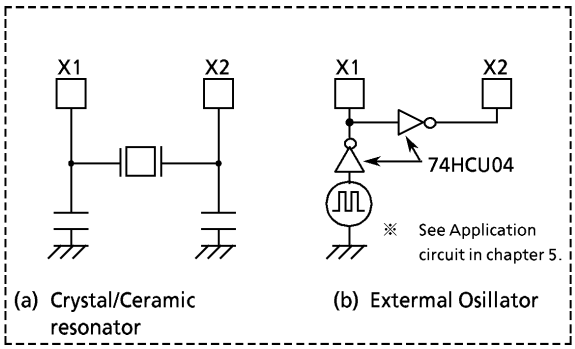


Figure 3.3.4 Examples of Resonator Connection

Note : Accurate Adjustment of the Oscillation Frequency

The CLK pin outputs 1/2 system clock frequency ($f_{SYS}/2$) to monitor the oscillation clock.
With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

(High-speed clock gear changing)

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

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

3.3.2 Prescaler Clock Controller

The 9-bit prescaler provides a clock to 8-bit Timer 0, 1, 8 bit PWM0, 1, 16-bit Timer 4, 5, and Serial Interface 0, 1. And the 5-bit prescaler provides a clock to 8 bit PWM0, 1.
The clock input to the 5-bit prescaler is a clock divided by 2 which is selected either f_{FPH}or fc/16 by SYSCR0 < PRCK1, 0> register.
The clock input to the 9-bit prescaler is a clock divided by 4 which is selected either f_{FPH}or fc/16 by SYSCR0 < PRCK1, 0> register.
<PRCK1, 0> register is initialized to "00" by resetting. The clock selected by <SYSCK> is input.

When the IDLE 1 mode (operates only oscillator) is used, set TRUN<PRRUN> to '0' to reduce the power consumption of 9, 5-bit prescaler before 'HALT' instruction is executed.

3.3.3 Internal Clock Pin Output Function

(1) PA7/SCOUT pin

PA7/SCOUT pin outputs the internal clocks f_{FPH} or f_{SYS}.
The port A control register PACR<PA7C> and the clock output control register CKOCR<SCOEN, SCOSEL> specifies the clock and the pins. PA7/SCOUT pin is used as the input port by reset.
Table 3.3.2 shows Pin states in the respective operation modes which is under condition that PA7/SCOUT pin is specified as SCOUT output.

Table 3.3.2 SCOUT pin states in the operation modes

Operation / Output clock	NORMAL	Halt mode	
		RUN, IDLE2, IDLE1	STOP
f_{FPH}	Outputs f_{FPH} clock.		Fixed to "0" or "1".
f_{SYS}	Outputs f_{SYS} clock.		

(2) CLK pin

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. CKOCR<CLKEN> is set to "0" after reset.
During reset, CLK pin is internally pulled up regardless of the value of <CLKEN> register.

Note : To set <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.

3.3.4 Standby Controller

(1) Halt mode

When the HALT instruction is executed, the operating mode changes RUN, IDLE2, IDLE1 or STOP mode depending on the contents of the HALT mode setting register WDMOD<HALTM1,0>. Figure 3.3.5 shows the watchdog timer mode registers.

Watchdog timer mode register

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	Watchdog timer control 0: disable 1: enable	Watchdog timer detect time selection 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warming-up timer 0: 2^{14}/select clock frequency 1: 2^{16}/select clock frequency	Halt mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Runaway detect internal reset control 1: Executes internal reset by runaway detect	STOP mode pin control 1: Drive pins in STOP mode

Pin state control in STOP mode

0	I/O off
1	Retains the state before halt

Halt mode setting

00	RUN mode (only CPU stop)
01	STOP mode (all circuits stop)
10	IDLE1 mode (only oscillator operating)
11	IDLE2 mode (a part I/O operating)

Warming-up time selection at returning from the stop mode (see Table 3.3.5)

0	2^{14}/ select clock frequency
1	2^{16}/ select clock frequency

Figure 3.3.5 Watchdog timer mode register

The futures of RUN, IDLE2, IDLE1 and STOP modes are as follows.

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

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

IDLE1 : Only the built-in oscillator operates, while all other built-in circuits stop.
The Power Consumption is reduced to 1/5 or less than that during NORMAL operation.

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

The operations in the halt state is described in Table 3.3.3.

Table 3.3.3 I/O Operation During Halt mode

Halt mode		RUN	IDLE2	IDLE1	STOP
WDMOD <HALTM1, 0>		00	11	10	01
Block	CPU	Halt			
	I/O port	Keep the state when the "HALT" instruction was executed.			See Table 3.3.6
	8 bit Timer	Operate		Stop	
	8 bit PWM Timer				
	16 bit Timer				
	Serial Channel				
	AD Converter				
	Watchdog Timer				
	Interrupt Controller				

(2) How to Release the Halt mode

These HALT states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combinations between the states of interrupt mask register <IFF2 to 0> and the halt modes. The details for releasing the HALT status are shown in Table 3.3.4.

- Released by requesting an interrupt

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

- Release by resetting

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

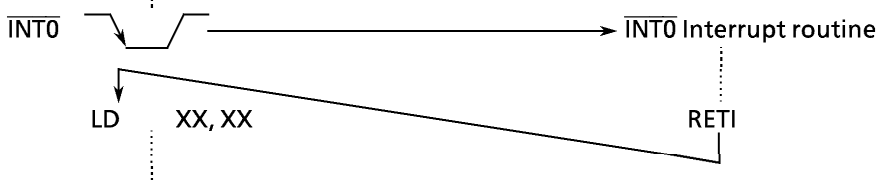
Table 3.3.4 Halt releasing source and Halt releasing operation

Interrupt receiving status			Interrupt enable (Interrupt level) ≧ (Interrupt mask)				Interrupt disable (interrupt level) < (interrupt mask)			
Halt mode			RUN	IDLE2	IDLE1	STOP	RUN	IDLE2	IDLE1	STOP
Halt Releasing Source	Interrupt	NMI	◆	◆	◆	◆*1	–	–	–	–
		INTWDT	◆	×	×	×	–	–	–	–
		INT0	◆	◆	◆	◆*1	○	○	○	○*1
		INT4 to 9	◆	◆	×	×	×	×	×	×
		INTT0, 1	◆	◆	×	×	×	×	×	×
		INTTR4 to 7	◆	◆	×	×	×	×	×	×
		INTRX0, TX0	◆	◆	×	×	×	×	×	×
		INTRX1, TX1	◆	◆	×	×	×	×	×	×
		INTAD	◆	×	×	×	×	×	×	×
	RESET		◆	◆	◆	◆	◆	◆	◆	◆

◆ : After releasing the halt mode, CPU starts interrupt processing. (RESET initializes LSI.)
○ : After releasing the halt mode, CPU starts executing an instruction that follows the HALT instruction.
× : It can not be used to release the halt mode.
– : The priority level (interrupt request level) of non-maskable interrupts is fixed to highest priority level "7". There is not this combination type.
*1 : Releasing the halt mode is executed after passing the warming-up time.
Note : When releasing the halt mode is executed by $\overline{\text{INT0}}$ interrupt of the level mode in the interrupt enabled status, hold level "L" until starting interrupt processing. If level "H" is set before holding level "L", interrupt processing is correctly started.

(Example releasing "RUN" mode)
$\overline{\text{INT0}}$ interrupt releases HALT state when the RUN mode is on.

```
Address
8203H    LD    (IIMC), 00H          ; selects INT0 interrupt falling edge.
8206H    LD    (INTE0AD), 06H       ; sets interrupt level to "6" for INT0
8209H    EI    5                    ; sets interrupt level to "5" for CPU
820BH    LD    (WDMOD), 00H         ; sets HALT mode to "RUN".
820EH    HALT                       ; halts CPU.
```



(3) Operation

① RUN mode

In the RUN mode, the system clock in the MCU continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction. In the HALT state, an interrupt request is sampled with the falling edge of the "CLK" signal.
Releasing the RUN mode is executed by the external/internal interrupts. (See Table 3.3.4 Halt releasing source and Halt releasing operation.)
Figure 3.3.6 shows the timing for releasing the HALT state by interrupts in the RUN/IDLE2 mode.

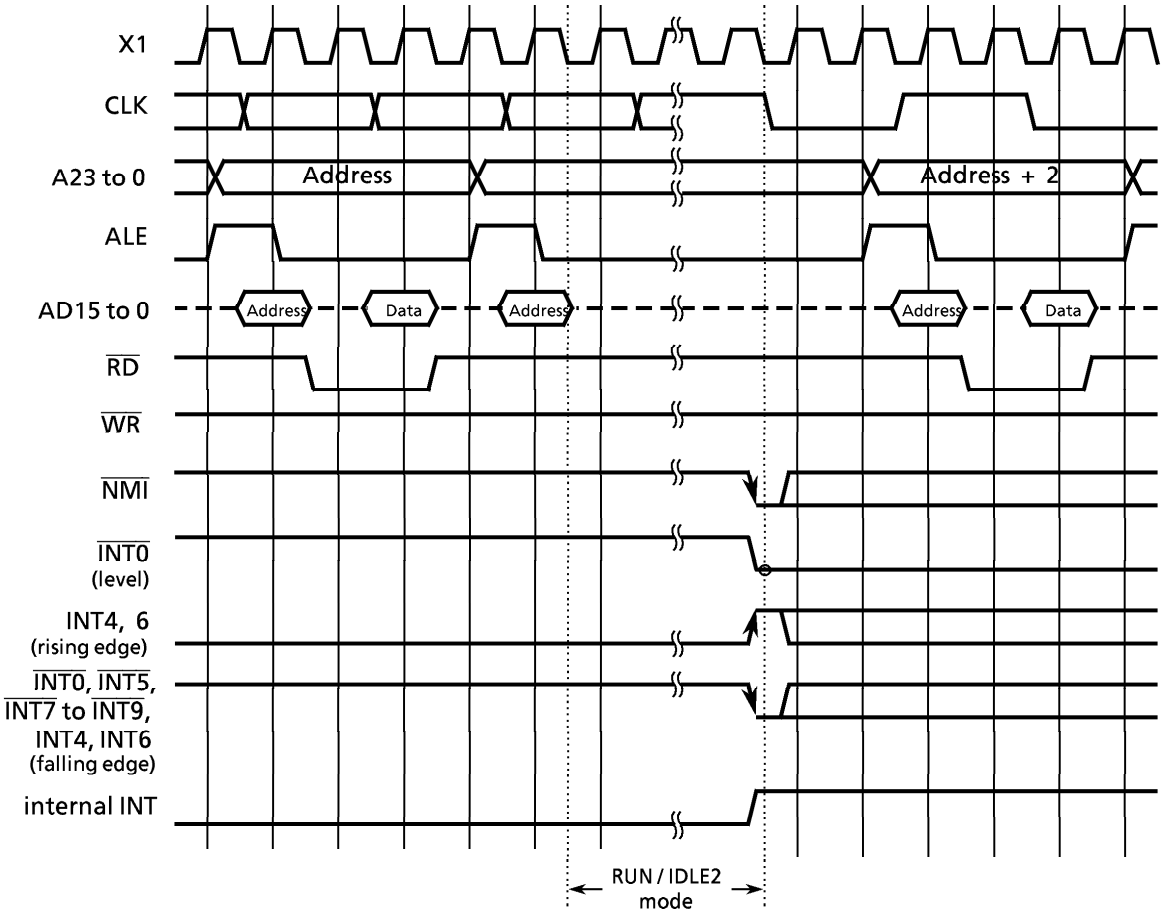


Figure 3.3.6 Timing Chart for Releasing the HALT State by Interrupt in RUN/IDLE2 modes

② IDLE2 mode

In the IDLE2 mode, the system clock is supplied to only specific internal I/O devices, and the CPU stops executing the current instruction.
In the IDLE2 mode, the HALT state is released by an interrupt with the same timing as in the RUN mode. The IDLE2 mode is released by external/internal interrupt, except INTWDT/INTAD interrupts. (See Table 3.3.4 Halt releasing source and Halt releasing operation.)
In the IDLE2 mode, the watchdog timer should be disabled before entering the halt status to prevent the watchdog timer interrupt occurring just after releasing the halt mode.

③ IDLE1 mode

In the IDLE1 mode, only the internal oscillator operates. The system clock in the MCU stops, the CLK pin is fixed at the level "H" in the output enable (CKOCR<CLKEN> = "1").
In the HALT state, and interrupt request is sampled synchronously with the system clock, however the HALT release (restart of operation) is performed synchronously with it.
IDLE1 mode is released by external interrupts (NMI, INT0). (See Table 3.3.4 Halt releasing source and Halt releasing operation.)
When the IDLE1 mode is used, setting TRUN<PRRUN> to "0" to stop 9, 5 bit prescaler before "HALT" instruction reduces the power consumption.

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

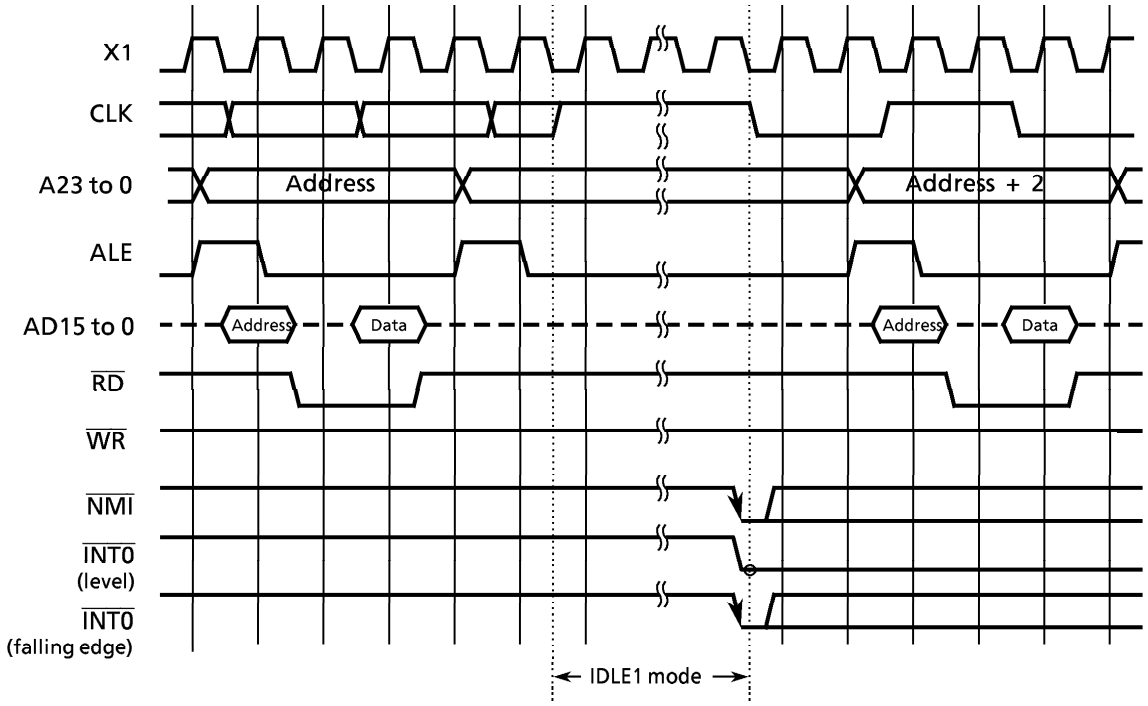


Figure 3.3.7 Timing Chart for Releasing the HALT State by Interrupt in IDLE1 modes

④ STOP mode

The STOP mode is selected to stop all internal circuits including the internal oscillator. The pin status in the STOP mode is depended on setting the watchdog timer mode register WDMOD <DRVE>. (See Figure 3.3.5 for setting WDMOD <DRVE>.) Table 3.3.6 summarizes the state of these pins in the STOP mode.
The STOP mode is released by external interrupts (NMI, INT0). When the STOP mode is released, the system clock is started outputting after warming up timer to get the stabilized oscillation. A warming-up time can be set using WDMOD <WARM>. See the example of warming-up time (Table 3.3.5).

In the system which supplies stable clock generated by an external oscillator, the warming-up time can be reduced using T45CR <WCU>.

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

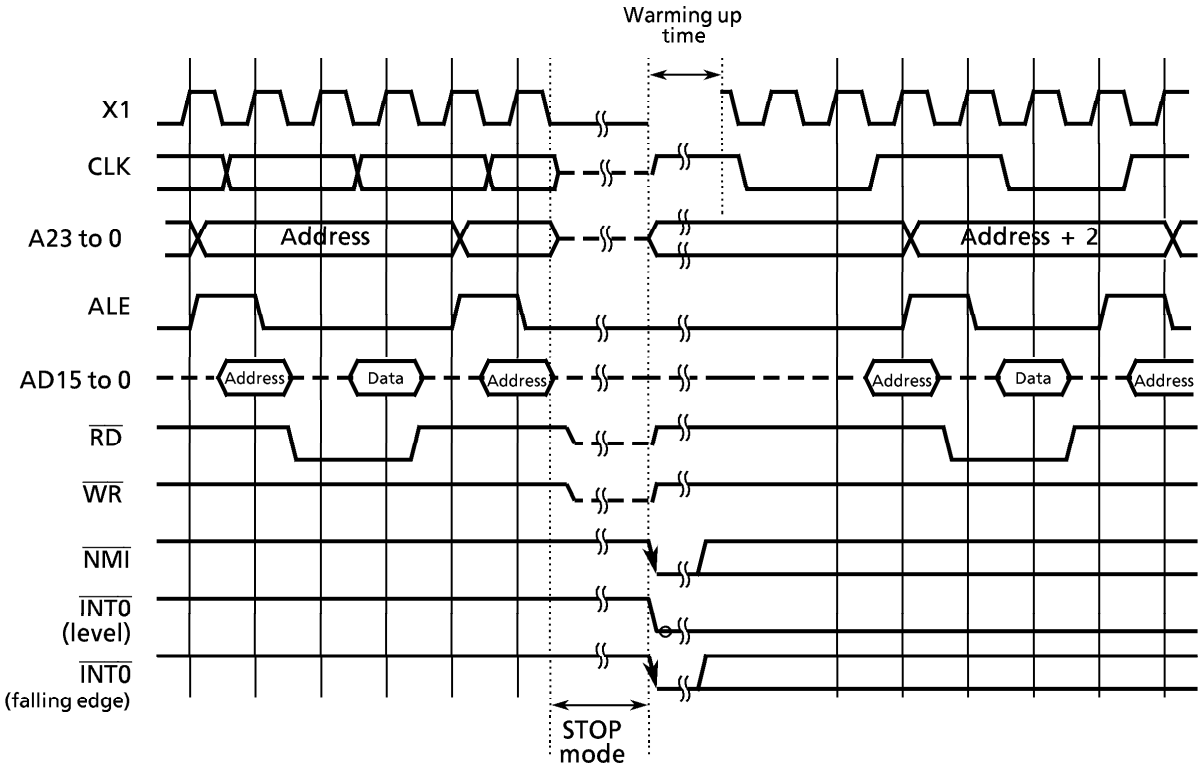


Figure 3.3.8 Timing Chart of HALT Released by Interrupt in STOP Mode

Table 3.3.5 The example of warming-up time after releasing the stop mode

Operation clock after the stop mode	Warming-up time [ms]		Clock
	WDMOD<WARM> = 0	WDMOD<WARM> = 1	
fc	0.8192	3.2768	fc = 20 MHz
fc/2	1.6384	6.5536	
fc/4	3.2768	13.1072	
fc/8	6.5536	26.2144	
fc/16	13.1072	52.4288	

How to calculate the warming-up time

WDMOD <WARM> = "0" : Operation clock after the 2^{14}/STOP mode
WDMOD <WARM> = "1" : Operation clock after the 2^{16}/STOP mode

Setting Example : The STOP mode is released by NMI.

```
Address
         SYSCR0     EQU    006EH
         SYSCR1     EQU    006FH
         WDMOD      EQU    005CH
8FFDH                LD     (SYSCR1), 00H      ; fSYS = fc/2
9000H                RES    4, (WDMOD)         ; Sets Warming Up Time to 2^14/fc
9002H                HALT
```

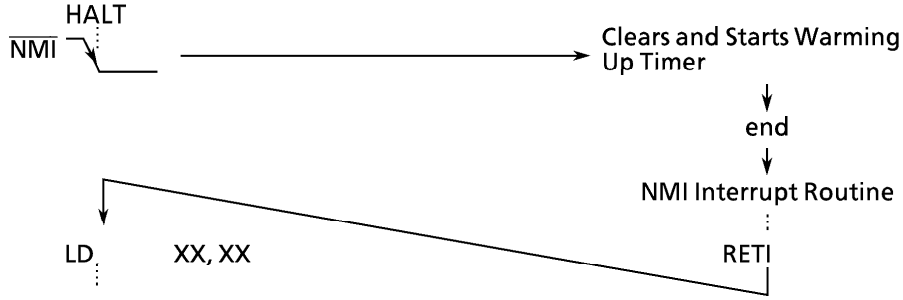


Table 3.3.6 Pin states in STOP mode

Pin Name	I/O	TMP93CS42A	
		DRVE = 0	DRVE = 1
P0	Input mode Output mode AD8~ to 15	▲ – –	▲ Output –
P1	Input mode Output mode / A8 to 15 AD0 to 7	▲ – –	▲ Output –
P2	Input mode Output mode / A0 to 7, A16 to 23	▲ ▲	▲ Output
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	Output	–	Output
P30	Output	Output	Output
P32 to P37	Input mode Output mode	PU* PU*	Input mode Output mode
P40, P41	Input mode Output mode	PU* PU*	Input mode Output mode
P42 (CS2 / CAS2)	Input mode Output mode	PD* PD*	Input mode Output mode
P5	Input	▲	▲
P6	Input mode Input mode Output mode	PU* PU*	Input mode Output mode
P70	Input mode Output mode	0 Input 0 Input	0 Input mode Output mode
P71 to P73	Input mode Output mode	PU* PU*	Input mode Output mode
P80, P82, P83, P84, P86	Input mode Output mode	PU* PU*	Input mode Output mode
P81, P85, P87 ($\overline{INT0}$)	Input mode Output mode	0 Input 0 Input	0 Input mode Output mode
P90 to P95	Input mode Output mode	PU* PU*	Input mode Output mode
PA6 to PA0	Input mode Output mode	– –	
PA7	Input mode Output mode, SCOUT	– –	Input mode Output mode
$\overline{NMI}$	Input	Input	Input
$\overline{INT8}$	Input	0 Input	0 Input
ALE	Output (<ALEEN> = '1')	"L" level output	"L" level output
CLK	Output (<CLKEN> = '1')	–	"H" level output
$\overline{RESET}$	Input		Input
$\overline{EA}$	Input	Input	Input
X1	Input	–	–
X2	Output	"H" level output	"H" level output
P97 to 96	Input mode Output mode	– –	Input Output*

– : Input for input mode / input pin is invalid; output mode / output pin is at high impedance.
0 input : Input gate in operation. Fix input voltage to "L" so that input pin stays constant.
Input : Input gate in operation. Fix input voltage to "L" or "H" so that input pin stays constant.
Output : Output state
Output* : Open-drain output state. Input gate in operation. Set output to "L" or attach pull-up on pin so that the input gate stays content.
PU : Programmable pull-up pin. Fix the pin to avoid through current since the input gate operates when a pull-up pin resistor is not set.
PU* : Programmable pull-up pin. Input gate disable state. No through current even if the pin is set t high impedance.
PD* : Programmable pull-down pin. Input gate disable state. No through current even if the pin is set t high impedance.
▲ : When HALT instruction is executed and CPU stops at the address of the port register, input gate in operation. Fix the pin to avoid through current and change the program. In the other cases, input for input mode is invalid, output mode is at high impedance.

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

3.4 Interrupts

Interrupts are controlled by the CPU interrupt mask register SR<IFF2-0> and the built-in interrupt controller.
interrupt sources :

- Interrupts from the CPU ··· 9 sources
  (Software interrupts, and Illegal (undefined) instruction execution)
- Interrupts from external pins ($\overline{NMI}$, $\overline{INT0}$ and INT4 to 9) ··· 8 sources
- Interrupts from built-in I/Os ··· 12 sources

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority (variable) can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority of 7.
When an interrupt is generated, the interrupt controller sends the value of the priority of the interupt source to the CPU. When more than one interupt 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. However software interrupts and illegal instruction execution interrupts are not compared with <IFF2 to 0>, the interrupt is processed. 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. When "EI" or "EI 0" is programmed, maskable interrupts with a priority of 1 or greater, and non-maskable interrupts are set in the interrupt instruction. (in the same way as the EI 1) 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 maskable interrupts to be accepted. The EI instruction becomes effective immediately after execution. (With the TLCS-90, the EI instruction becomes effective after execution of the subsequent instruction.)
In addition to the general-purpose interrupt processing mode described above, there is also a 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.

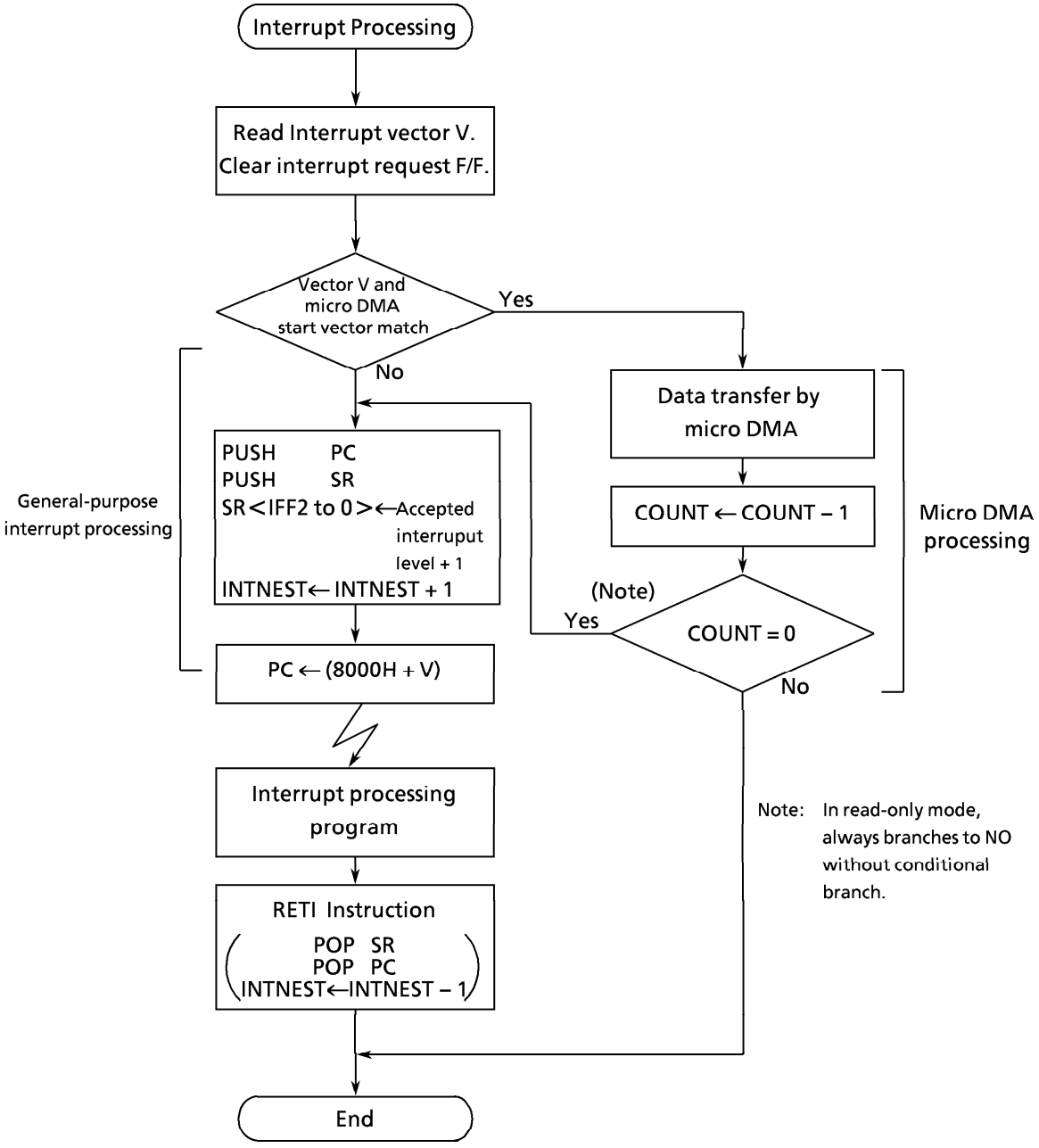


Figure 3.4.1 Interrupt Processing Flowchart

3.4.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows. In the software interrupts or the illegal instruction execution interrupts from CPU, the following (1) and (3) are not executed.

(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, then clears the interrupt request.
The default priority is fixed as follows: the smaller the vector value, the higher the priority.

(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 8000H + interrupt vector, then starts the interrupt processing routine.

The following diagram shows all the above processing state number.

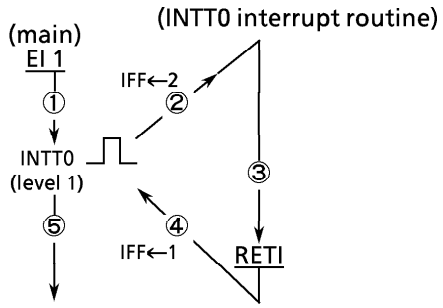
Bus Width of Stack Area	Bus Width of Interrupt Vector Area	Interrupt processing state number	
		MAX mode	MIN mode
8	8	35	31
	16	31	27
16	8	29	27
	16	25	23

To complete 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.

The CPU does not accept an interrupt request of the same level as that of the interrupt being processed. (Non-maskable interrupts can be accepted, causing interrupt processing to nest.)

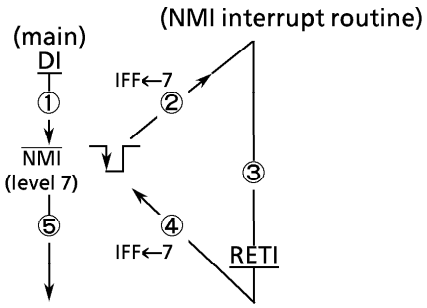
Resetting initializes the CPU mask registers <IFF2 to 0> to 7; therefore, maskable interrupts are disabled.

(1) Maskable interrupt



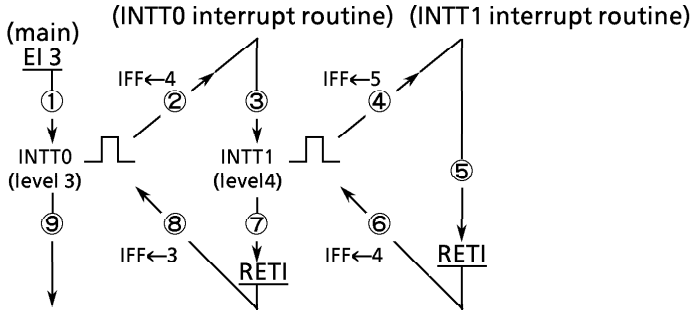
During execution of the main program, the CPU accepts an interrupt request. The CPU increments the IFF so that the interrupts of level 1 are not accepted during processing the interrupt routine.

(2) Non-maskable interrupt



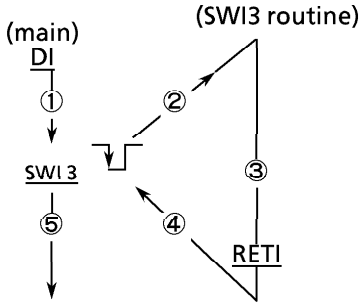
DI instruction is executed in the main program, so that the interrupts of only level 7 are accepted. The CPU does not increment the IFF even if the CPU accepts an interrupt request of level 7.

(3) Interrupt nesting



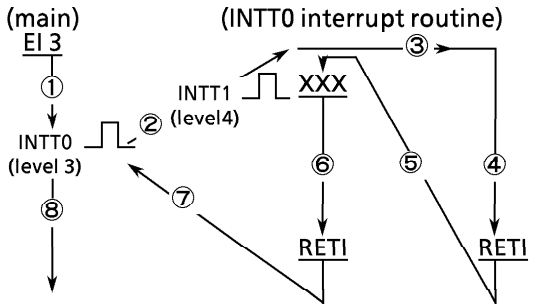
During processing the interrupts of level 3, the IFF is set to 4. When an interrupt with a level higher than level 4 is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

(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 008000H to 0080FFH (256 bytes) of the TMP93CS42A are assigned for interrupt vector area.

Table 3.4.1 TMP93CS42A Interrupt Table

Default priority	Type	Interrupt source	Vector value "V"	Address refer to vector	Mico DMA start vector
1		Reset, or SWI0 instruction	0 0 0 0 H	8 0 0 0 H	–
2		SWI 1 instruction	0 0 0 4 H	8 0 0 4 H	–
3		Illegal instruction, or SWI2	0 0 0 8 H	8 0 0 8 H	–
4	Non-	SWI 3 instruction	0 0 0 C H	8 0 0 C H	–
5	maskable	SWI 4 instruction	0 0 1 0 H	8 0 1 0 H	–
6		SWI 5 instruction	0 0 1 4 H	8 0 1 4 H	–
7		SWI 6 instruction	0 0 1 8 H	8 0 1 8 H	–
8		SWI 7 instruction	0 0 1 C H	8 0 1 C H	–
9		NMI : $\overline{\text{NMI}}$ pin	0 0 2 0 H	8 0 2 0 H	08H
10		INTWD : Watchdog timer	0 0 2 4 H	8 0 2 4 H	09H
11		INT0 : $\overline{\text{INT0}}$ pin input	0 0 2 8 H	8 0 2 8 H	0AH
12		INT4 : $\overline{\text{INT4}}$ pin input	0 0 2 C H	8 0 2 C H	0BH
13		INT5 : $\overline{\text{INT5}}$ pin input	0 0 3 0 H	8 0 3 0 H	0CH
14		INT6 : $\overline{\text{INT6}}$ pin input	0 0 3 4 H	8 0 3 4 H	0DH
15		INT7 : $\overline{\text{INT7}}$ pin input	0 0 3 8 H	8 0 3 8 H	0EH
–		(Reserved)	0 0 3 C H	8 0 3 C H	–
16		INTT0 : 8-bit timer0	0 0 4 0 H	8 0 4 0 H	10H
17		INTT1 : 8-bit timer1	0 0 4 4 H	8 0 4 4 H	11H
18		INT8 : $\overline{\text{INT8}}$ pin input	0 0 4 8 H	8 0 4 8 H	12H
19		INT9 : $\overline{\text{INT9}}$ pin input	0 0 4 C H	8 0 4 C H	13H
20		INTTR4 : 16-bit timer4 (TREG4)	0 0 5 0 H	8 0 5 0 H	14H
21	Maskable	INTTR5 : 16-bit timer4 (TREG5)	0 0 5 4 H	8 0 5 4 H	15H
22		INTTR6 : 16-bit timer5 (TREG6)	0 0 5 8 H	8 0 5 8 H	16H
23		INTTR7 : 16-bit timer5 (TREG7)	0 0 5 C H	8 0 5 C H	17H
24		INTRX0 : Serial receive (Channel.0)	0 0 6 0 H	8 0 6 0 H	18H
25		INTTX0 : Serial send (Channel.0)	0 0 6 4 H	8 0 6 4 H	19H
26		INTRX1 : Serial receive (Channel.1)	0 0 6 8 H	8 0 6 8 H	1AH
27		INTTX1 : Serial send (Channel.1)	0 0 6 C H	8 0 6 C H	1BH
28		INTAD : AD conversion completion	0 0 7 0 H	8 0 7 0 H	1CH
–		(Reserved)	0 0 7 4 H	8 0 7 4 H	–
to		to	to	to	to
–		(Reserved)	0 0 F C H	8 0 F C H	–

Setting to Reset / Interrupt Vector

① Reset Vector

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

The vector base addresses are depended on the products.

Type No.	Vector base address	PC setting sequence after reset	Notes
TMP93CS42A/PS42A	008000H	PC (7 to 0) ← address 8000H PC (15 to 8) ← address 8001H PC (23 to 16) ← address 8002H	P27 to 20/A23 to 16 pins input ports with pull-down due to reset. The logic data is "00H". When Port 2 is used as A23 to 16 pins to access the program ROM, set PC (23 to 16) to "00H" and the reset vector to "0000H to FFFFH". (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)

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

```
ORG     8000H
DL      008100H         ; Reset = 8100H

ORG     8020H
DL      009ABCH         ; NMI = 9ABCH

ORG     8070H
DL      123456H         ; INTAD = 123456H

ORG     8100H
LD      A, B
        :
ORG     9ABCH
LD      B, C
        :
ORG     123456H
LD      C, A
        :
```

(cf)

ORG, DL are the Assembler Directive.

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

3.4.2 Micro DMA

In addition to the conventional interrupt processing, the TMP93CS42A 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 micro DMAcan process at very high speed compared with the TLCS-90 micro DMA 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 set in the interrupt controller. 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. In read-only mode, which is provided for DRAM refresh, the value in the counter is ignored and dummy read is repeated.
32-bit control registers are used for setting transfer source / destination addresses. However, the TMP93CS42A 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 DMAprocessing.
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. Reset the transfer counter in the processing routine of the general-purpose interrupt.
Interrupt sources processed by micro DMA processing are 20 sources 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 (the other mode except the Read-only mode is same as this) (Condition : MAX mode, 16 bit Bus width for 16M Byte, 0 wait)

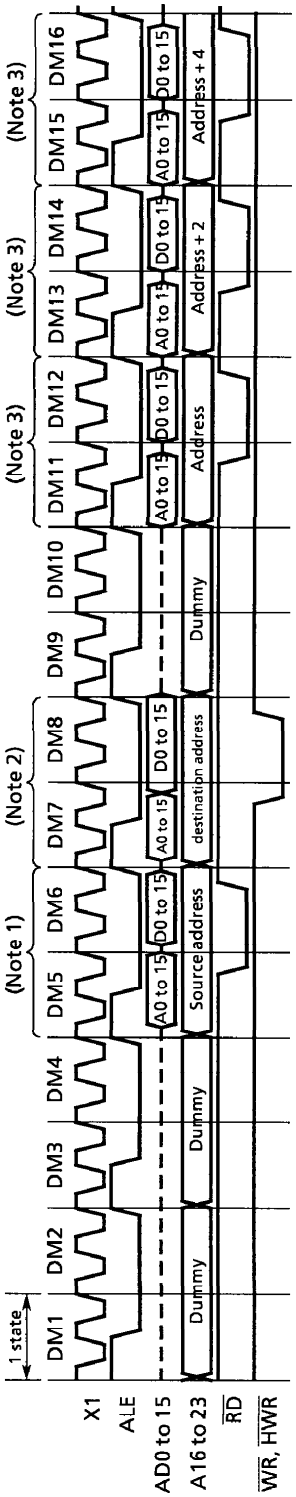
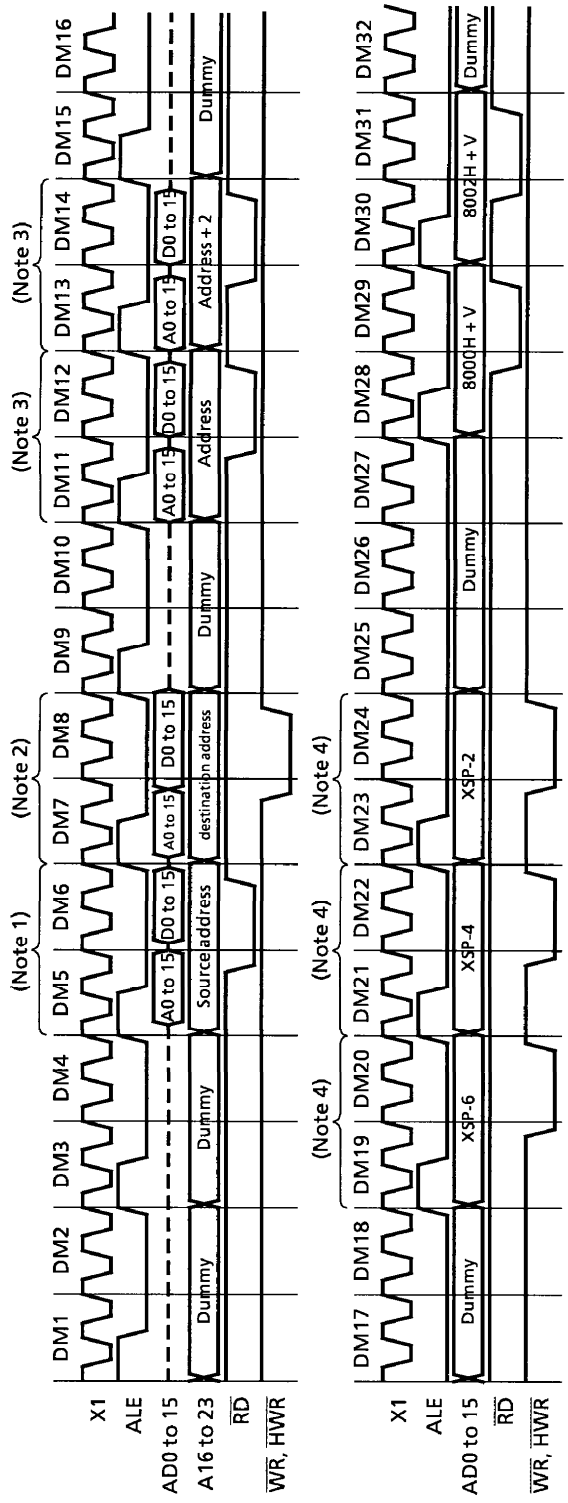


Figure 3.4.2 Micro DMA cycle (COUNT≠0)

Note 1: This is added 2 states the case of the bus width of source address area is 8 bits or the address starts from an odd number.

Note 2: This added 2 states the case of the bus width of destination address area is 8 bits or the address starts from an odd number.

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



Note 1: This is added 2 states the case of the bus width of source address area is 8 bits or the address starts from an odd number.

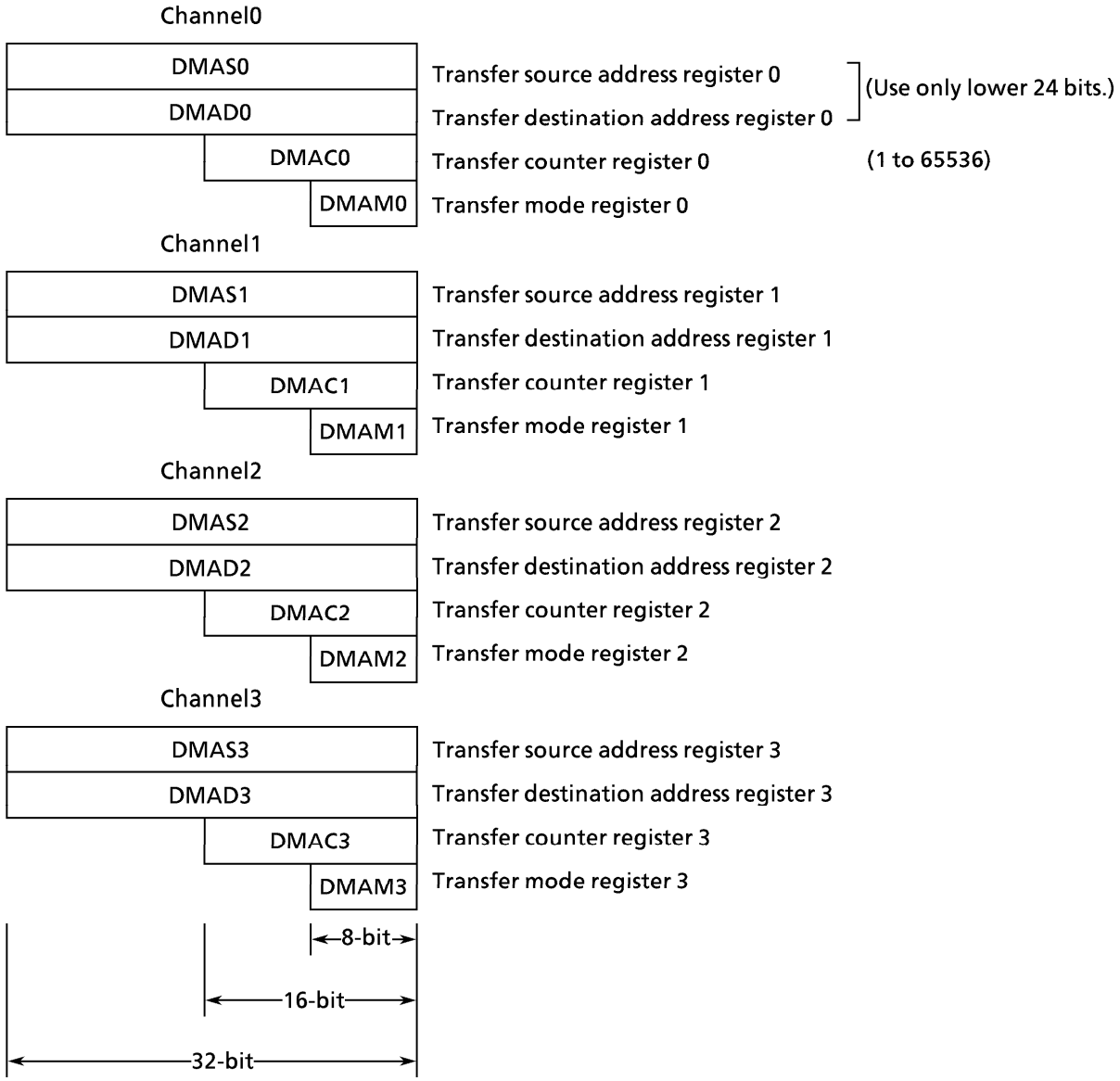
Note 2: This added 2 states the case of the bus width of destination address area is 8 bits or the address starts from an odd number.

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

Note 4: This is added 2 states the case of the bus width of stack address area is 8 bits or the stack pointer starts from an odd number.

Figure 3.4.3 icro DMA cycle (COUNT = 0)

(2) Register configuration (CPU control register)

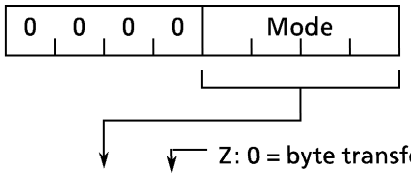


These Control Register can not be set only "LDC cr, r" instruction.

(eg.)

```
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



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

					execution time
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	0	Read-only mode ..... for DRAM refresh Dummy← (DMASn) ; Reads 4 bytes. DMASn←DMASn + 4 ; Increments lower word only. DMACn←DMACn – 1	14 states (1.4 μs)
1	0	1	1	Counter mode ..... for interrupt counter DMASn←DMASn + 1 DMACn←DMACn – 1 if DMACn = 0 then INT.	11 states (1.1 μs)

Note 1 : n :corresponds to high-speed μDMA channels 0 to 3.
DMADn + / DMASn + : Post-increment (Increments register value after transfer.)
DMADn – / DMASn – : Post-decrement (Decrement register value after transfer.)

Note 2 : Execution time : When setting source address/destination address area to 16-bit bus, 0WAIT.

Note 3 : Do not use the codes other than the above mentioned codes for transfer mode register.

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

※ Clock Condition

System Clock: fc
Clock Gear : 1 (fc)

When the hardware configuration is as follows:
DRAM mapping size: =1 MB
DRAM data bus size: =8 bits
DRAM mapping address range: =100000H to 1FFFFFH

Set the following registers first; refresh is performed automatically.

① Register initial value setting

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

② Timer setting
Set the timers so that interrupts are generated at intervals of 62.5 μs or less.

③ Interrupt controller setting
Set the timer interrupt mask higher than the other interrupts mask. Write the above timer interrupt vector value in the High-Speed micro DMA start vector register, DMA0V.

(Operation description)

The DRAM data bus is an 8-bit bus and the micro DMA is in read-only mode (4 byte), so refresh is performed for four times per interrupt.
When a 512 refresh / 8ms DRAM is connected, DRAM refresh is performed sufficiently if the microDMA is started every 15.625 μs × 4 = 62.5 μs or less, since the timing is15.625 μs / refresh.

(Overhead)

Each processing time by the micro DMA is 1.8 μs (18 states) at 20 MHz with an 8-bit data bus.
In the above example, the micro DMA is started every 62.5 μs, 1.8 μs / 62.5 μs = 0.0288; thus, the overhead is 2.88 %.

(Note)

When the Bus is released which must wait to accept the interrupt, DRAM refresh is not performed because of the micro DMA is generated by an interrupt.

3.4.3 Interrupt Controller

Figure 3.4.4 is a block diagram of the interrupt circuits. The left half of the diagram shows the interrupt controller; the right half includes the CPU interrupt request signal circuit and the HALT release signal circuit.
Each interrupt channel (total of 20 channels) in the interrupt controller has an interrupt request flag, interrupt priority setting register, and a register for storing the micro DMA start vector. The interrupt request flag is used to latch interrupt requests from peripheral devices.

The flag is cleared to 0 at the following conditions.

- at reset
- when the CPU reads the interrupt vector after acceptance of interrupt
- when the CPU executes an instruction that clears the interrupt of that channel (writes 0 in <IxxC> of the interrupt priority setting register).

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

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

The status of the interrupt request flip-flop is detected by reading the clear bit. Detects whether there is an interrupt request for an interrupt channel.
The interrupt priority can be set by writing the priority in the interrupt priority setting register (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 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. 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.

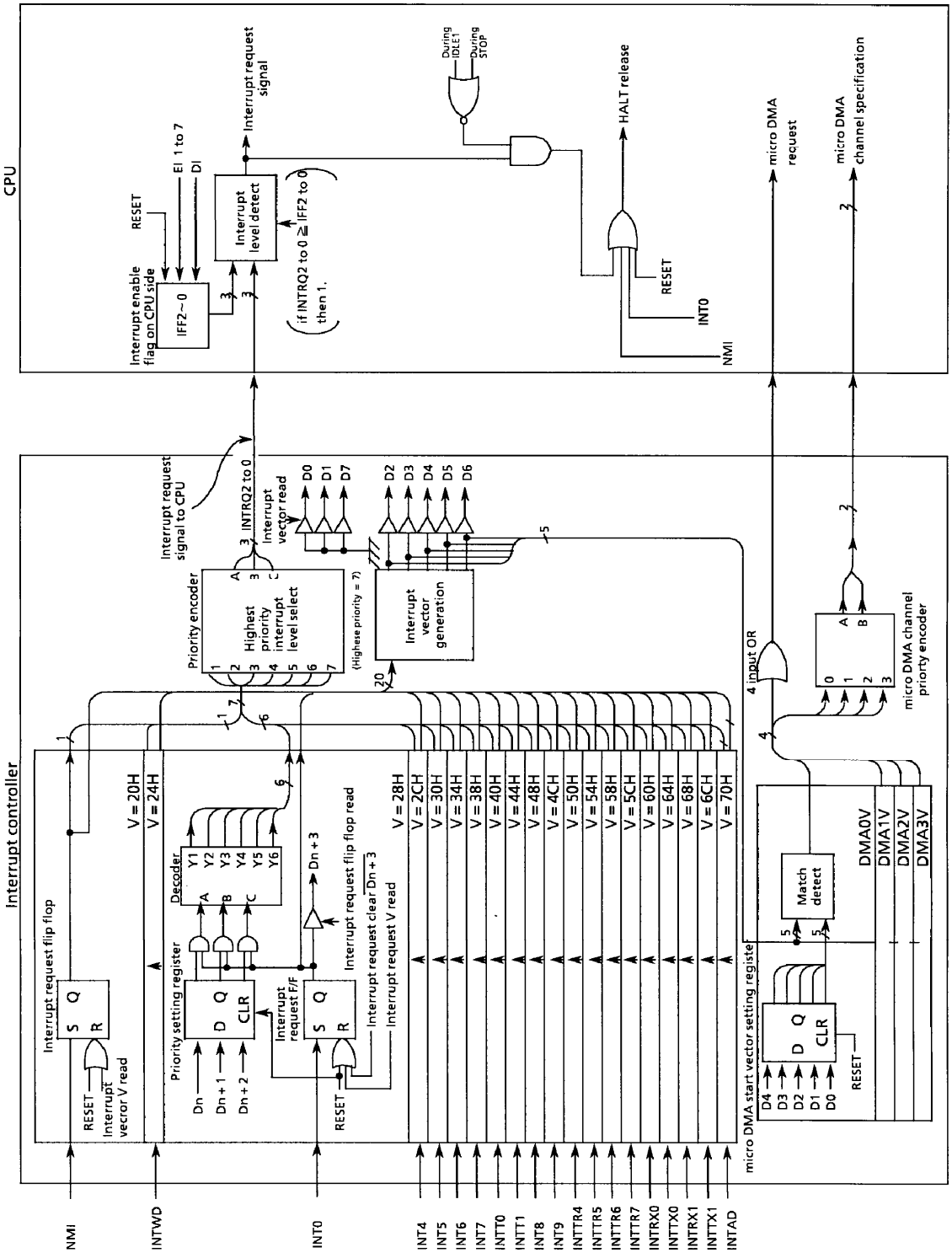


Figure 3.4.4 Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

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	
INTE89	0074H	INT9				INT8				
		I9C	I9M2	I9M1	I9M0	I8C	I8M2	I8M1	I8M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET54	0075H	INTTR5 (TREG5)				INTTR4 (TREG4)				
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET76	0076H	INTTR7 (TREG7)				INTTR6 (TREG6)				
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES0	0077H	INTTX0				INTRX0				
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES1	0078H	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	

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

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

Note 1 : Read-modify-write is prohibited.

Note 2 : Note about clearing interrupt request flag
The interrupt request flag of INTAD, INTRX0, INTRX1 are not cleared by writing "0" to IxxC because of they are level interrupts. They can be cleared only by resetting or reading the conversion value or the receive buffer.

Figure 3.4.5 Interrupt priority setting register

(2) External interrupt control

Interrupt Input Mode Control Register

IIMC (007BH)

	7	6	5	4	3	2	1	0
Bit symbol						I0IE	I0LE	NMIREE
Read/Write						W	W	W
After reset						0	0	0
Function						1: $\overline{INT0}$ input enable	0: $\overline{INT0}$ edge mode 1: $\overline{INT0}$ level mode	1: Can be accepted in $\overline{NMI}$ rising edge.

$\overline{INT0}$ input enable (note 1)

0	$\overline{INT0}$ disable (P87 function only)
1	$\overline{INT0}$ Input enable

$\overline{NMI}$ rising edge enable

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

$\overline{INT0}$ level enable

0	Rising edge detect interrupt
1	High level interrupt

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

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

Execution example:

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

Note3 : Read-modify-write is prohibited.

Note4 : IIMC<bit7 to 3> is always read as "1".

Note5 : See Electrical characteristics in section 4 for external interrupt input pulse.

Figure 3.4.6 Interrupt Input Mode Control Register

Table 3.4.2 Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
$\overline{NMI}$	—	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
$\overline{INT0}$	P87	Falling edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 1
INT4	P80	Rising edge	T4MOD<CAP12M1, 0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD<CAP12M1, 0> = 1, 0
$\overline{INT5}$	P81	Falling edge	———
INT6	P84	Rising edge	T5MOD<CAP34M1, 0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T5MOD<CAP34M1, 0> = 1, 0
$\overline{INT7}$	P85	Falling edge	———
$\overline{INT8}$	—	Falling edge	———
$\overline{INT9}$	P70	Falling edge	———

(3) Micro DMA start vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the bit 2 to 6 interrupt vector with each channel's micro DMA start vector. When both match, the interrupt is processed in micro DMA mode for the channel whose value matched.
If the interrupt vector matches more than two channel, the channel with the lower channel number has a higher priority.

Micro DMA 0 Start Vector

DMA0V (007CH)

	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 DMA 1 Start Vector

DMA1V (007DH)

	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 DMA 2 Start Vector

DMA2V (007EH)

	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 DMA 3 Start Vector

DMA3V (007FH)

	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.							

Note : Read-modify-write is not possible for DMA0V to DMA3V.

(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 start the interrupt processing from the address "8028H".

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 to 0> by execution of POR SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

3.5 Functions of Ports

The TMP93C42A has 80 bits for I/O ports.
These port pins have I/O functions for the built-in CPU and internal I/Os as well as general-purpose I/O port functions. Table 3.5.1 lists the function of each port pin. Table 3.5.2 lists I/O registers and specification.

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

Table 3.5.1 Functions of Ports

Port name	Pin name	Number of pins	Direction	R	Direction setting unit	Pin name for built-in function
Port0	P00 to P07	8	I/O	–	Bit	AD0 to AD7
Port1	P10 to P17	8	I/O	–	Bit	AD8 to AD15 / A8 to A15
Port2	P20 to P27	8	I/O	↓	Bit	A0 to A7 / A16 to A23
Port3	P30 (72 pin)	1	Output	–	(Fixed)	$\overline{RD}$
	P30 (61 pin)	1	Output	–	(Fixed)	–
	P31	1	Output	–	(Fixed)	$\overline{WR}$
	P32	1	I/O	↑	Bit	$\overline{HWR}$
	P33	1	I/O	↑	Bit	$\overline{WAIT}$
	P34	1	I/O	↑	Bit	$\overline{BUSRQ}$
	P35	1	I/O	↑	Bit	$\overline{BUSAK}$
	P36	1	I/O	↑	Bit	R/$\overline{W}$
	P37	1	I/O	↑	Bit	$\overline{RAS}$
Port4	P40	1	I/O	↑	Bit	$\overline{CS0}$ / $\overline{CAS0}$
	P41	1	I/O	↑	Bit	$\overline{CS1}$ / $\overline{CAS1}$
	P42	1	I/O	↓	Bit	$\overline{CS2}$ / $\overline{CAS2}$
Port5	P50 to P54	5	Input	–	(Fixed)	AN0 to AN4
	P55	1	Input	–	(Fixed)	RXD0
	P56	1	Input	–	(Fixed)	$\overline{CTS0}$
	P57	1	Input	–	(Fixed)	RXD1
Port6	P60 to P67	8	I/O	↑	Bit	–
Port7	P70	1	I/O	–	Bit	TI0/$\overline{INT9}$
	P71	1	I/O	↑	Bit	TO1
	P72	1	I/O	↑	Bit	TO2
	P73	1	I/O	↑	Bit	TO3
Port8	P80	1	I/O	↑	Bit	TI4 / $\overline{INT4}$
	P81	1	I/O	–	Bit	TI5 / $\overline{INT5}$
	P82	1	I/O	↑	Bit	TO4
	P83	1	I/O	↑	Bit	TO5
	P84	1	I/O	↑	Bit	TI6 / $\overline{INT6}$
	P85	1	I/O	–	Bit	TI7 / $\overline{INT7}$
	P86	1	I/O	↑	Bit	TO6
	P87	1	I/O	–	Bit	$\overline{INT0}$
Port9	P90	1	I/O	↑	Bit	TXD0
	P91	1	I/O	↑	Bit	–
	P92	1	I/O	↑	Bit	–
	P93	1	I/O	↑	Bit	TXD1
	P94	1	I/O	↑	Bit	–
	P95	1	I/O	↑	Bit	SCLK1
	P96	1	I/O	–	Bit	–
	P97	1	I/O	–	Bit	–
PortA	PA0 to PA6	7	I/O	–	Bit	
	PA7	1	I/O	–	Bit	SCOUT (PA7)

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	P0 (0 to 7)	Input port	×	0	None
		Output port	×	1	
		AD (0 to 7) bus	×	×	
Port 1	P1 (0 to 7)	Input port	×	0	0
		Output port	×	1	0
		AD (8 to 15) bus (Note 1)	×	0	1
		A (8 to 15) output (Note 1)	×	1	1
Port 2	P2 (0 to 7)	Input port (without PD)	1	0	0
		Input port (with PD)	0	0	0
		Output port	×	1	0
		A (0 to 7) Output	1	0	1
		A (16 to 23) output	1	1	1
Port 3	P30 (Pin No. 72)	Output port	×	None	0
		Outputs $\overline{RD}$ only when accessing external space	×		1
	P30 (Pin No. 61)	Output Port	×	None	×
	P31	Output port	×	None	0
		Outputs $\overline{WR}$ only when accessing external space	×		1
	P3 (2 to 7)	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P32	$\overline{HWR}$ Output	×	1	1
	P33	$\overline{WAIT}$ Input (without PU)	0	0	None
		$\overline{WAIT}$ Input (with PU)	1	0	
	P34	$\overline{BUSRQ}$ Input (without PU)	0	0	1
		$\overline{BUSRQ}$ Input (with PU)	1	0	1
	P35	$\overline{BUSAK}$ Output	×	1	1
	P36	R/$\overline{W}$ Output	×	1	1
	P37	$\overline{RAS}$ Output	×	1	1
Port 4	P4 (0 to 1)	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P42	Input port (without PD)	1	0	0
		Input port (with PD)	0	0	0
		Output port	×	1	0
	P40	$\overline{CS0}$ Output (Note 2)	×	1	1
	P41	$\overline{CS1}$ Output (Note 2)	×	1	1
	P42	$\overline{CS2}$ Output (Note 2)	×	1	1
Port 5	P5 (0 to 7)	Input port	×	None	
	P5 (0 to 4)	AN (0 to 4) Input (Note 3)	×		
	P55	RXD0 Input	×		
	P56	$\overline{CTS0}$ Input	×		
	P57	RXD1 Input	×		
Port 6	P6 (0 to 7)	Input port (without PU)	0	0	None
		Input port (with PU)	1	0	
		Output port	×	1	

Note 1 : This function is changed by chip select/wait controller.
Note 2 : CS/WAIT control register BnCS<BnCAS> selects the wave form output from P40 to 42 pins, $\overline{CS0}$ to $\overline{CS2}$ or $\overline{CAS0}$ to $\overline{CAS2}$.
Note 3 : The channel for AD input is selected by ADMOD2<ADCHn> (n = 0 to 2).

Table 3.5.2 I/O registers and specification (2/2)

X : Don't care

Port	Name	Specification	I/O register		
			Pn	PnCR	PnFC
Port 7	P7 (1 to 3)	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P70	TI0/$\overline{\text{INT9}}$ Input (without PU)	×	0	None
		Input port (with PU)	×	0	
		Output port	×	1	
	P71	TO1Output port	×	1	1
	P72	TO2Output port	×	1	1
	P73	TO3Output port	×	1	1
Port 8	P8 (0 to 7)	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P80	TI4/INT4 Input (without PU)	0	0	None
		TI4/INT4 Input (with PU)	1	0	
	P81	TI5/$\overline{\text{INT5}}$ Input (without PU)	×	0	None
	P84	TI6/INT6 Input (without PU)	0	0	None
		TI6/INT6 Input (with PU)	1	0	
	P85	TI7/$\overline{\text{INT7}}$ Input (without PU)	×	0	None
	P82	TO4 Output	×	1	1
	P83	TO5 Output	×	1	1
	P86	TO6 Output	×	1	1
	P87 (Note 4)	$\overline{\text{INT0}}$ Input (without PU)	×	0	None
Port 9	P9 (0 to 5)	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P90	TXD0 Output	×	1	1
	P93	TXD1 Output	×	1	1
	P95	SCLK1 Output	×	1	1
		SCLK1 Input (without PU)	0	0	0
		SCLK1 Input (with PU)	1	0	0
	P9 (6 to 7)	Input port	×	0	None
		Output port (Note 5)	×	1	
Port A	PA (0 to 7)	Input port	×	0	None
		Output port	×	1	
	PA7	SCOUTOutput port (Note 6)	×	1	

Note 4 : Using P87 pin as $\overline{\text{INT0}}$, IIMC register has to be set enable interrupt.
Note 5 : Using P96/P97 as output port, Output is through the open-drain buffer.
Note 6 : Using PA7 as SCOUT, PAFC register has to be written suitable value.

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

※ Note about the Bus Release and programmable pull-up / down I/O ports.

When the bus is released ($\overline{BUSAK}$ = "0"), the output buffer of AD0 to 15, A0 to 23, control signal ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$ / $\overline{CAS0}$ to $\overline{CS2}$ / $\overline{CAS2}$) is off and their state become high-impeadance.
However, the output of built-in programmable pull up / down resistors are kept before the bus is released. These programmable pull up / down resistors can be selected ON/OFF by programmable when they are used as the input ports.
The case of they are used as the output ports, they can not be selected ON/OFF by programmable.
Table 3.5.3 shows the pin state when the bus is released.

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

Pin Name	The Pin State (when the bus is released)	
	Used as the port	Used as the function
P00 to P07 (AD0 to AD7) P10 to P17 (AD8 to 15/A8 to 15)	The state is not changed. (do not become to high-impedance (High-z).)	become high-impedance (High-z).
P30 ($\overline{RD}$: Pin No. 72) P31 ($\overline{WR}$)	becomes high-impedance (High-z).	becomes high-impedance (High-z).
P30 (Pin No. 61)	The state is not Changed. (do not become to high-impedance (High-z).)	———
P32 ($\overline{HWR}$) P37($\overline{RAS}$)	The output buffer is OFF. The programmable pull up resistor is ON the case of only the output latch is equal to "1".	The output buffer is OFF. The programmable pull up resistor is ON irrespective of the output latch.
P36 (R/$\overline{W}$) P40 ($\overline{CS0}$ / $\overline{CAS0}$) P41 ($\overline{CS1}$ / $\overline{CAS1}$)	The output buffer is OFF. The programmable pull up resistor is ON the case of only the output latch is equal to "1".	The output buffer is OFF. The programmable pull up resistor is ON irrespective of the output latch.
P42 ($\overline{CS2}$ / $\overline{CAS2}$)	The output buffer is OFF. The programmable pull down resistor is ON the case of only the output latch is equal to "0".	The output buffer is OFF. The programmable pull down resistor is undefined.
P20-P27 (A16 to 23)	The state is not Changed. (do not become to high-impedance (High-z).)	The output buffer is OFF. The programmable pull down resistor is ON the case of only the output latch is equal to "0".

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

When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate.
So, the watchdog timer also continues to run. Therefore, be careful about bus releasing time and setting the detection time of the WDT.

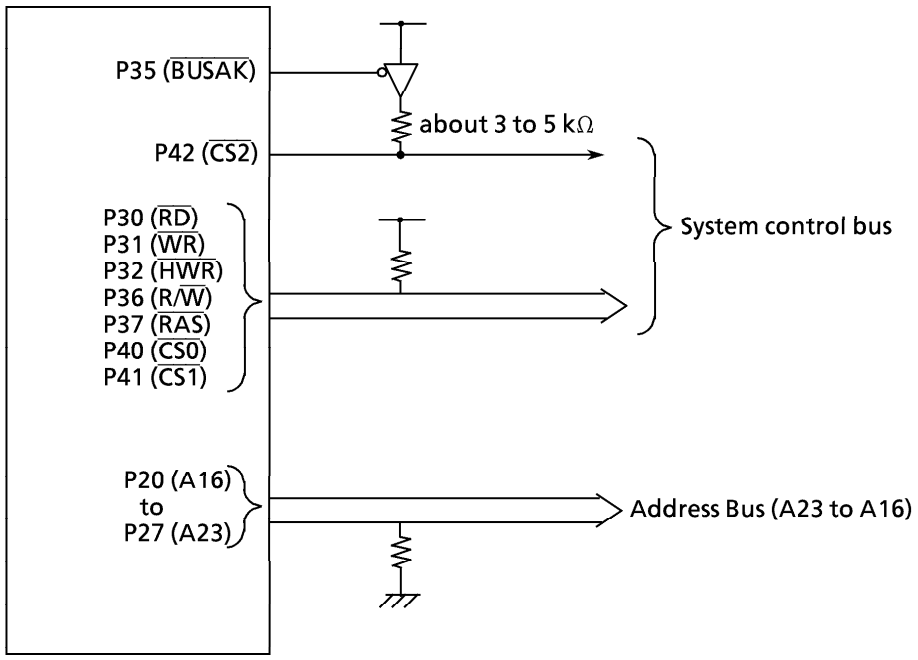


Figure 3.5.1 Example of the interface circuit
(The case of using bus releasing function)

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

Resetting sets P30 ($\overline{RD}$), P31 ($\overline{WR}$) to output, P40 ($\overline{CS0}$), P41 ($\overline{CS1}$), P32 ($\overline{HWR}$), P36 (R/$\overline{W}$), P37 ($\overline{RAS}$), and P35 ($\overline{BUSAK}$) to input with pull up resistor, P42 ($\overline{CS2}$) and P20 to 27 (A16 to 23) to input with pull down resistor.
The above circuit is necessary to fix the signal level after reset because of the external pull up resistor collisions with the internal pull down resistor.
The value of this external pull up resistor must be 3 to 5 kΩ. (the value of the internal pull down resistor is about 50 to 150 kΩ)
P20 to P27 (A16 to 23) also needs circuit like P42 ($\overline{CS2}$) to fix the signal level.
But for the P20 to 27 (A16 to 23) which does not have means "L" is active, add pull down directly like above circuit.

3.5.1 Port 0 (P00 to P07)

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

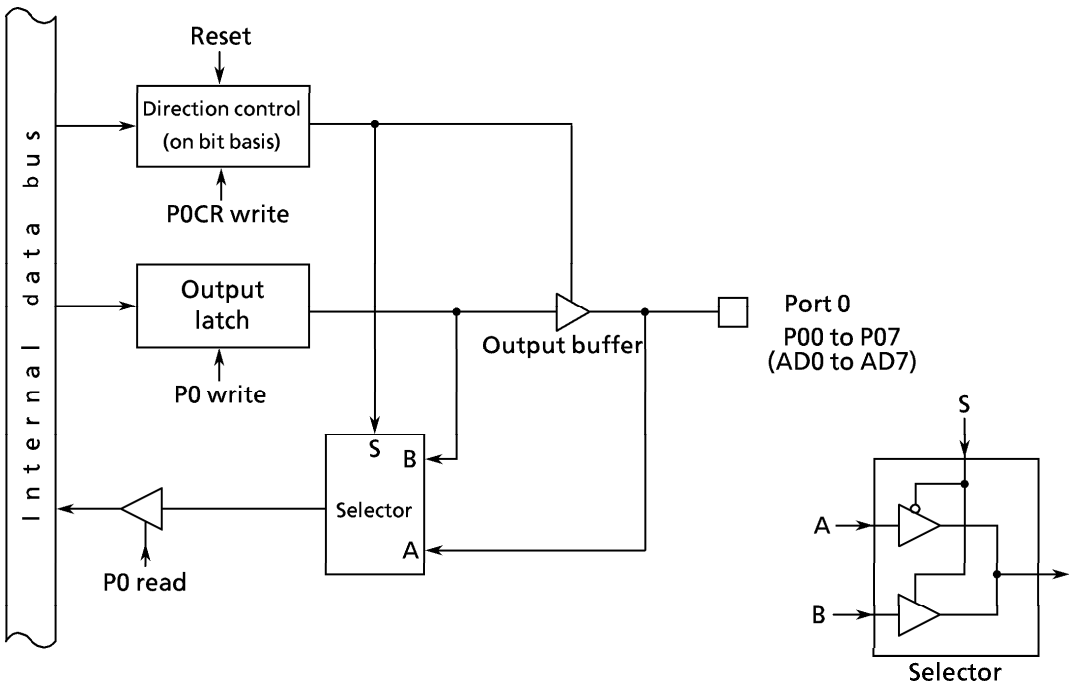


Figure 3.5.2 Port 0

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR and function register P1FC. Resetting resets all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets Port 1 to input mode.
In addition to functioning as a general-purpose I/O port, Port 1 also functions as an address data bus (AD8 to 15) or an address bus (A8 to 15).

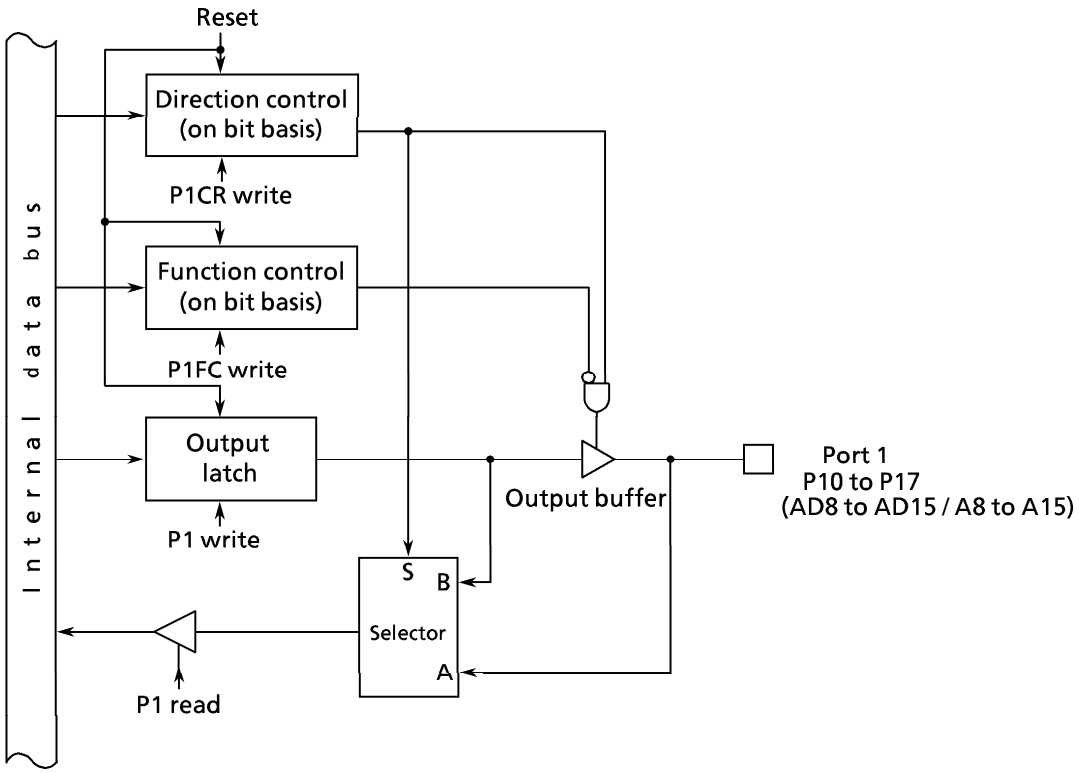


Figure 3.5.3 Port 1

Port 0 Register

P0 (0000H)	7	6	5	4	3	2	1	0
Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
Read/Write	R / W							
After reset	Input mode (Output latch register becomes undefined.)							

Port 0 Control Register

P0CR (0002H)	7	6	5	4	3	2	1	0
Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT (At external access, Port 0 becomes AD7 to 0 and P0CR is cleared to 0.)							

→ Port 0 I/O setting

0	Input
1	Output

Port 1 Register

P1 (0001H)	7	6	5	4	3	2	1	0
Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R / W							
After reset	Input mode (Output latch register is cleared to "0".)							

Port 1 Control Register

P1CR (0004H)	7	6	5	4	3	2	1	0
Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P1FC below.>>							

Port 1 Function Register

P1FC (0005H)	7	6	5	4	3	2	1	0
Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00 : IN, 01 : OUT, 10 : AD15-8, 11 : A15-8							

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

→ 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 3.5.4 Registers for Ports 0 and 1

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P2CR and function register P2FC. Resetting resets all bits of output latch P2, control register P2CR and function register P2FC to 0. It also sets Port 2 to input mode and connects a pull-down resistor.
In addition to functioning as a general-purpose I/O port, Port 2 also functions as an address data bus (A0 to 7) and an address bus (A16 to 23). Using Port 2 as address bus (A0 to 7 or A16 to 23), write "1" to output latches and be off the programmable pull-down resistors.

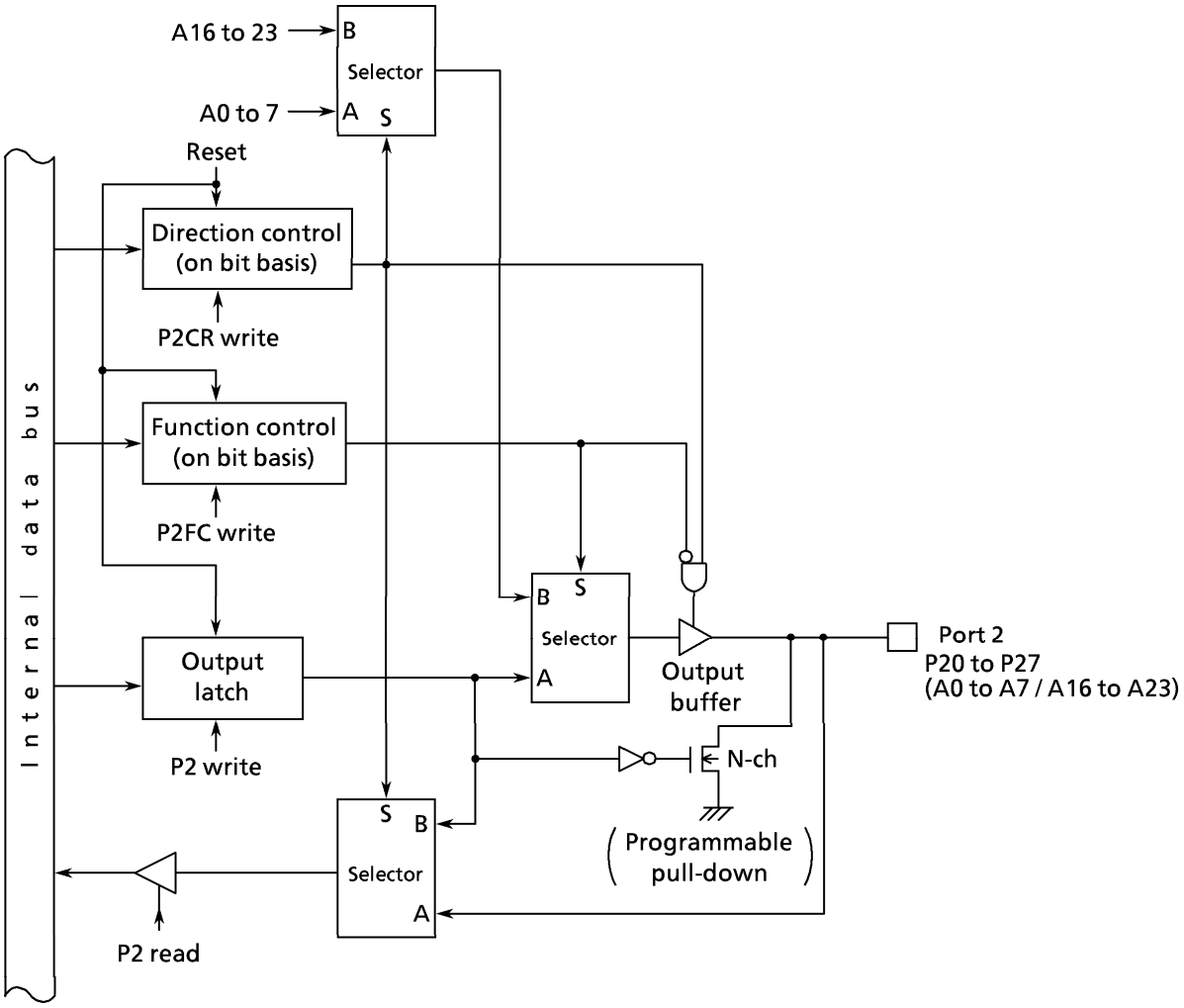


Figure 3.5.5 Port 2

Port 2 Register

P2 (0006H)

	7	6	5	4	3	2	1	0
Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read / Write	R/W							
After reset	Input mode (Output latch register is cleared to "0".)							

Port 2 Control Register

P2CR (0008H)

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

Port 2 Function Register

P2FC (0009H)

	7	6	5	4	3	2	1	0
Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P2FC / P2CR = 00 : IN, 01 : OUT, 10 : A7 to 0, 11 : A23 to 16							

Port 2 function setting

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

Note 1 : Read-modify-write is prohibited for registers P2CR and P2FC.

Note 2 : When port P2 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.

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

Figure 3.5.6 Registers for Port 2

3.5.4 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose I/O port.
I/O can be set on a bit basis, but note that P30 and P31 are used for output only. Both 61 pin and 72 pin output the output latch register <P30>. Only 72 pin is used as $\overline{RD}$ strobe signal, and 61 pin always output the latch register. I/O is set using control register P3CR and function register P3FC. Resetting resets all bits of output latch P3, control register P3CR (bits 0 and 1 are unused), and function register P3FC to 0. Resetting also outputs 1 from P30 and P31, sets P32 to P37 to input mode, and connects a pull-up resistor.
In addition to functioning as a general-purpose I/O port, Port 3 also functions as an I/O for the CPU's control / status signal.

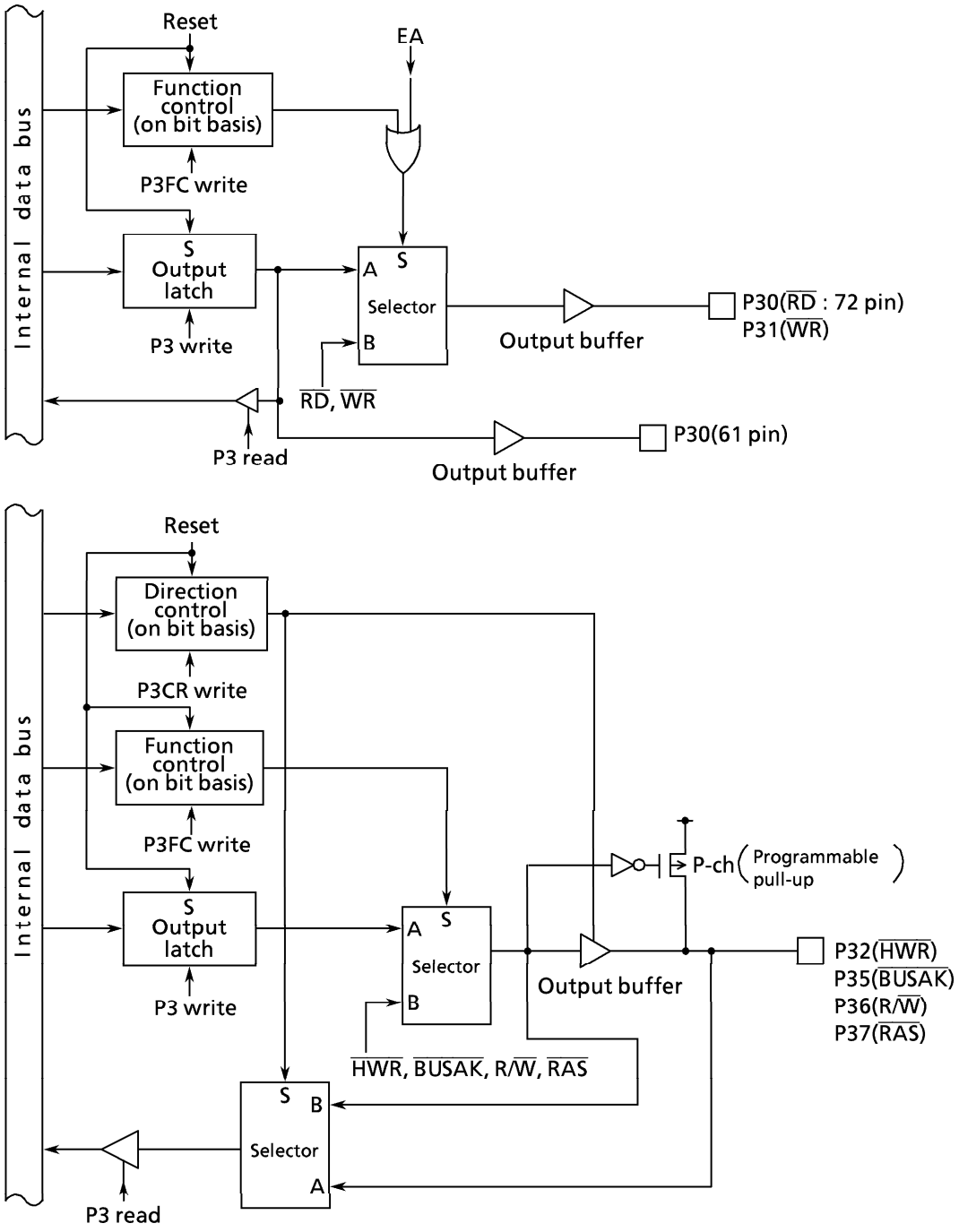


Figure 3.5.7 Port 3 (P30, P31, P32, P35, P36, P37)

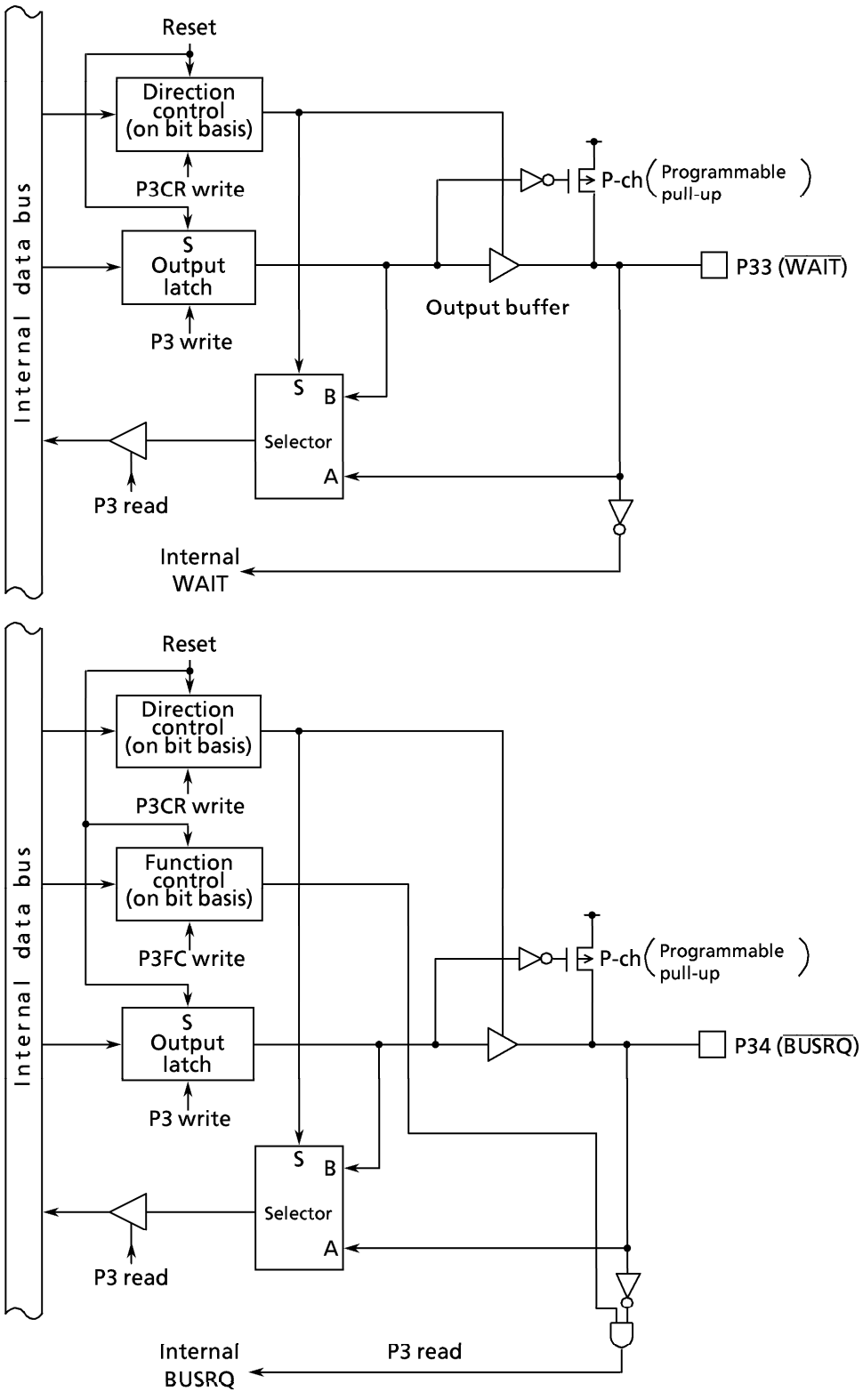


Figure 3.5.8 Port3 (P33, P34)

Port 3 Register

P3 (0007H)

	7	6	5	4	3	2	1	0
Bit symbol	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
	Input mode (Pull-up)						Output mode	

Port 3 Control Register

P3CR (000AH)

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

I/O setting

0	Input
1	Output

Port 3 Function Register

P3FC (000BH)

	7	6	5	4	3	2	1	0
Bit symbol	P37F	P36F	P35F	P34F		P32F	P31F	P30F
Read/Write	W							
After reset	0	0	0	0		0	0	0
Function	0 : PORT 1 : $\overline{RAS}$	0 : PORT 1 : R/$\overline{W}$	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$		0 : PORT 1 : $\overline{HWR}$	0 : PORT 1 : $\overline{WR}$	0 : PORT 1 : $\overline{RD}$

$\overline{BUSRQ}$ setting

P3FC <P34F>	1
P3CR <P34C>	0

$\overline{BUSAK}$ setting

P3FC <P35F>	1
P3CR <P35C>	1

R/$\overline{W}$ setting

P3FC <P36F>	1
P3CR <P36C>	1

$\overline{RAS}$ setting

P3FC <P37F>	1
P3CR <P37C>	1

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

<P30> / <P30F>	0	1
0	"0" output	"1" output
1	$\overline{RD}$ output only for external access	

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

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

$\overline{HWR}$ setting

P3FC <P32F>	1
P3CR <P32C>	1

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

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

Note 3 : When P33/$\overline{WAIT}$ pin is used as a $\overline{WAIT}$ pin, set P3CR<P33C> to "0" and Chip Select/WAIT control register<BnW1,0> to "10".

Figure 3.5.9 Registers for Port 3

3.5.5 Port 4 (P40 to P42)

Port 4 is a 3-bit general-purpose I/O port. I/O can be set on a bit basis using control register P4CR and function register P4FC. Resetting does the following:

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

In addition to functioning as a general-purpose I/O port, Port 4 also functions as a chip select output signal ($\overline{CS0}$ to $\overline{CS2}$ or $\overline{CAS0}$ to $\overline{CAS2}$).

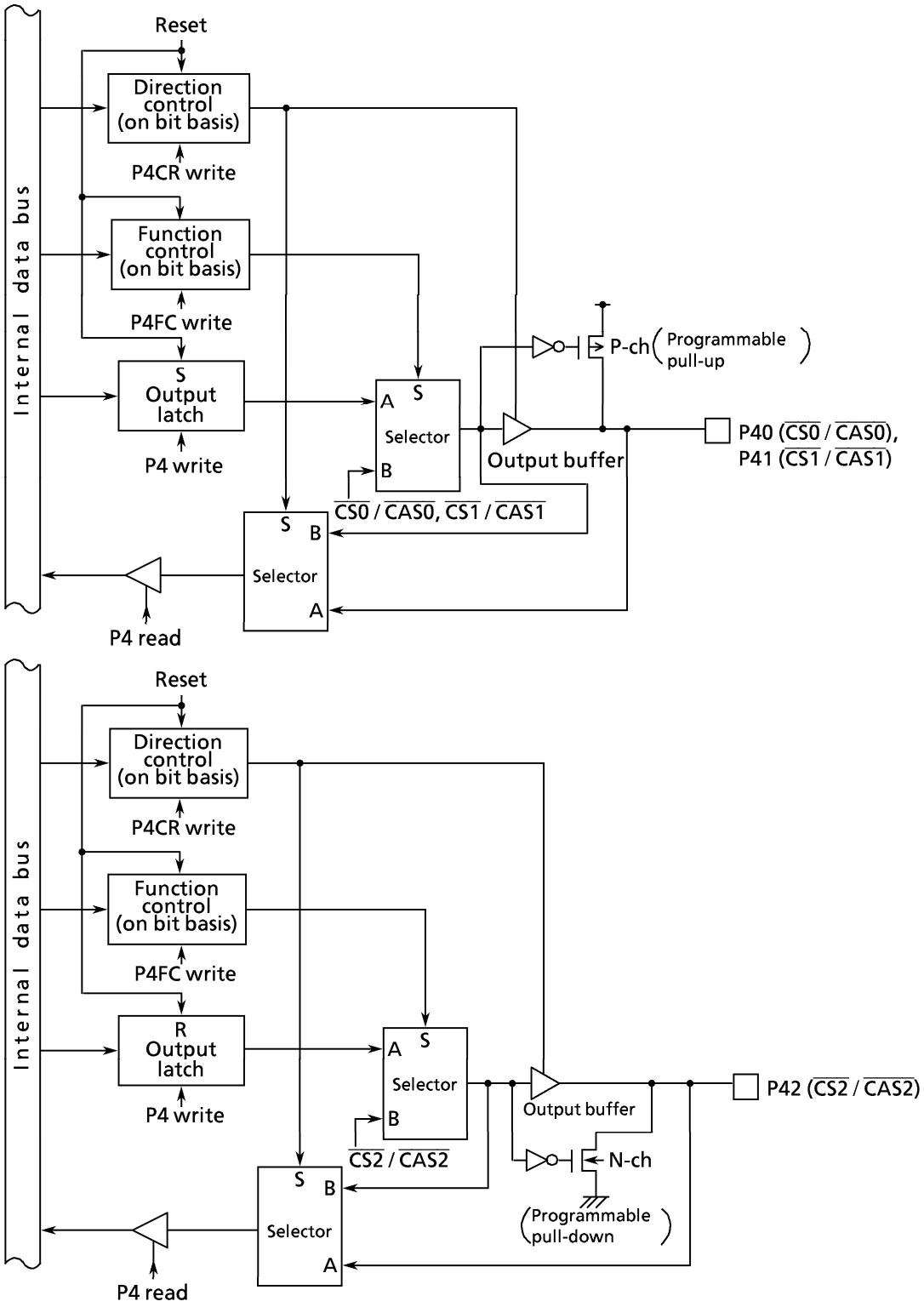


Figure 3.5.10 Port 4

Port 4 Register

P4 (000CH)

	7	6	5	4	3	2	1	0
Bit symbol						P42	P41	P40
Read / Write						R/W		
After reset						0 (Pull-down)	1 (Pull-up)	1 (Pull-up)
						Input mode		

Port 4 Control Register

P4CR (000EH)

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

I/O setting

0	Input
1	Output

Port 4 Function Register

P4FC (0010H)

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

0	Port (P40)
1	$\overline{CS0}$ / $\overline{CAS0}$

0	Port (P41)
1	$\overline{CS1}$ / $\overline{CAS1}$

0	Port (P42)
1	$\overline{CS2}$ / $\overline{CAS2}$

Note 1: Read-modify-write is prohibited for registers P4CR and P4FC.

Note 2: When port P4 is used in the input mode, P4 register controls the built-in pull-up/-down resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up/-down resistor may be depended on the states of the input pin.

Note 3: To output chip select signal (CS0/CAS0 to CS2/CAS2), set the corresponding bits of the control register P4CR and the function register P4FC to "1".
Chip select/wait controller (B0CS, B1CS, B2CS) registers select the function of $\overline{CS}$/$\overline{CAS}$.

Note 4: P4<bit 7 to 3> is always read as "1".

Figure 3.5.11 Registers for Port 4

3.5.6 Port 5 (P50 to P57)

Port 5 is an 8-bit input port, also used as an analog input pin for the internal AD Converter and input pins of the serial interface.

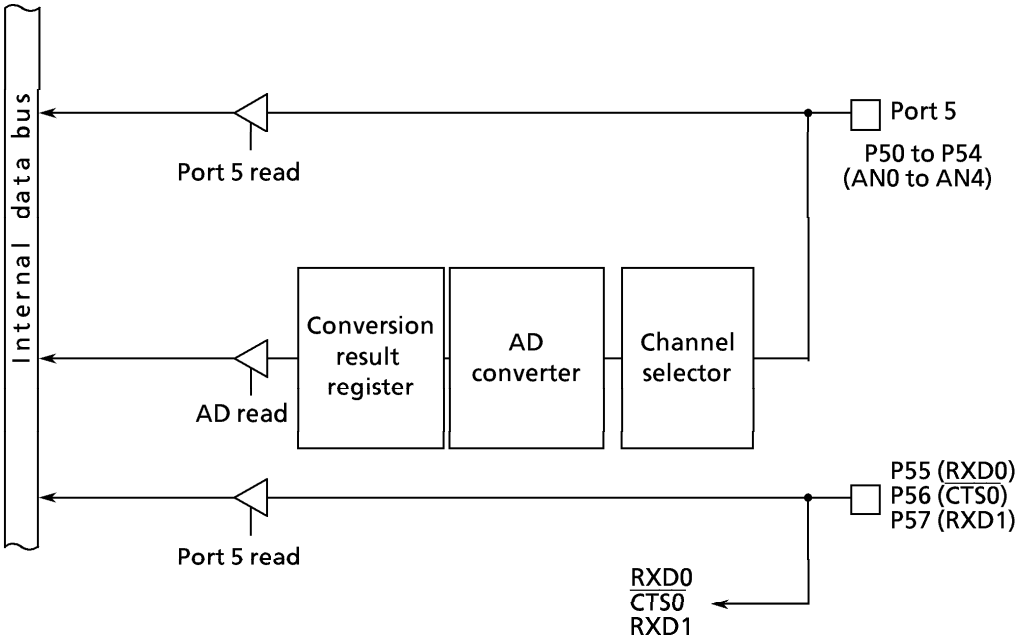


Figure 3.5.12 Port 5

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 AD Converter is set by AD Converter mode register ADMOD2.

Figure 3.5.13 Registers for Port 5

3.5.7 Port 6 (P60 to P67)

Port 6 is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 6 as an input port and connects a pull-up resistor. It also sets all bits of the output latch to 1. Resetting resets the function register P6CR to 0, and sets all bits to input ports.

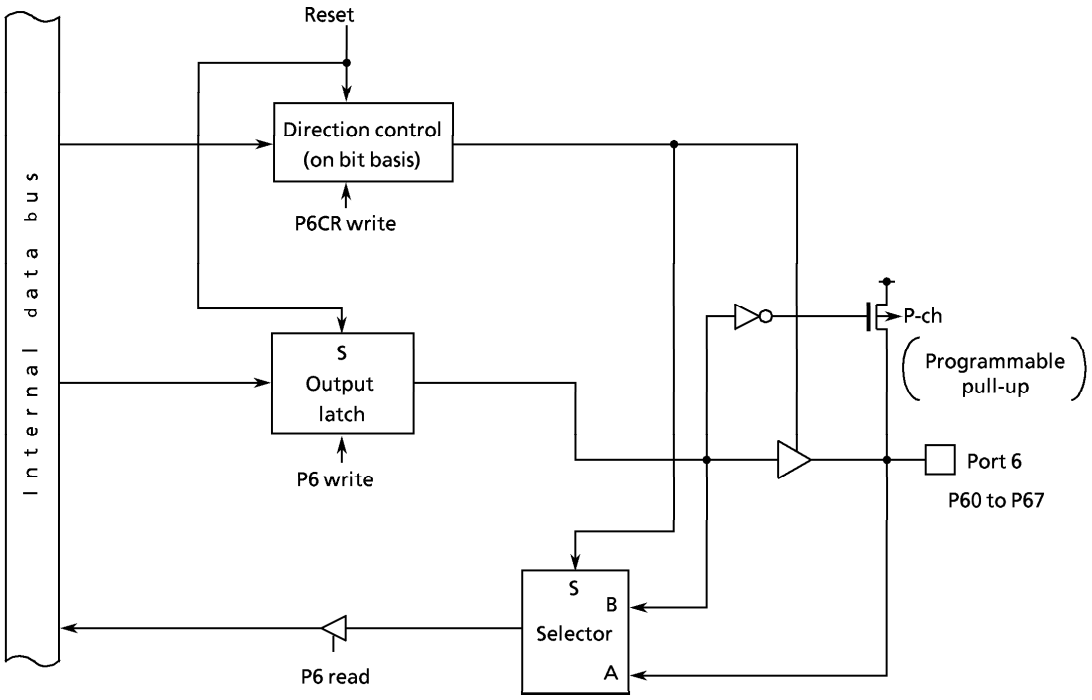


Figure 3.5.14 Port 6

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	Input mode (Pull-up)							
	1	1	1	1	1	1	1	1

Port 6 Control Register

P6CR (0014H)

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

Port 6 I/O setting

0	Input
1	Output

Note 1: Read-modify-write is prohibited for registers P6CR.
Note 2: 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.

Figure 3.5.15 Registers for Port 6

3.5.8 Port 7 (P70 to P73)

Port 7 is a 4-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 7 as an input port and connects a pull-up resistor except P70. In addition to functioning as a general-purpose I/O port, Port 70 also functions as an input clock pin TI0/$\overline{\text{INT9}}$; Port 71 as an 8-bit timer output (TO1), Port 72 as a PWM0 output (TO2), and Port 73 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the Port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7CR, P7FC to 0, and sets all bits to input ports.

The input level of P70 is V_{IH}, 2.8[V] and V_{IL}, 0.8[V], that is different from that of P71 to P73.

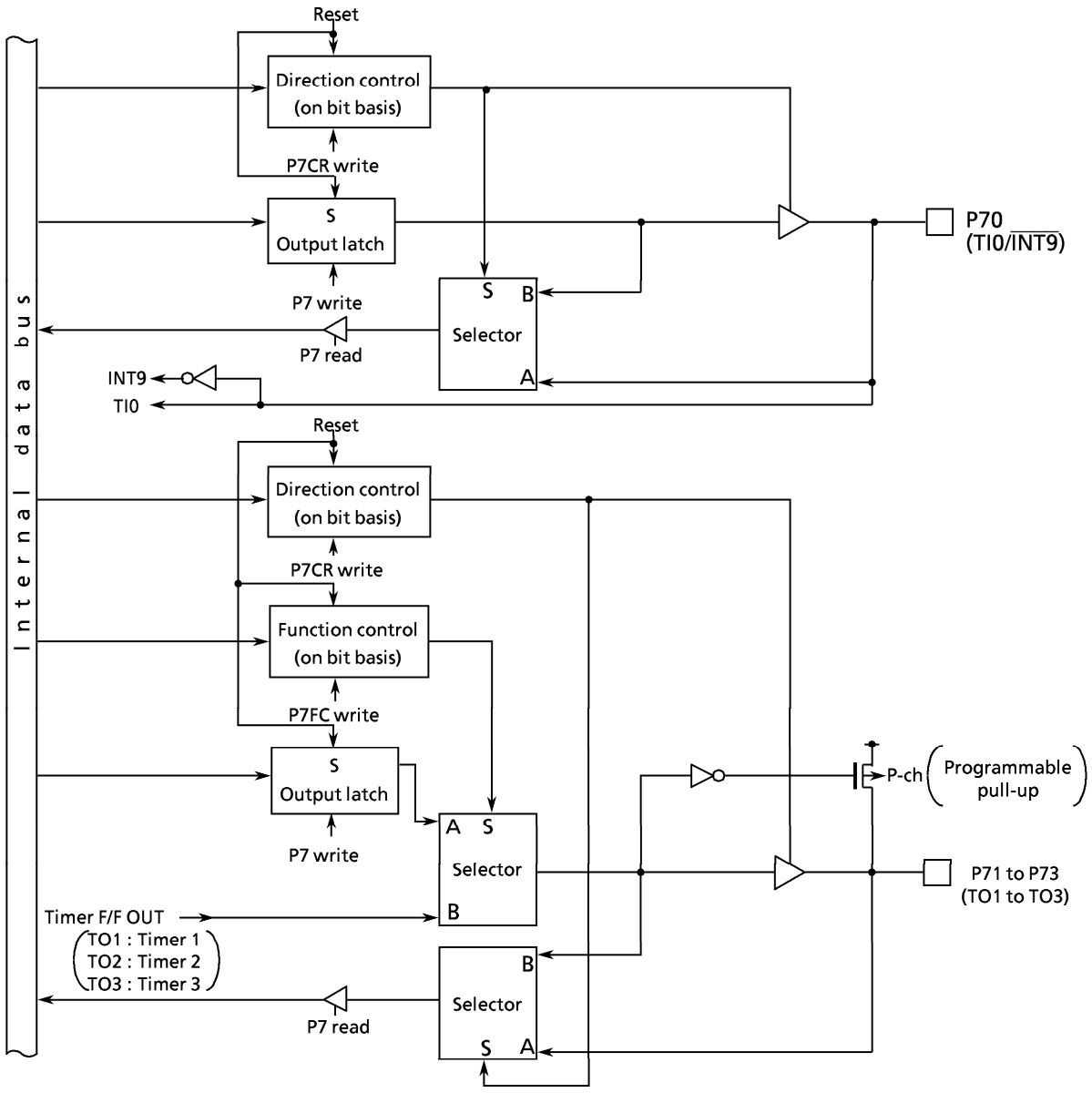


Figure 3.5.16 Port 7

Port 7 Register

		7	6	5	4	3	2	1	0
P7 (0013H)	Bit symbol					P73	P72	P71	P70
	Read/Write					R/W			
	After reset					1	1	1	1
						Input mode (Pull-up)			No pull-up

Port 7 Control Register

		7	6	5	4	3	2	1	0
P7CR (0015H)	Bit symbol					P73C	P72C	P71C	P70C
	Read/Write					W			
	After reset					0	0	0	0
	Function					0 : IN		1 : OUT	

→ Port 7 I/O setting

0	Input
1	Output

Port 7 Function Register

		7	6	5	4	3	2	1	0
P7FC (0017H)	Bit symbol					P73F	P72F	P71F	
	Read/Write					W			
	After reset					0	0	0	
	Function					0 : PORT 1 : TO3	0 : PORT 1 : TO2	0 : PORT 1 : TO1	

→ Setting P71 as TO1

P7FC <P71F>	1
P7CR <P71C>	1

→ Setting P72 as TO2

P7FC <P72F>	1
P7CR <P72C>	1

→ Setting P73 as TO3

P7FC <P73F>	1
P7CR <P73C>	1

Note 1 : Read-modify-write is prohibited for registers P7CR and P7FC.
Note 2 : When port P7 is used in the input mode, P7 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.
Note 3 : P70/T10/$\overline{\text{INT9}}$ pin does not have a register changing PORT/FUNCTION.
For example, when it is used as an input port, the input signal is inputted to 8 bit Timer 0 as a timer input 0 (T10), and to the interrupt controller as $\overline{\text{INT9}}$ input.
Note 4 : P7<bit 7 to 4> is always read as "1".

Figure 3.5.17 Registers for Port 7

3.5.9 Port 8 (P80 to P87)

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets Port 8 as an input port and connects a pull-up resistor except P81, P85 and P87. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, Port 8 also functions as an input for 16-bit timer 4 and 5 clocks, an output for 16-bit timer F/F 4, 5, and 6 output, and an input for $\overline{\text{INT0}}$. Writing "1" in the corresponding bit of the Port 8 function register (P8FC) enables those functions. Resetting resets the function register P8CR, P8FC to "0" and sets all bits to input ports.

(1) P80 to P86

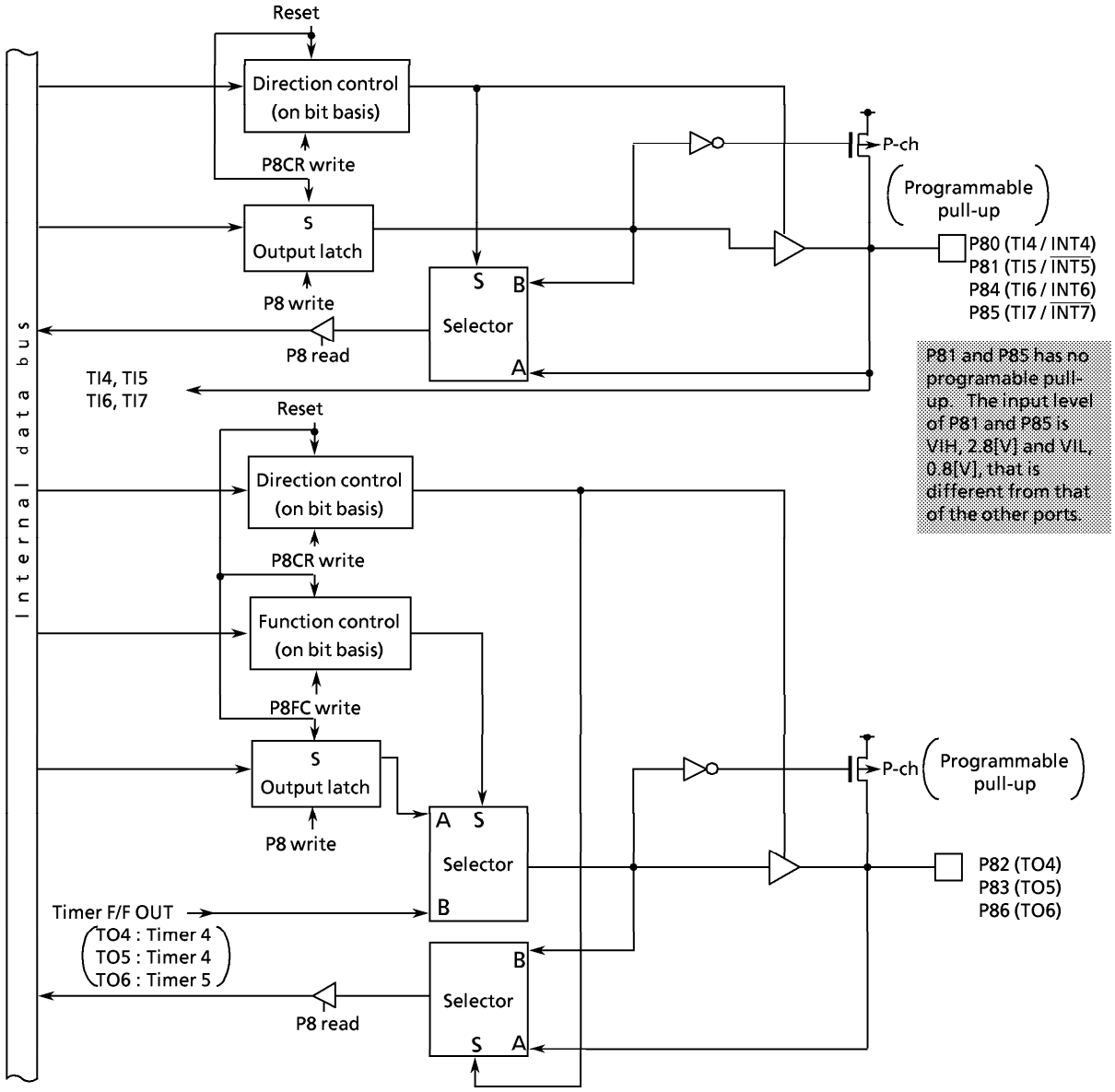


Figure 3.5.18 Port 8 (P80 to P86)

(2) P87 ($\overline{INT0}$)

Port 87 is a general-purpose I/O port, and also used as an $\overline{INT0}$ pin for external interrupt request input. The input level of P87 is V_{IH}, 2.8 [V] and V_{IL}, 0.8 [V], that is different from that of P80, P82 to P84 and P86.

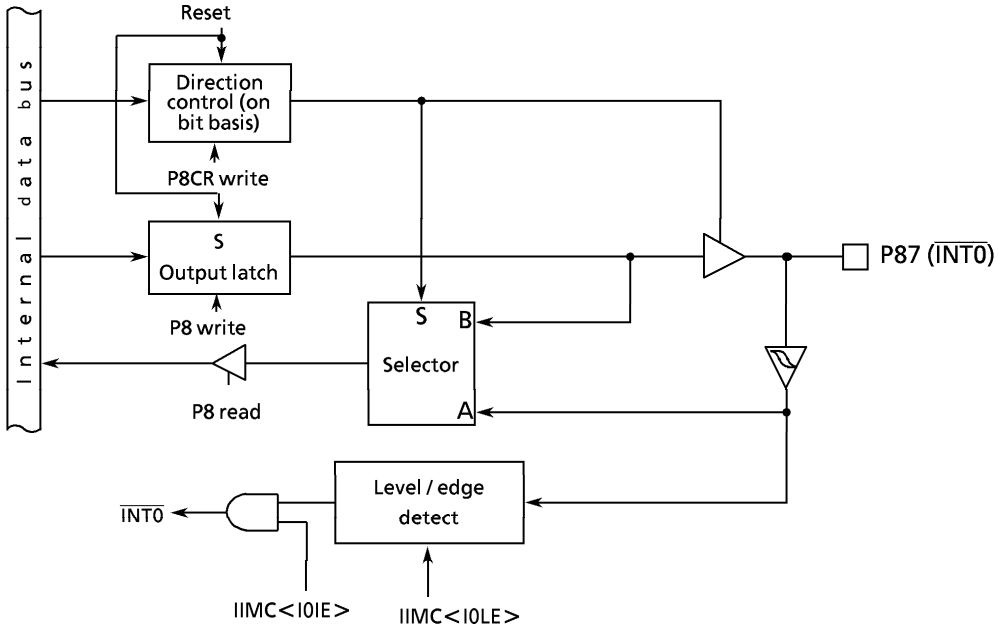


Figure 3.5.19 Port 87

Port 8 Register

P8 (0018H)

	7	6	5	4	3	2	1	0
Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read / Write	R/W							
After reset	1	1	1	1	1	1	1	1
	Input mode							

Port 8 Control Register

P8CR (001AH)

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

Port 8 I/O setting

0	Input
1	Output

Port 8 Function Register

P8FC (001CH)

	7	6	5	4	3	2	1	0
Bit symbol		P86F			P83F	P82F		
Read / Write		W			W	W		
After reset		0			0	0		
Function		0 : PORT 1 : TO6			0 : PORT 1 : TO5	0 : PORT 1 : TO4		

Setting P82 as TO4

P8FC <P82F>	1
P8CR <P82C>	1

Setting P83 as TO5

P8FC <P83F>	1
P8CR <P83C>	1

Setting P84 as TO6

P8FC <P86F>	1
P8CR <P86C>	1

Note 1: Read-modify-write is prohibited for registers P8CR and P8FC.

Note 2: When port P8 is used in the input mode, P8 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: P80/T14, P81/T15, P84/T16, P85/T17 pins do not have a register changing PORT/FUNCTION.
For example, when it is used as an input port, the input signal is inputted to 16 bit timer as timer input. When P87/$\overline{INT0}$ pin is used as an $\overline{INT0}$ pin, set P8CR<P87C> to "0" and IIMC<I0IE> to "1".

Figure 3.5.20 Registers for Port 8

3.5.10 Port 9 (P90 to P97)

- Port 90 to 95

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

- Port 96 to 97

  Port 96 to 97 is a 2-bit general-purpose I/O port. I/Os can be set on a bit basis.
  The output buffer for P96 to 97 is an open drain type buffer.
  Resetting sets output latch and control registers to "1" and outputs high-impedance (HZ).

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

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

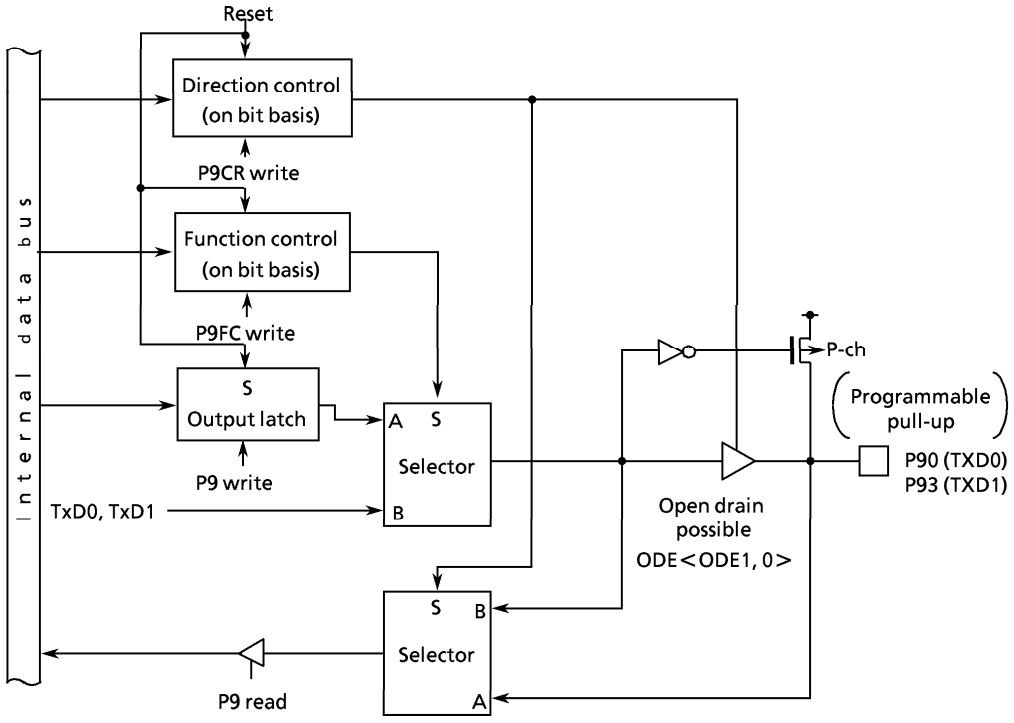


Figure 3.5.21 Ports 90 and 93

(2) Port 91, 92, 94

Port 91, 92 and 94 are I/O ports.

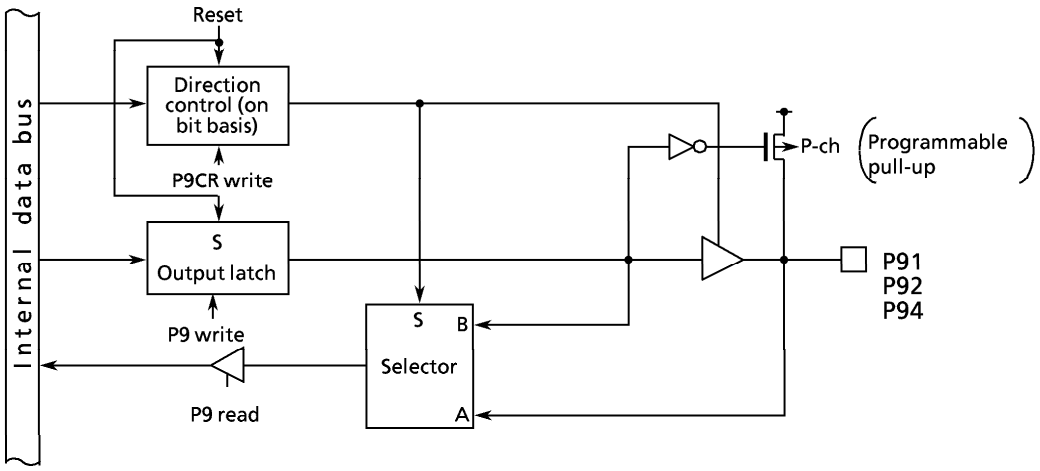


Figure 3.5.22 Ports 91, 92 and 94

(3) Port 95 (SCLK1)

Port 95 is a general-purpose I/O port. It is also used as a SCLK1 I/O pin for serial channel 1.

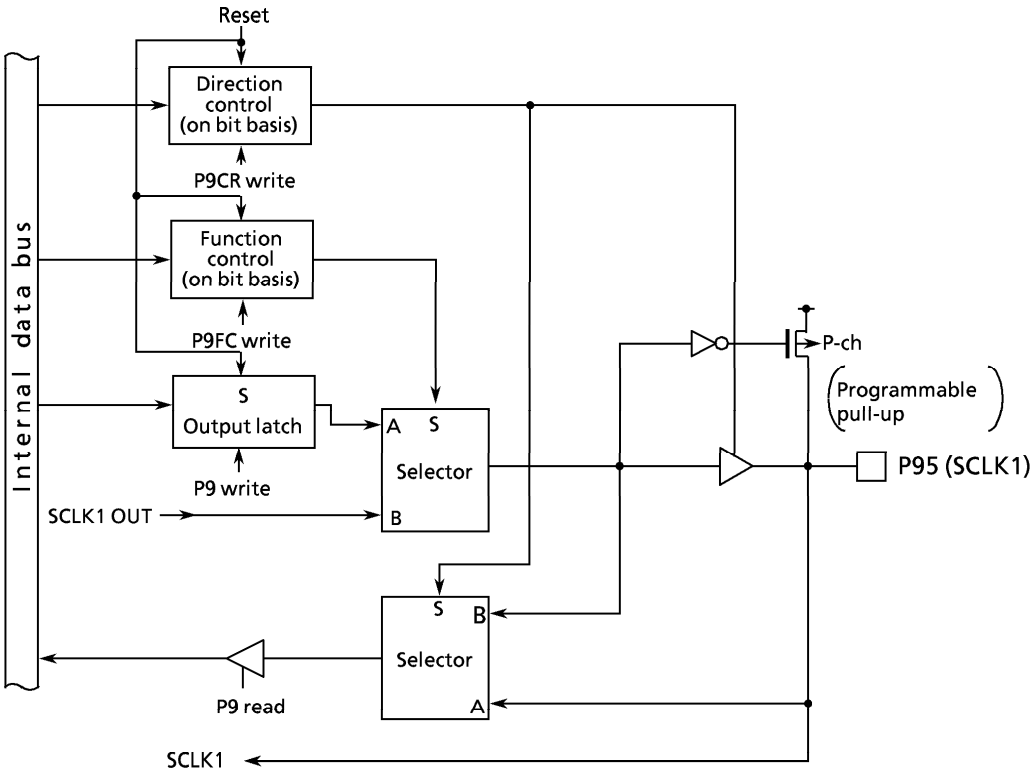


Figure 3.5.23 Port 95

(4) Port 96, 97

Port 96, 97 is general purpose I/O ports. The output is open drain.

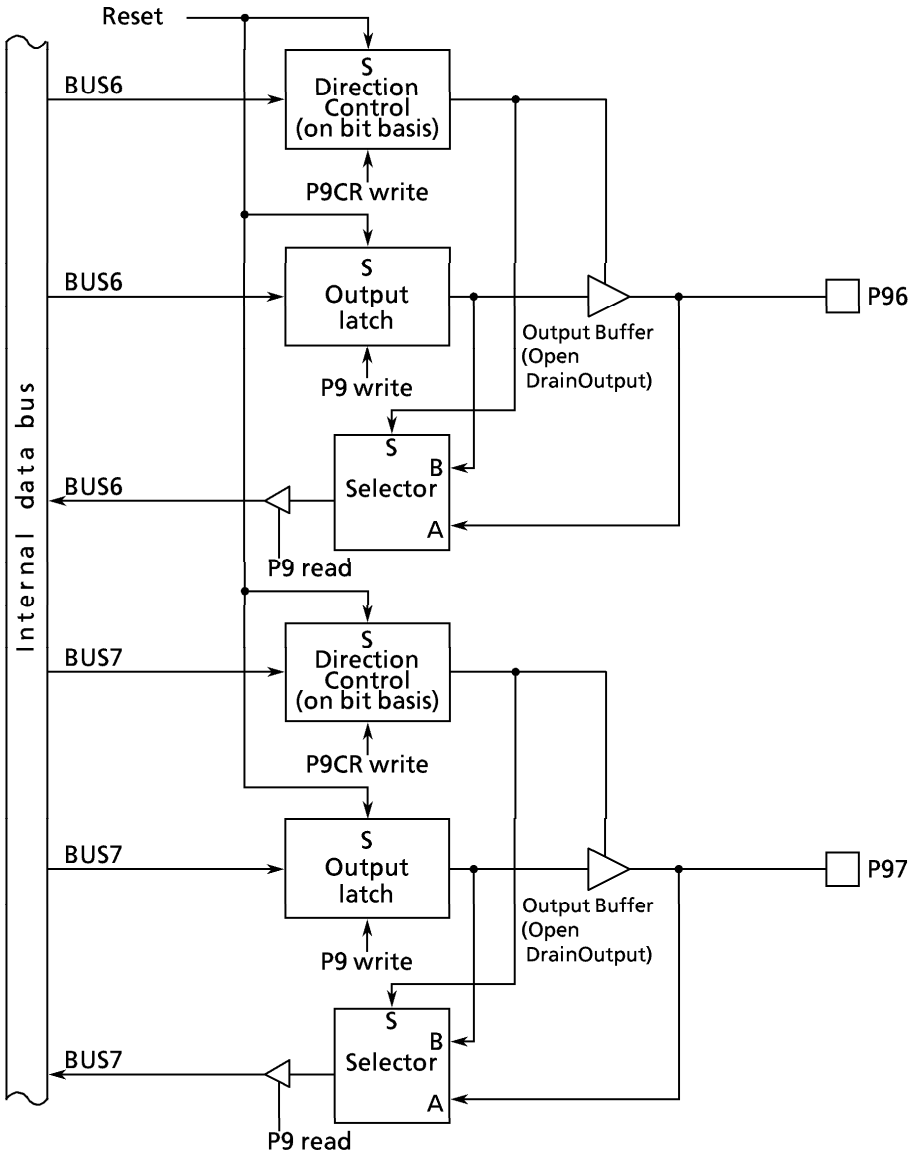


Figure 3.5.24 Port 96 to 97

Port 9 Register

P9 (0019H)

	7	6	5	4	3	2	1	0
Bit symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
	Output mode		Input mode					

Port 9 Control Register

P9CR (001BH)

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

Port 9 I/O setting

0	Input
1	Output

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

Port 9 Function Register

P9FC (001DH)

	7	6	5	4	3	2	1	0
Bit snmbol			P95F		P93F			P90F
Rean/Write			W		W	W		W
Aften reset			0		0	0		0
Function			0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	Set to 0		0 : PORT 1 : TxD0

P90 TxD0 output setting (Note)

P9FC <P90F>	1
P9CR <P90C>	1

P93 TxD1 output setting (Note)

P9FC <P93F>	1
P9CR <P93C>	1

P95 SCLK1 output setting

P9FC <P95F>	1
P9CR <P95C>	1

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

Note 2: When port P9 is used in the input mode, P9 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: To set the TxD pin to open drain, write "1" in bit 0 (for TxD0 pin) or 1 (for TxD1) pin of the ODE register.

Note 4: When ports 96 and 97 are used in the output mode, input gate in operation.
Set output to "L" or attach pull-up on pin to reduce the consumption of power, before the HALT instruction is executed.

Figure 3.5.25 Register for Port 9

3.5.11 Port A (PA0 to PA7)

Port A is an 8-bit general-purpose I/O port. I/Os can be set on a bit basis by control register PACR. Resetting sets Port A to an input port and resets PACR to "0".

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

The output clock is f_{FPH} or f_{SYS}, which is selected by CKOCR<SCOSEL>. SCOUT function is enabled by writing "1" to PACR<PA7C> and CKOCR<SCOEN>.

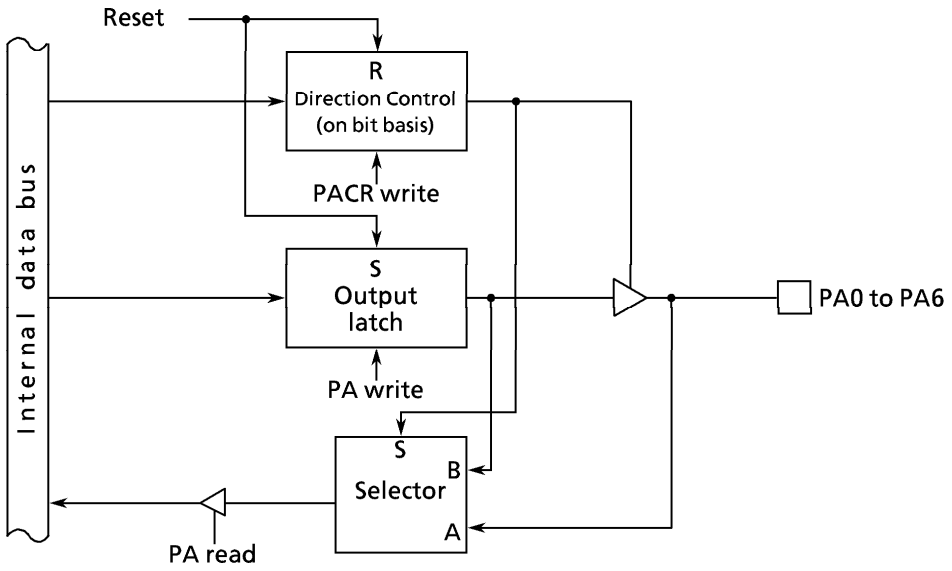


Figure 3.5.26 Port A0 to A6

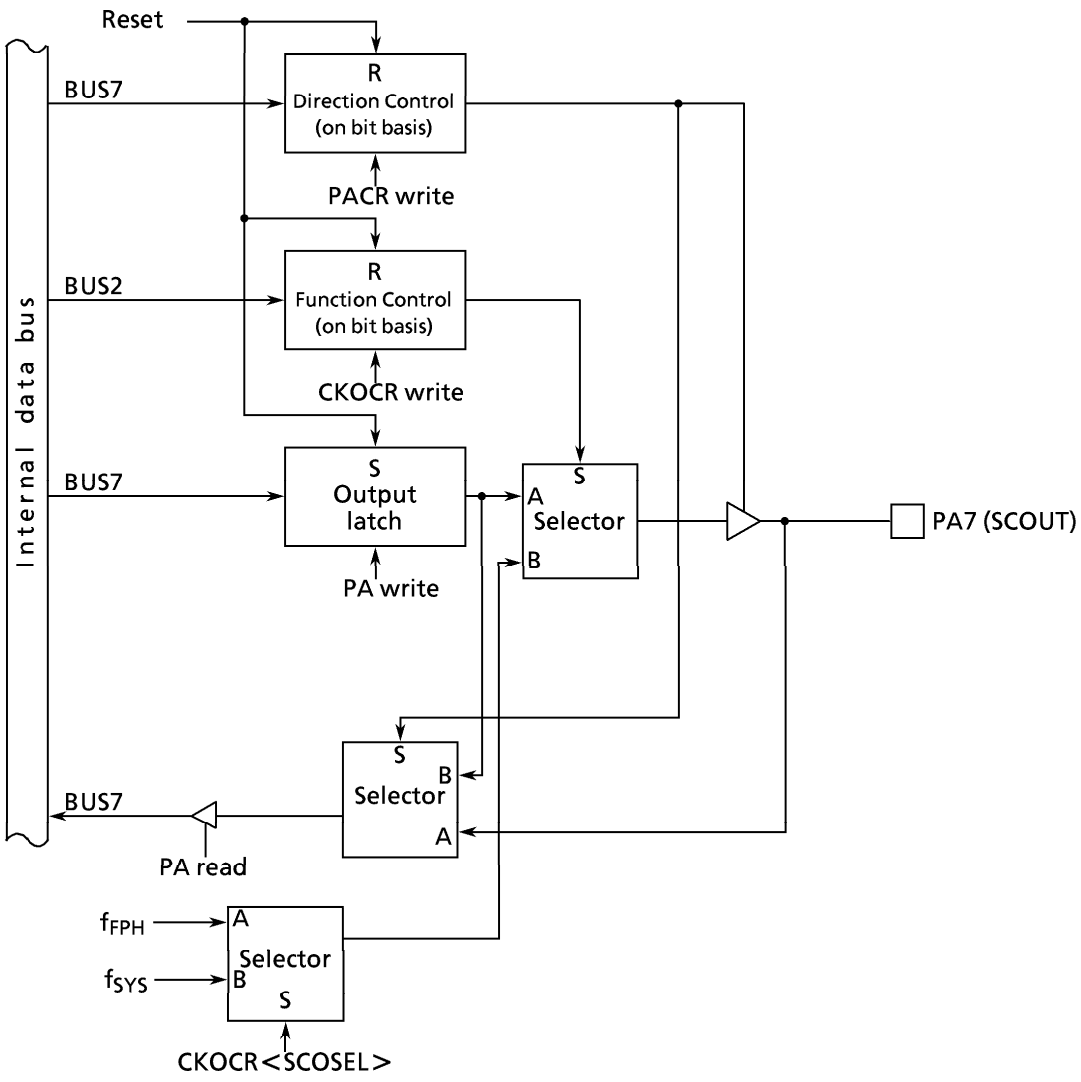


Figure 3.5.27 Port A7

Port A Register

PA (001EH)

	7	6	5	4	3	2	1	0
Bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
	Input mode							

Port A Control Register

PACR (001FH)

	7	6	5	4	3	2	1	0
Bit symbol	PA7C	PA6C	PA5C	A94C	PA3C	PA2C	PA1C	PA0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	0 : IN				1 : OUT			

Clock Output Control Register

CKOCR (006DH)

	7	6	5	4	3	2	1	0
Bit symbol					SCOSEL	SCOEN	ALEEN	CLKEN
Read/Write	R/W							
After reset					0	0	0	0
	Write to 0	Write to 0			Clock Select	Clock Select		

Note 1: Read-modify-write is prohibited for registers PACR.
Note 2: The CLK pin is pulled up internally during reset.
Note 3: The output clock from SCOUT pin is f_{FPH} or f_{SYS} clock.
eg) The case of connected 20 MHz oscillator to X1, X2 pin.
<SCOSEL> = "0" → 20 MHz clock
<SCOSEL> = "1" → 10 MHz clock
Note 4: CKOCR<bit 7-4> is always read as "1". Write "00" to CKOCR<bit 7-4>.

CLK pin output control (Note 1)

0	"High-z" output
1	CLK output

ALE pin output control

0	"High-z" output
1	ALE output

SCOUT/PA7 pin control

<SCOEN>	PACR<PA7C> <SCOSEL>	0	1
0	0	Input Port	Output mode
	1		
1	0		f_{FPH} clock output (Note3)
	1		f_{SYS} clock output

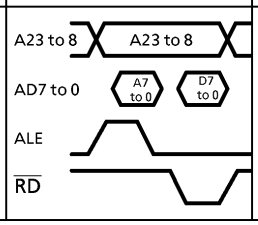
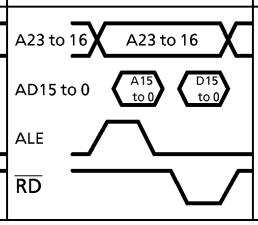
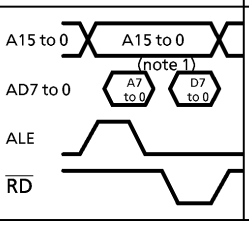
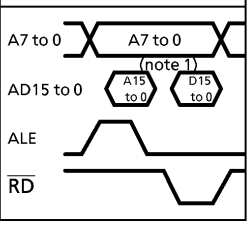
Figure 3.5.28 Registers for PortA

3.6 Chip Select / Wait Controller

TMP93CS42A has a built-in chip select / wait controller used to control chip select ($\overline{CS0}$ to $\overline{CS2}$ pins), wait ($\overline{WAIT}$ pin), and data bus size (8 or 16 bits) for any of the three block address areas.

3.6.1 Address / Data bus pins

Port 0/AD0 to 7, Port 1/AD8 to 15 and Port 2 / A16 to 23 / A0 to 7 function as address / data bus for connecting the external memories.

		①	②	③	④
Number of address bus pins		max 24 (to 16 MB)	max 24 (to 16 MB)	max 16 (to 64 KB)	max 8 (to 256B)
Number of data bus pins		8	16	8	16
Number of multiplexed pins		8	16	0	0
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 CK0CR<ALEEN>, ALE signal can be stopped outputting.

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

3.6.2 Chip select / Wait Control Registers

Table 3.6.1 shows control registers.
One block address areas are controlled by 1-byte CS / WAIT control registers (B0CS, B1CS, and B2CS).

(1) Enable

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

(2) CS/CAS Waveform select

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

(3) Data bus size select

Bit 4 (B0BUS, B1BUS, and B2BUS) of the control register is used to specify data bus size. Setting this bit to 0 accesses the memory in 16-bit data bus mode; setting it to 1 accesses the memory in 8-bit data bus mode.
Changing data bus size depending on the access address is called dynamic bus sizing. Table 3.6.2 shows the details of the bus operation.

(4) Wait control

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

(5) Address area specification

Control register bits 1 and 0 (B0C1-0, B1C1-0, B2C1-0) are used to specify the target address area. Setting these bits to 00 enables settings as follows and outputs a low strobe signal from chip select pins.

* CS0 setting enabled when 2000H to 3FFFH is accessed.
* CS1 setting enabled when 4000H to 5FFFH is accessed.
* CS2 setting enabled when 6000H to 7FFFH is accessed.

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

Table 3.6.1 Chip select / wait control register

Code	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block0 CS/WAIT control register	0068H	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 2000H to 3FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B1CS	Block1 CS/WAIT control register	0069H	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 4000H to 5FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B2CS	Block2 CS/WAIT control register	006AH	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 6000H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	

Note : Read-modify-write is prohibited for B0CS, B1CS and B2CS.

Table 3.6.2 Dynamic bus sizing

Operand data size	Operand start address	Memory data size	CPU address	CPU data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (even number)	8 bits	2n + 0	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.

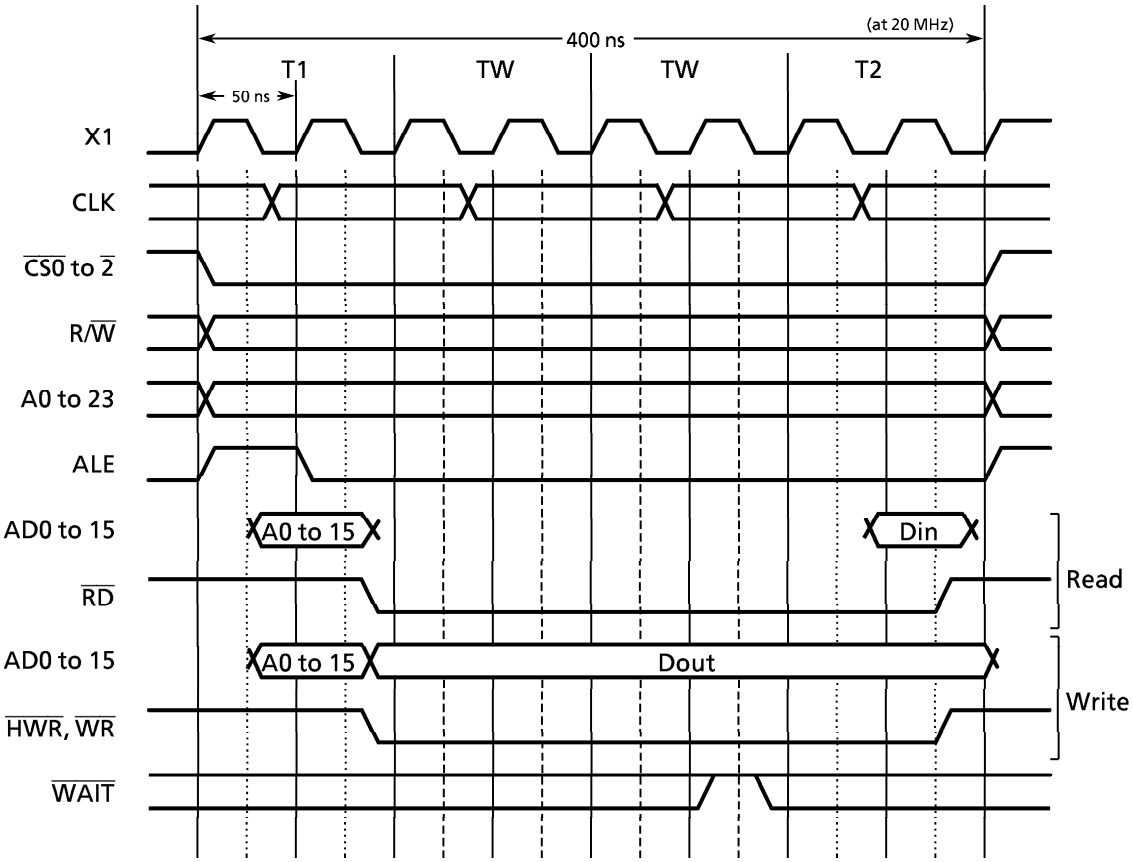


Figure 3.6.1 2WAIT + n Read/Write Cycle (n = 0)

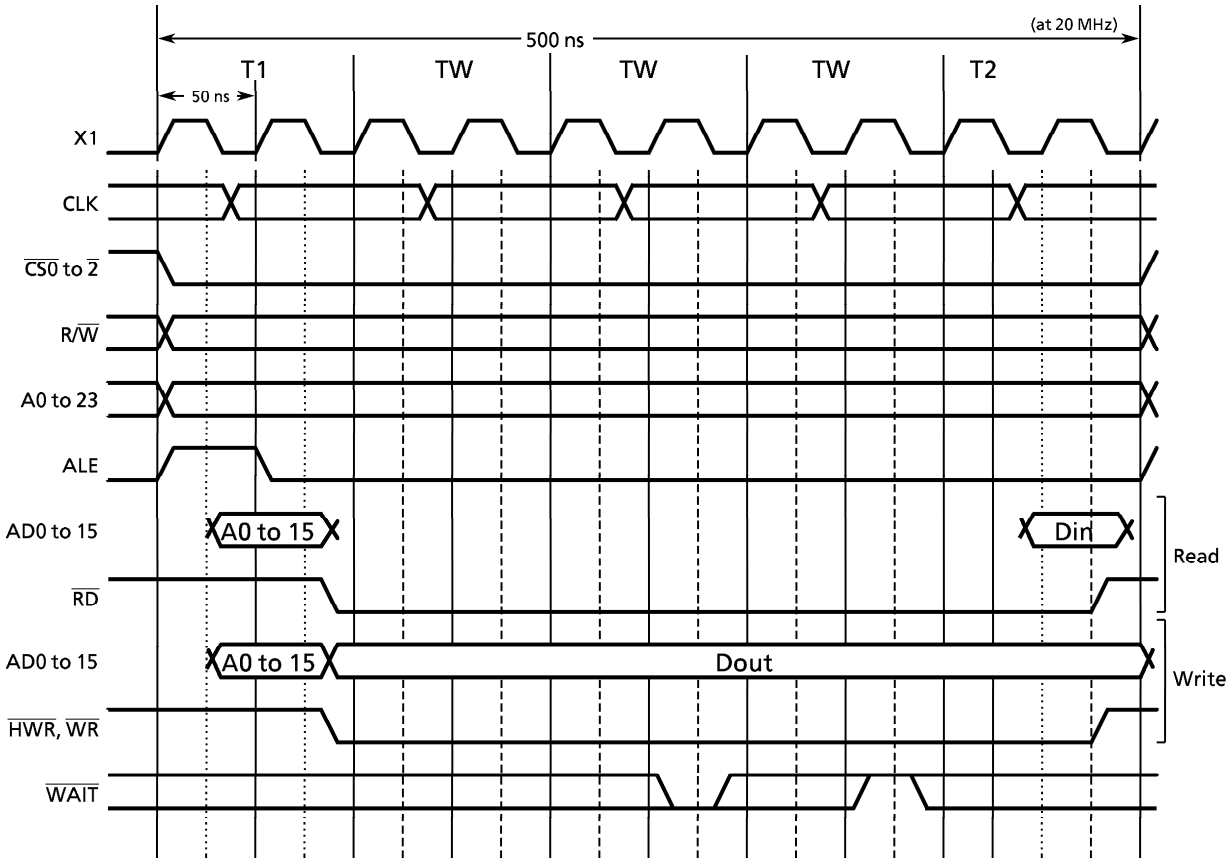
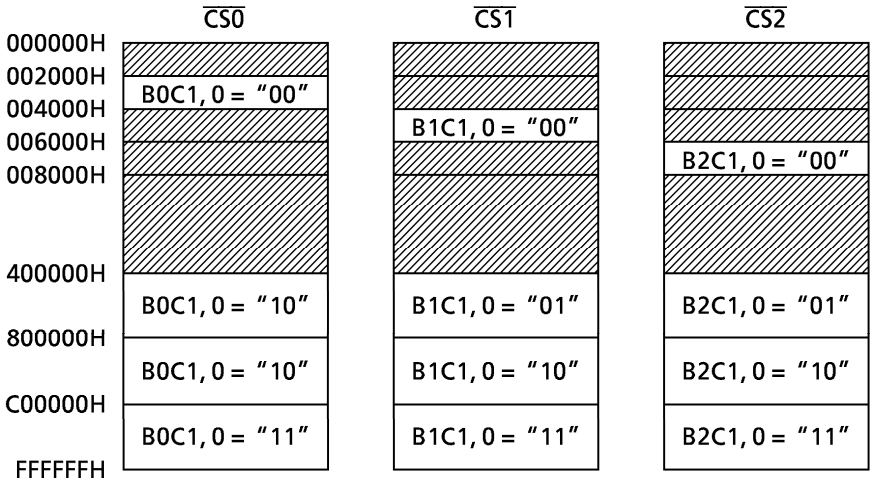


Figure 3.6.2 2WAIT + n Read/Write Cycle (n = 1)

3.6.3 Chip Select Addresses Image

An image of the actual chip select is shown below. Out of the whole memory area, address areas that can be specified are divided into four parts. Addresses from 000000H to 3FFFFFH are divided differently: 2000H to 3FFFH is specified for CS0; 4000H to 5FFFH, for CS1; and 6000H to 7FFFH, for CS2.



Supplement : External areas other than $\overline{CS0}$ to $\overline{CS2}$ are accessed in 16-bit data bus (0 wait) mode.
When using the chip select/wait controller, do not specify the same address area more than once.

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

3.6.4 Example of Usage

(1) Example of Usage-1

Figure 3.6.3 is an example in an external memory is connected to the TMP93CS42A. In this example, a ROM is connected using 16 bit bus; a RAM is connected using 8 bit bus.

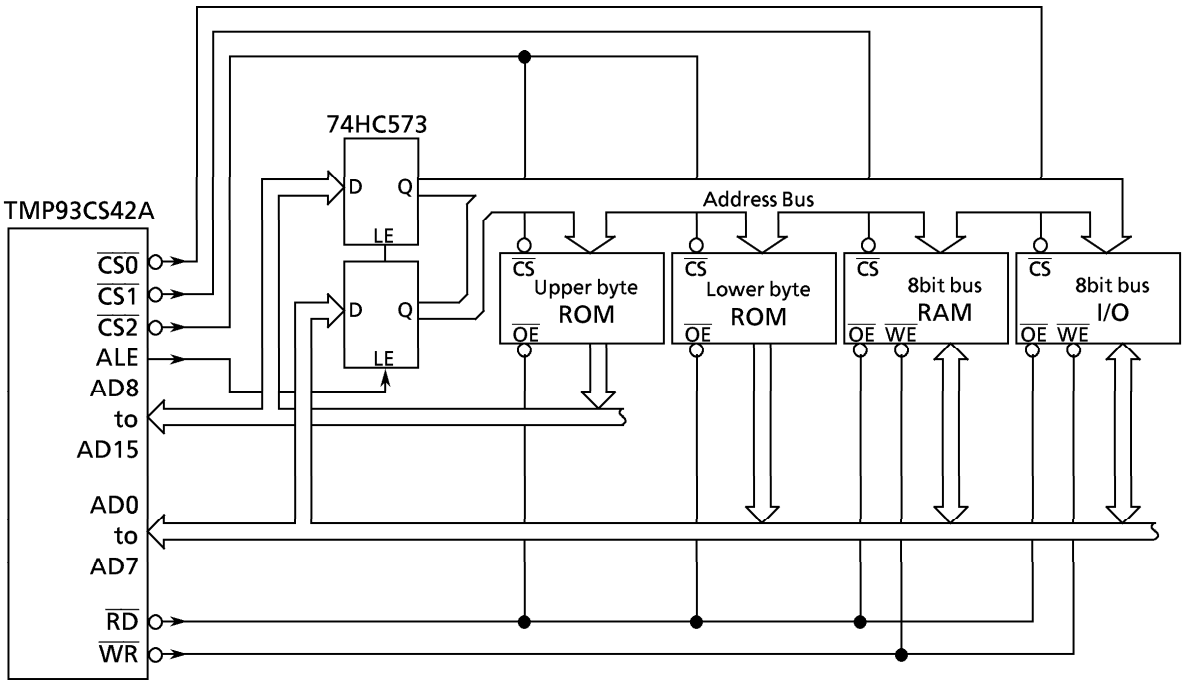


Figure 3.6.3 Example of External Memory Connection (ROM = 16 bits, RAM and I/O = 8 bits)

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

```
P4CR   EQU    0EH
P4FC   EQU    10H
B0CS   EQU    68H
B1CS   EQU    69H
B2CS   EQU    6AH
LD     (B0CS), 1X010000B    ; CS0 = 8bit, 2WAIT, 2000H to 3FFFH
LD     (B1CS), 1X011100B    ; CS1 = 8bit, 0WAIT, 4000H to 5FFFH
LD     (B2CS), 1X000100B    ; CS2 = 16bit, 1WAIT, 6000H to 7FFFH
LD     (P4CR), XXXXX111B    } CS0, CS1, CS2 output mode setting
LD     (P4FC), XXXXX111B    }
```

Note: X ; Don't care

(2) Example of Usage-2

Figure 3.6.4 is an example in an external memory is connectedto the TMP93CS42A. In this example, a ROM, RAM, and I/O are connected using 8 bit bus.

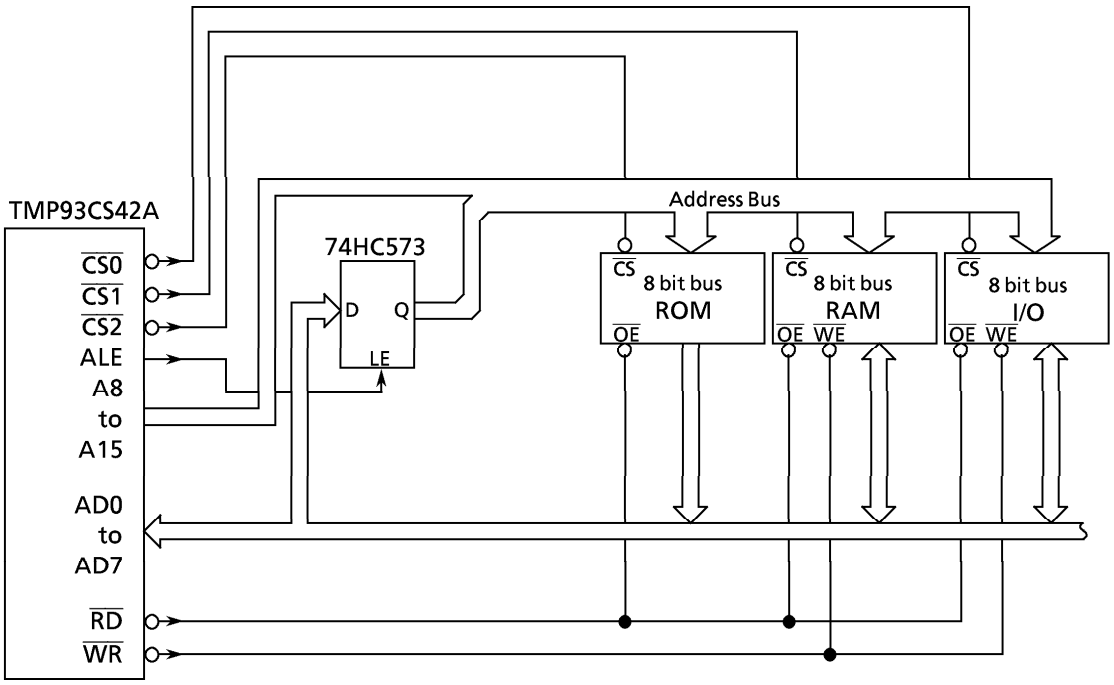


Figure 3.6.4 Example of External Memory Connection (ROM and RAM and I/O = 8 bit)

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

```
P4CR   EQU    0EH
P4FC   EQU    10H
B0CS   EQU    68H
B1CS   EQU    69H
B2CS   EQU    6AH
LD     (B0CS), 1X010000B     ; CS0 = 8 bit, 2WAIT, 2000H to 3FFFH
LD     (B1CS), 1X011100B     ; CS1 = 8 bit, 0WAIT, 4000H to 5FFFH
LD     (B2CS), 1X010100B     ; CS2 = 8 bit, 1WAIT, 6000H to 7FFFH
LD     (P4CR), XXXXX111B     } CS0, CS1, CS2 output mode setting
LD     (P4FC), XXXXX111B     }
```

Note: X ; Don't care

(3) Example of Usage-3

Figure 3.6.5 is an example in which an external memory is connected to the TMP93CS42A. In this example, a ROM 128 K byte is connected using 16 bit bus, and RAM 256 K byte using 16 bit bus.

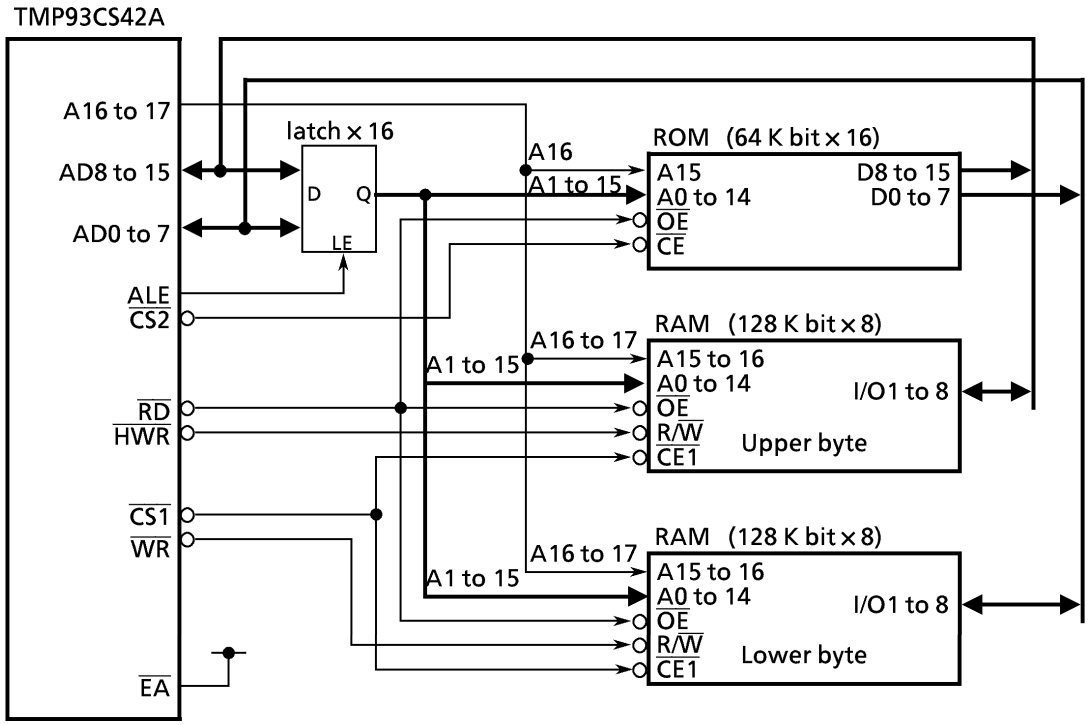


Figure 3.6.5 Example of External Memory Connection (ROM and RAM = 16 bits)

TMP93CS42A has built-in ROM and RAM. When ROM and RAM is insufficient in 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	$\overline{CS}$ pin	Data bus
ROM	Internal	64 KBytes	008000H to 017FFFH	–	16 bits
	External	128 KBytes	400000H to 41FFFFH	$\overline{CS2}$	16 bits
SRAM	Internal	2 KBytes	000080H to 00087FH	–	16 bits
	External	256 KBytes	800000H to 83FFFFH	$\overline{CS1}$	16 bits

3.7 8-Bit Timers

The TMP93CS42A contains two 8-bit timers (timers 0 and 1), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

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

Figure 3.7.1 shows the block diagram of 8-bit timer (timer 0 and timer 1).
Each timer consists of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flip-flop (TFF1) is provided for pair of timer 0 and timer 1.
Among the input clock sources for the timers, the internal clocks of ϕT1, ϕT4, ϕT16, and ϕT256 are obtained from the 9-bit prescaler shown in Figure 3.7.2.
The operation modes and timer flip-flops of the 8-bit timer are controlled by three control registers TMOD, TFFCR, and TRUN.

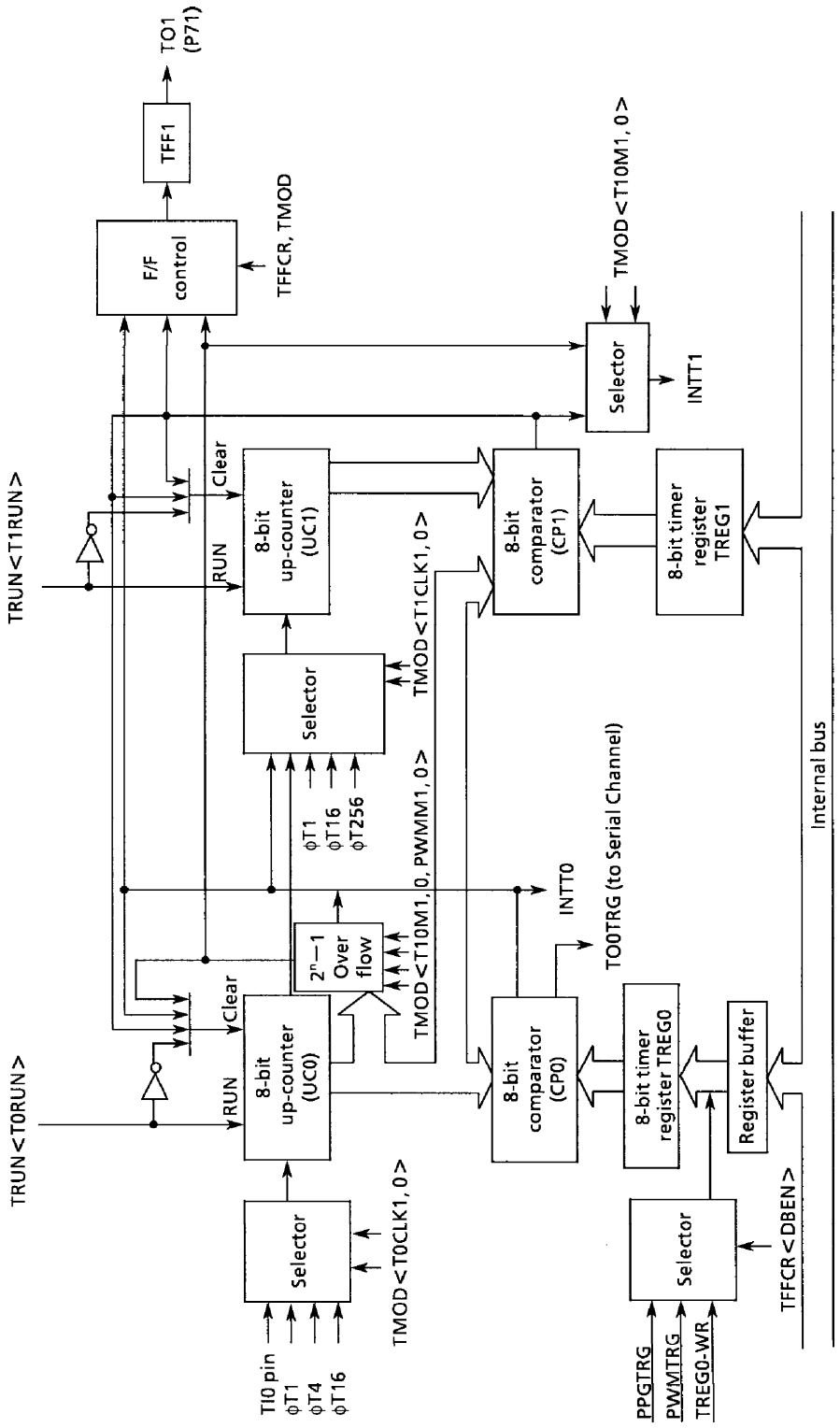


Figure 3.7.1 Block Diagram of 8-bit Timers (Timers 0 and 1)

① Prescaler, Prescaler clock select

There are 9 bit prescaler and prescaler clock selection registers to generate input clock for 8-bit Timer 0, 1, 16-bit Timer 4, 5 and Serial Interface 0, 1.
Figure 3.7.2 shows the block diagram. Table 3.7.1 shows prescaler clock resolution into 8-/16-bit Timer.

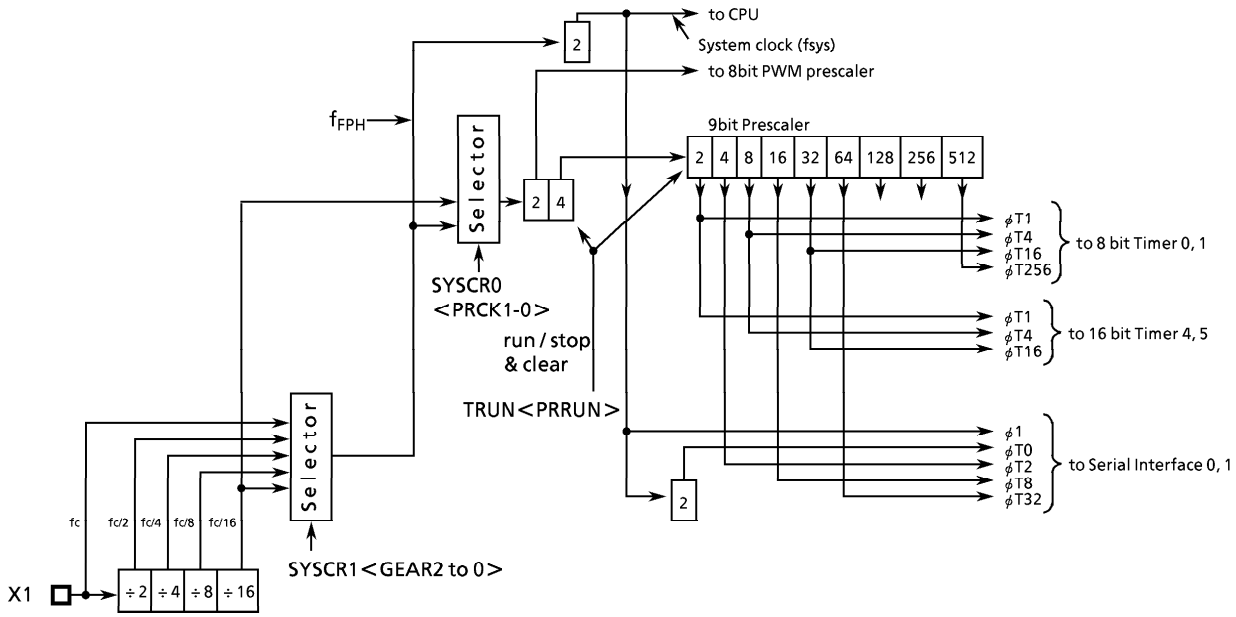


Figure 3.7.2 The Block Diagram of Prescaler

Table 3.7.1 Prescaler Clock Resolution to 8, 16 bit Timer

at fc = 20 MHz

Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Prescaler Clock Resolution			
		φT1	φT4	φT16	φT256
00 (f_{FPH})	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)
10 (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

8-bit Timer

The clock selected among f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1, 0>.
Resetting set <PRCK1, 0> to "00" selects the f_{FPH} clock input divided by 4.
The 8 bit Timer 0, 1 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". The prescaler is cleared to zero and stops operation when <PRRUN> is set to "0". Resetting clear <PRRUN> to "0" and stops the prescaler.
When the IDLE1 mode (only the oscillator operates) is used, set TRUN <PRRUN> to '0' to reduce the power consumption of the prescaler before the "HALT" instruction is executed.

② Up-counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by TMOD.
The input clock of timer 0 is selected from the external clock from TI0 pin and the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of TMOD register.
The input clock of timer 1 depends on the operation modes. When set to 16-bit timer mode, the overflow output of timer 0 is used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (match detection signal) of timer 0 according to the set value of TMOD register.

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

Operation mode is also set by TMOD register. When reset, it is initialized to TMOD<T10M1, 0>=00 whereby the up-counter is placed in the 8-bit timer mode.
The counting and stop and clear of up-counter can be controlled for each interval timer by the timer operation control register TRUN. When reset, all up-counters will be cleared to stop the timers.

③ Timer register

This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, matches the value of up-counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up-counter overflows.
Timer register TREG0 is of double buffer structure, each of which makes a pair with register buffer.
The timer flip-flop controll register TFFCR<DBEN> bit controls whether the double buffer structure should be enabled or disabled. It is disabled when <DBEN> = 0 and enabled when they are set to 1.
In the condition of double buffer enable state, the data is 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 <DBEN> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <DBEN> to 1, and write the following data in the register buffer.

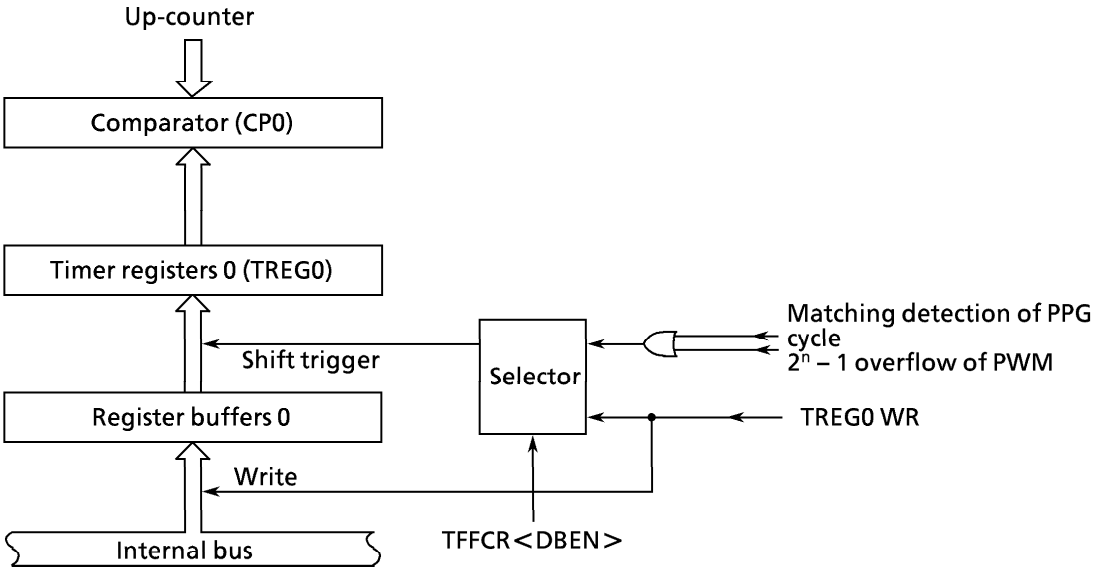


Figure 3.7.3 Configuration of Timer Register 0

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

The memory address of each timer register is as follows.

TREG0 : 000022H
TREG1 : 000023H

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

⑤ Timer flip-flop

The timer flip-flop (TFF1) is a flip-flop inverted by the match detect signal(8-bit comparator output). Inverting is disabled or enabled by the timer flip-flop control register TFFCR<TFF1IE>.
After reset operation, the value of TFF1 is undefined. Writing "01" or "10" to TFFCR<TFF1C1, 0> sets "0" or "1" to TFF1. Additionally, writing "00" to this bit inverts the value of TFF1. (software inversion)
TFF1 is output to TO1 pin (also used as P71). When using as the timer output, the timer flip-flop should be set by port 7 function register P7FC beforehand.

TRUN (0020H)

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

→ Count Operation

0	Stop and clear
1	Count

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

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol							PRCK1	PRCK0
Read / Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	Set to 1	Set to 0	Set to 1	Set to 0	Set to 0	Set to 0	select prescaler clock 00 : f_{FPH} 01 : (reserved) 10 : fc/16 11 : (reserved)	

→ select prescaler clock

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

} Clock divided by 4

Figure 3.7.4 Timer Operation Control Register / System Clock Control Register

TMOD (0024H)

	7	6	5	4	3	2	1	0
Bit symbol	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read / Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00 : 8 bit timer 01 : 16 bit timer 10 : 8 bit PPG 11 : 8 bit PWM		PWM cycle 00 : – 01 : 2^6-1 10 : 2^7-1 11 : 2^8-1		Source clock of timer1 00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		Source clock of timer0 00 : TI0 01 : ϕT1 10 : ϕT4 11 : ϕT16	

Prohibit Read Modify Write

Input clock of timer 0

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

Input clock of timer 1

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

Select PWM cycle

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

Set the operation mode of timer 0 and 1.

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

Figure 3.7.5 Timer Mode control Register (TMOD)

TFFCR (0025H)

	7	6	5	4	3	2	1	0
Bit symbol				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read / Write				R/W	W		R/W	
After reset				0	1	1	0	0
Function				Double buffer 0: Disable 1: Enable	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care ※ Always read as "11"		TFF1 Inversion trigger3 0: Disable 1: Enable	TFF1 Inversion source 0: Timer 0 1: Timer 1

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

	TMOD<T10M1,0> = 00	01	10	11
0	Inversion by timer0 match signal	16-bit timer mode Inversion by match signal	PPG mode Inversion by match signal of each timer 0 and timer 1	PWM mode Inversion by match and overflow signal of timer 0
1	Inversion by timer1 match signal			

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

Double Buffer Control of TREG0

0	Disable double Buffer
1	Enable double Buffer

Note : TFFCR<bit 7 to 5>, <bit 3, 2> are always read as "1".

Figure 3.7.6 Timer Flip-flop Control Register (TFFCR)

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

(1) 8-bit timer mode

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

① Generating interrupts in a fixed cycle

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

Example : To generate timer 1 interrupt every 10 μs at fc = 20 MHz, set each register in the following manner.

※ Clock Condition { clock gear : 1 (fc); prescaler clock : fc

	MSB 7 6 5 4 3 2 1 0 LSB	
TRUN	← - X - - - - 0 -	Stop timer 1, and clear it to "0".
TMOD	← 0 0 X X 1 0 - -	Set the 8-bit timer mode, and select ϕT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG1	← 0 0 0 1 1 0 0 1	Set the timer register 10 μs ÷ ϕT1 = 25 = 19H
INTET10	← 1 1 0 1 - - - -	Enable INTT1, and set it to "Level 5".
TRUN	← 1 X - - - - 1 -	Start timer 1 counting.

Note : ×; Don't care —; No change

Use the 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 (TFF1) is inverted at constant intervals, and its status is output to timer output pin (TO1).

Example : To output a 2.4 μs square wave pulse from TO1 pin at fc = 20 MHz, set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

※ Clock Condition { clock gear : 1 (fc); prescaler clock : f_{FPH}

	7 6 5 4 3 2 1 0	
TRUN	← – X – – – – 0 –	Stop timer 1, and clear it to "0".
TMOD	← 0 0 X X 0 1 – –	Set the 8-bit timer mode, and select φT1(0.4 μs at fc = 20 MHz) as the input clock.
TREG1	← 0 0 0 0 0 0 1 1	Set the timer register 2.4 μs ÷ φT1 ÷ 2 = 3.
TFFCR	← – – – – 1 0 1 1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P7CR	← X X X X – – 1 –	Select P71 as TO1 pin.
P7FC	← X X X X – – 1 X	
TRUN	← 1 X – – – – 1 –	Start timer 1 counting.

Note : X ; Don't care – ; No change

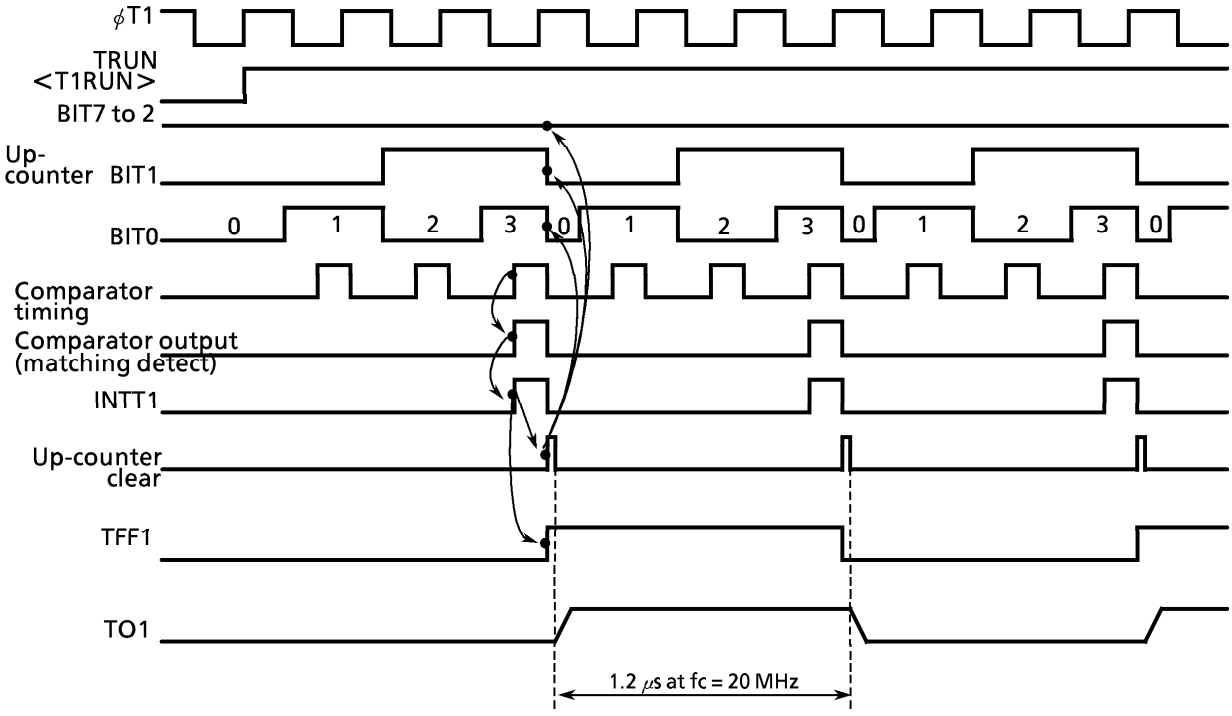


Figure 3.7.7 Square Wave (50% Duty) Output Timing Chart

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

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

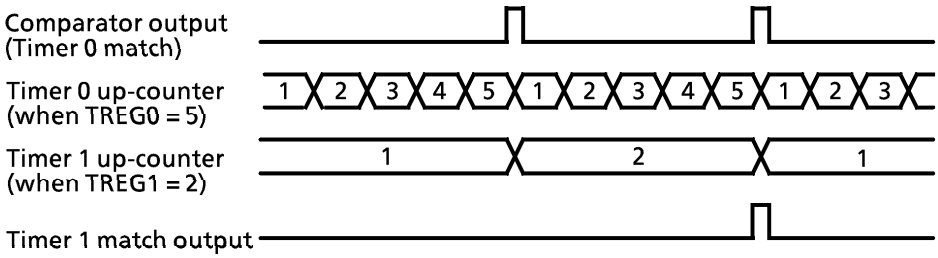


Figure 3.7.8 Timer 1 count up by timer 0

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1.
To make a 16-bit timer mode, set timer 0/timer 1 mode register TMOD<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, regardless of the set value of TMOD<T1CLK1,0>. 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, and the upper 8 bits are set by TREG1. Note that TREG0 always must be set first. (Writing data into TREG0 disables the comparator temporarily, and the comparator is restarted by writing data into TREG1.)

Setting example : To generate an interrupt INTT1 every 0.4 seconds at fc = 20 MHz, set the following values for timer registers TREG0 and TREG1.

※ Clock Condition { 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 TREG1 = F4H and TREG0 = 24H, respectively.

The comparator match signal is output from timer 0 each time the up-counter UC0 matches TREG0, where the up-counter UC0 is not be cleared.
With the timer 1 comparator, the match detect signal is output at each comparator timing when up-counter UC1 and TREG1 values match. When the match detect signal is output simultaneously from both comparators of timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

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

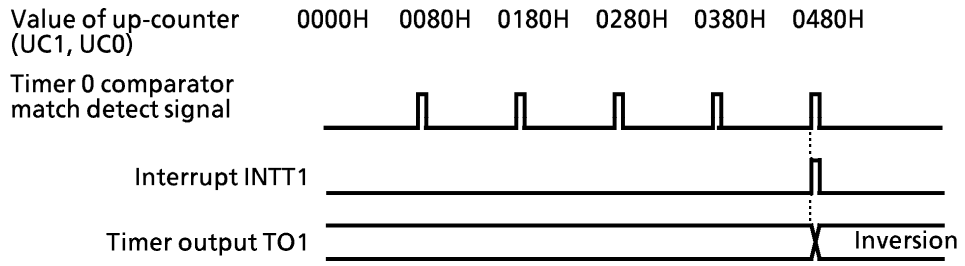


Figure 3.7.9 Timer output by 16-bit timer mode

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

Square wave pulse can be generated at any frequency and duty by timer 0. The output pulse may be either low-active or high-active.
In this mode, timer 1 cannot be used.
Timer 0 outputs pulse to TO1 pin (also used as P71).

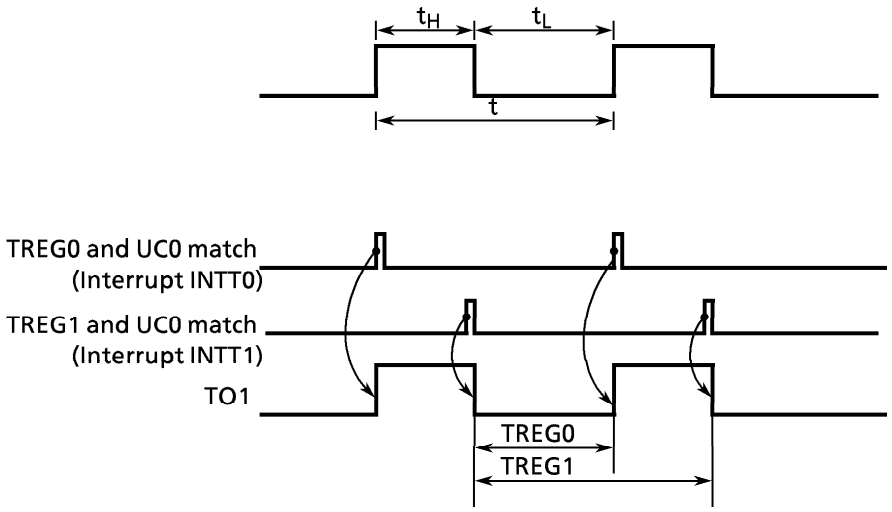


Figure 3.7.10 8 bit PPG output waveforms

In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up-counter (UC0) matches the timer registers TREG0 and TREG1.
However, it is required that the set value of TREG0 is smaller than that of TREG1.
Though the up-counter (UC1) of timer 1 is not used in this mode, UC1 should be set for counting by setting TRUN<T1RUN>to 1.
Figure 3.7.11 shows the block diagram for this mode.

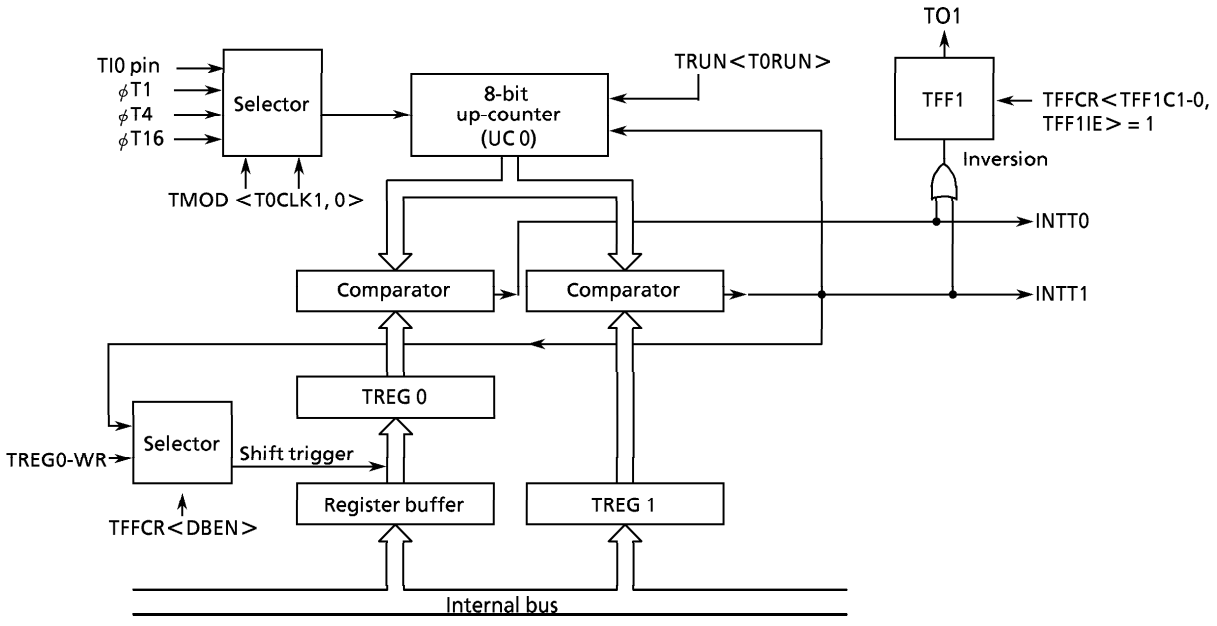


Figure 3.7.11 Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TREG0 is enabled in this mode, the value of register buffer will be shifted in TREG0 each time TREG1 matches UC0.
Use of the double buffer makes easy the handling of low duty waves (when duty is varied).

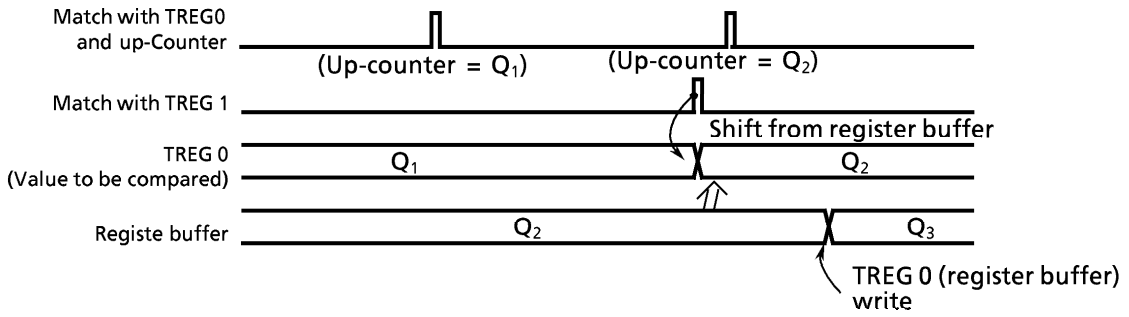
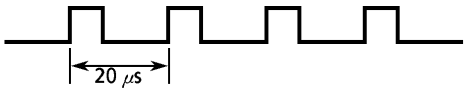


Figure 3.7.12 Operation of Register buffer

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



※ Clock Condition { 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 1 (TREG1) to TREG1 = 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 0 (TREG0) to TREG0 = 10 = 0AH.

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

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

(4) 8-bit PWM Output mode

This mode is valid only for timer 0. In this mode, maximum 8-bit resolution of PWM pulse can be output.
PWM pulse is output to TO1 pin (also used as P71) when using timer 0. Timer 1 can also be used as 8-bit timer.
Timer output is inverted when up-counter (UC0) matches the set value of timer register TREG0 or when 2^n-1 (n = 6, 7, or 8; specified by TMOD<PWMM1,0>) 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) $\neq 0$

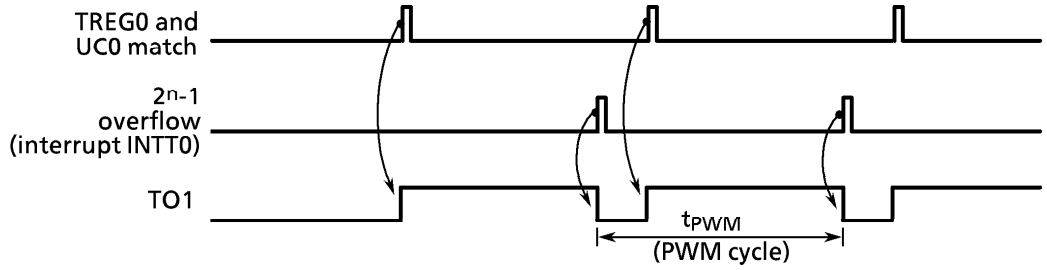


Figure 3.7.13 8-bit PWM waveforms

Figure 3.7.14 shows the block diagram of this mode.

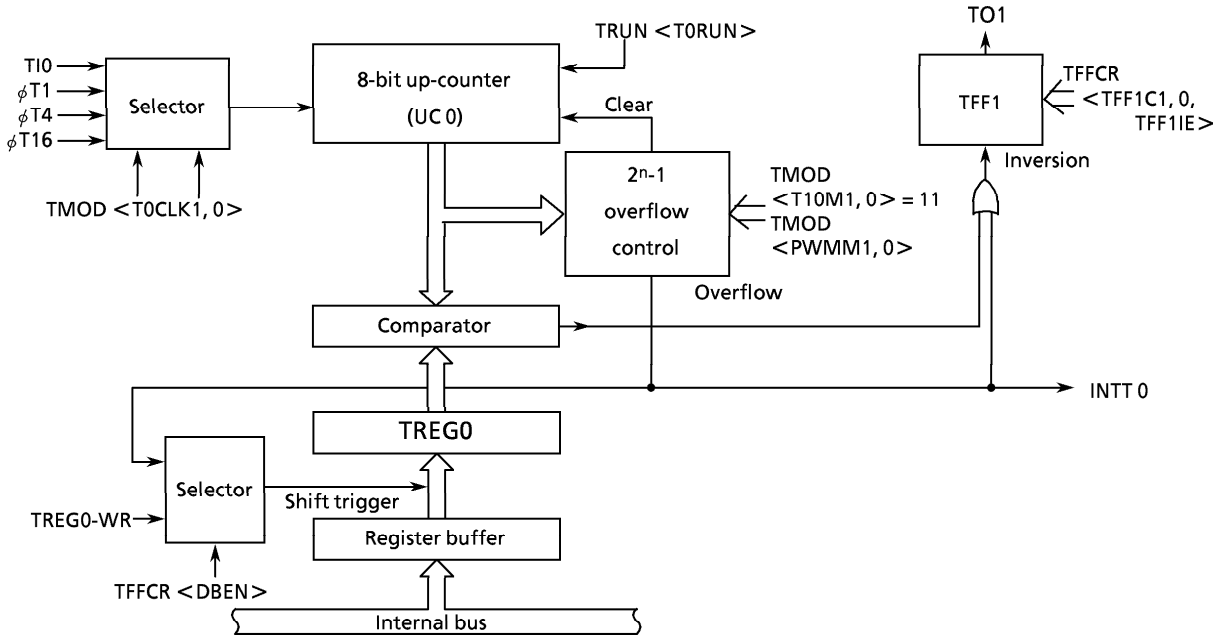


Figure 3.7.14 Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TREG0 if 2^n-1 overflow is detected when the double buffer of TREG0 is enabled.
Use of the double buffer makes easy the handling of small duty waves.

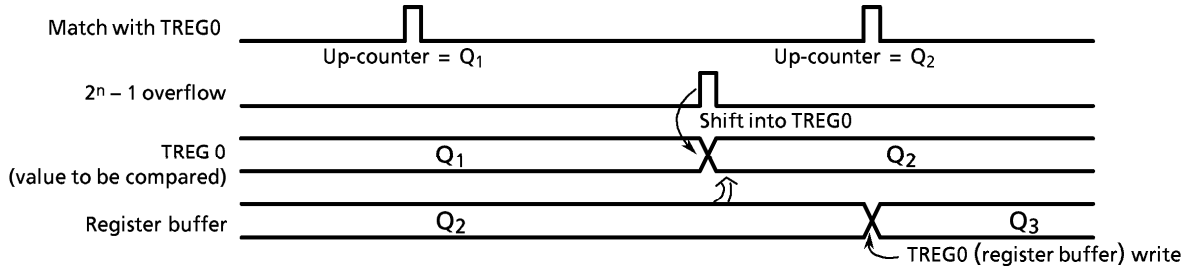
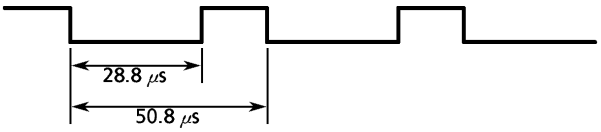


Figure 3.7.15 Operation of Register buffer

Example : To output the following PWM waves to TO1 pin at fc = 20 MHz.



※ Clock Condition { clock gear : 1 (fc); prescaler clock : f_{FPH}

To realize 50.8 μs of PWM cycle by ϕT1 = 0.4 μs (at fc = 20 MHz),
50.8 μs ÷ 0.4 μs = 127 = 2^n-1
Consequently, n should be set to 7.
As the period of low level is 28.8 μs, for ϕT1 = 0.4 μs,
set the following value for TREG0.
28.8 μs ÷ 0.4 μs = 72 = 48H

	MSB 7 6 5 4 3 2 1 LSB 0	
TRUN	← – X – – – – – 0	Stop timer 0, and clear it to "0".
TMOD	← 1 1 1 0 – – 0 1	Set 8-bit PWM mode (cycle: 2^7 – 1) and select ϕT1 as the input clock.
TREG0	← 0 1 0 0 1 0 0 0	Writes "48H".
TFFCR	← X X X X 1 0 1 X	Clears TFF1, enable the inversion and double buffer.
P7CR	← X X X X – – 1 –	Set P71 as the TO1 pin.
P7FC	← X X X X – – 1 X	
TRUN	← 1 X – – – – – 1	Start timer 0 counting.

Note : X ; Don't care – ; No change

Table 3.7.2 PWM Cycle

at fc = 20 MHz

select prescaler clock <PRCK1, 0>	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})	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	26.11 ms
10 (fc/16 clock)	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) Table 3.7.3 shows the list of 8-bit timer modes.

Table 3.7.3 Timer Mode Setting Registers

Register name	TMOD				TFFCR
Name of function in register	T10M	PWMM	T1CLK	T0CLK	TFF1IS
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, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 2 channels	00	–	Lower timer match: ϕT1, ϕT16, ϕT256 (00, 01, 10, 11)	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG × 1channel	10	–	–	External clock, ϕT1, ϕT4 , ϕT16 (00, 01, 10, 11)	–
8-bit PWM × 1channel	11	2^6-1, 2^7-1, 2^8-1 (01, 10, 11)	–	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 1channel	11	–	ϕT1, ϕT16 , ϕT256 (01, 10, 11)	–	Output disabled

Note : - ; Don't care

3.8 8-Bit PWM Timers

The TMP93CS42A has two built-in 8-bit PWM timers (timers 2 and 3).
They have the following operating mode.

Figure 3.8.1, 3.8.2 are block diagram of 8-bit PWM timer (timers 2 and 3). PWM timers consist of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Two timer flip-flops (TFF2 for timer 2 and TFF3 for timer 3) are provided.
Input clocks ϕP1, ϕP4, and ϕP16 for the PWM timers can be obtained using the 5 bitbuilt-in prescaler.(PWM dedicated prescaler)
PWM timer operating mode and timer flip-flops are controlled by four control registers (P0MOD, P1MOD, PFFCR, and TRUN).
PWM timer 0 and 1 can be used independently.
All PWM timer operate in the same manner, and thus only the operation of PWM timer 0 will be explained below.

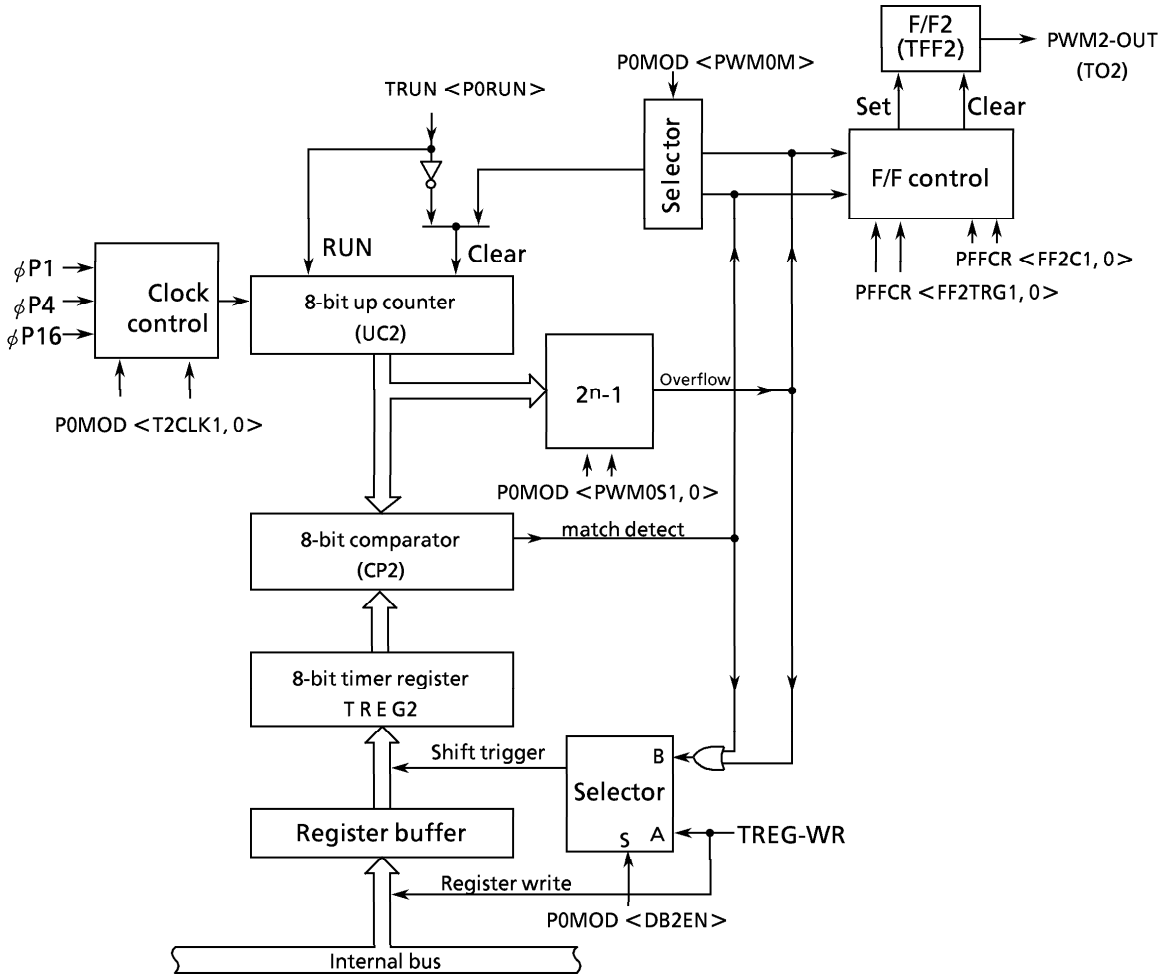


Figure 3.8.1 Block diagram of 8-bit PWM timer 0 (timer 2)

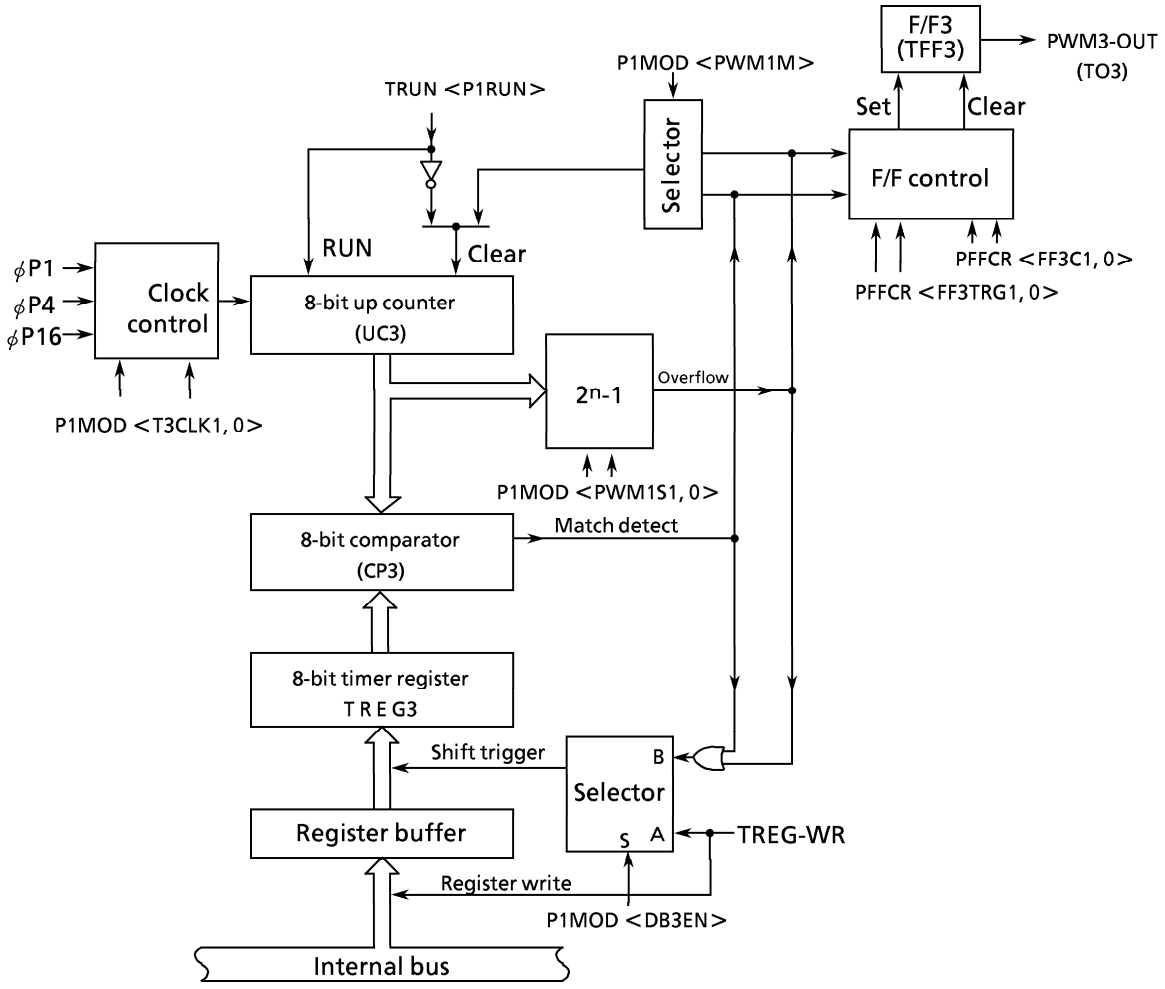


Figure 3.8.2 Block diagram of 8-bit PWM timer 1 (timer 3)

① Prescaler

There are 5 bit prescaler and prescaler clock selection register to generate input clock for 8 bit PWM Timer 0, 1.
Figure 3.8.3 shows the block diagram. Table 3.8.1 shows prescaler clock resolution into 8 bit PWM Timer 0, 1.

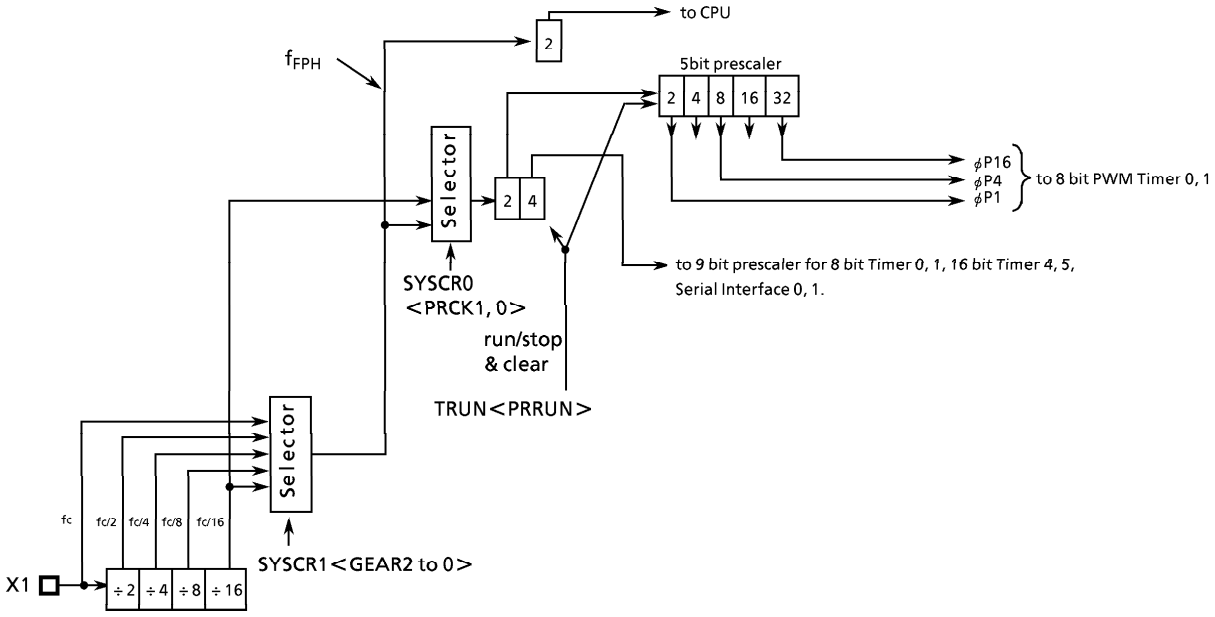


Figure 3.8.3 The Block Diagram of Prescaler

Table 3.8.1 Prescaler Clock Resolution to 8 bit PWM Timer 0, 1

at fc = 20 MHz

Select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Prescaler Clock Resolution		
		φP1	φP4	φP16
00 (f_{FPH})	000 (fc)	$fc/2^2$ (0.2 μs)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)
	001 (fc/2)	$fc/2^3$ (0.4 μs)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)
	010 (fc/4)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)
	011 (fc/8)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)
	100 (fc/16)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)
10 (fc/16 clock)	XXX	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)

XXX : Don't care

The clock selected among f_{FPH} clock and fc / 16 clock is divided by 2 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1, 0>.
Resetting sets <PRCK1, 0> to "00" selects the f_{FPH} clock input divided by 2. The register TRUN <PRRUN> which controls this prescaler is also used as the other timers. So, this prescaler can not be operated independently.
The 8 bit PWM Timer0, 1 selects between 3 clock inputs : ϕP1, ϕP4, and ϕP16 among the prescaler outputs.
This prescaler also can be run or stopped by TRUN<PRRUN> described of the 8 bit Timer. Counting starts when <PRRUN> is set to "1". The prescaler is cleared zero and stops operation when <PRRUN> is set to "0". Resetting clear <PRRUN> to "0" and stops the prescaler.
When the IDLE1 mode (only the oscillator operates) is used, set TRUN <PRRUN> to '0' to reduce the power consumption of the prescaler before the "HALT" instruction is executed.

② Up-counter

An 8-bit binary counter which counts up using the input clock specified by PWM mode register P0MOD<T2CLK1, 0>.
The input clock for the up-counter is selected from the internal clocks ϕP1, ϕP4, and ϕP16 (PWM dedicated prescaler output) depending on the <T2CLK1, 0>.
Operating mode is set by P0MOD <PWM0M>. At reset, it is initialized to '0' , thus, the up-counter is placed in PWM mode. In PWM mode, the up-counter is cleared when a 2n-1 overflow occurs; in timer mode, the up-counter is cleared at compare and match.
Count / stop & clear of the up-counter can be controlled for each PWM timer using the timer operation control register TRUN. Resetting clears all up-counters and stops timers.

③ Timer register

The 8-bit register is used for setting an interval time. When the value set in the timer register (TREG2) matches the value in the up-counter, the match detect signal of the comarator becomes active.
Timer register TREG2 is each paired with register buffer to make a double buffer structure.
TREG2 is controlled double buffer enable / disable by P0MOD <DB2EN> : disabled when <DB2EN> = 0, enabled when <DB2EN> = 1.

Data is transferred from register buffer to timer register when a 2^n-1 overflow occurs in PWM mode, or when compare and match occurs in 8-bit timer mode. That is, with a PWM timer, the timer mode can be operated in double buffer enable state, unlike PWM mode and timer mode for timers 0 and 1.
At reset, <DB2EN> is initialized to 0 to disable double buffer. The same data value is written to both the register buffer and the timing register. To use double buffer, write the data in the timer register at first, then set <DB2EN> to 1, and write the following data in the register buffer.

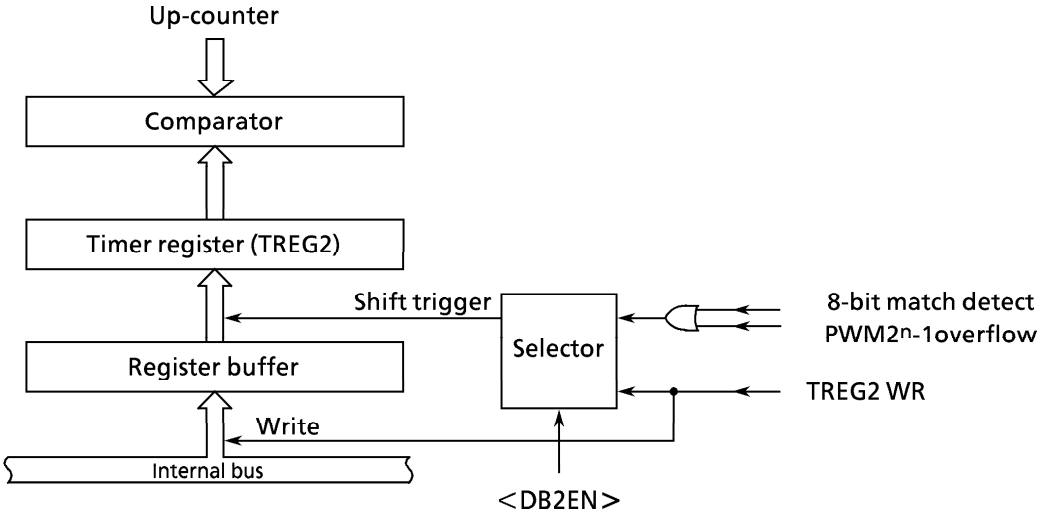


Figure3.8.4 Structure of Timer Registers 2

Memory addresses of the timer registers are as follows:

TREG2 : 000026H
TREG3 : 000027H

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

Register buffer values can be read when reading the above addresses. Thetimer register is write only, and it can not read.

④ Comparator

Compares the value in the up-counter with the value in the timer register (TREG2). When they match, the comparator outputs the match detect signal. In timer mode, the comparator clears the up-counter to 0 at compare and match. It also inverts the value of the timer flip-flop if timer flip-flop invert is enabled.

⑤ Timer flip-flop

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

P0MOD (0028H)

	7	6	5	4	3	2	1	0
Bit symbol	FF2RD	DB2EN		PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0
Read / Write	R	W						
After reset	–	0		0	0	0	0	0
Function	Flip-flop (F/F2) output data	1: Double Buffer2 Enable		0: PWM mode 1: Timer mode	00: φP1 01: φP4 10: φP16 11: Don't care		00: 2^{6}-1 01: 2^{7}-1 10: 2^{8}-1 11: Don't care	

Read-modify-write is prohibited.

Select PWM0 cycle

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

Select PWM0 input clock

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

select PWM0 mode

0	PWM mode
1	8-bit timer mode

control double buffer

0	Disable
1	Enable

PWM timer Flip-flop2 (TFF2) output value (TO2)

Figure 3.8.5 8-Bit PWM0 Mode Control Register

P1MOD (0029H)

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

Read-modify-write is prohibited.

Select PWM1 cycle

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

Select PWM1 input clock

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

Select PWM1 mode

0	PWMmode
1	8-bit timer mode

Control double buffer

0	Disable
1	enable

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

Figure 3.8.6 8-Bit PWM1 Mode Control Register

PFFCR (002AH)

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

Select PWM timer F/F2 (TFF2) trigger

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

Control PWM timerF/F2 (TFF2)

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

Select PWM timer F/F3 (TFF3) trigger

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

Control PWM timerF/F3 (TFF3)

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

Figure 3.8.7 8-Bit PWM F / F Control Register

TRUN (0020H)

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

→ Count Operation

0	Stop and clear
1	Count

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

Note : TRN<bit6> is always read as "1".

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol							PRCK1	PRCK0
Read / Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	Set to 1	Set to 0	Set to 1	Set to 0	Set to 0	Set to 0	select prescaler clock 00 : f_{FPH} 01 : (reserved) 10 : fc/16 11 : (reserved)	

→ select prescaler clock

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

} divided clock

Figure 3.8.8 Timer Operation Control Register / System Clock Control Register

The following explains PWM timer operations.

(1) PWM timer mode

PWM output changes under the following two conditions.

Condition 1:

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

Condition 2:

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

The up-counter (UC2) is cleared by a 2^n-1 counter overflow.
The PWM timer can output 0% to 100% duty pulses because a 2^n-1 counter overflow has a higher priority. That is, to obtain 0% output (always low), the mode used to set TFF2 to 0 due to overflow (PFFCR<FF2TRG1,0> = 1,0) must be set and 2^n-1 (Value for overflow) must be set in TREG2. To obtain 100% output (always high), the mode must be changed: PFFCR<FF2TRG1, 0> = 1, 1 then the same operation is required.

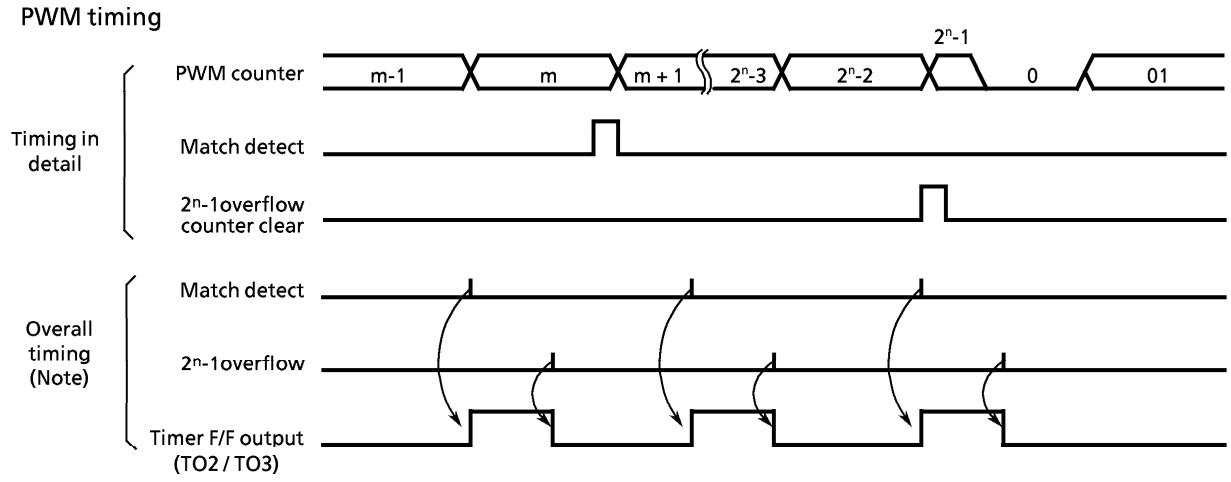


Figure3.8.9 Output Waves in PWM Timer Mode

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

Figure 3.8.10 is a block diagram of this mode.

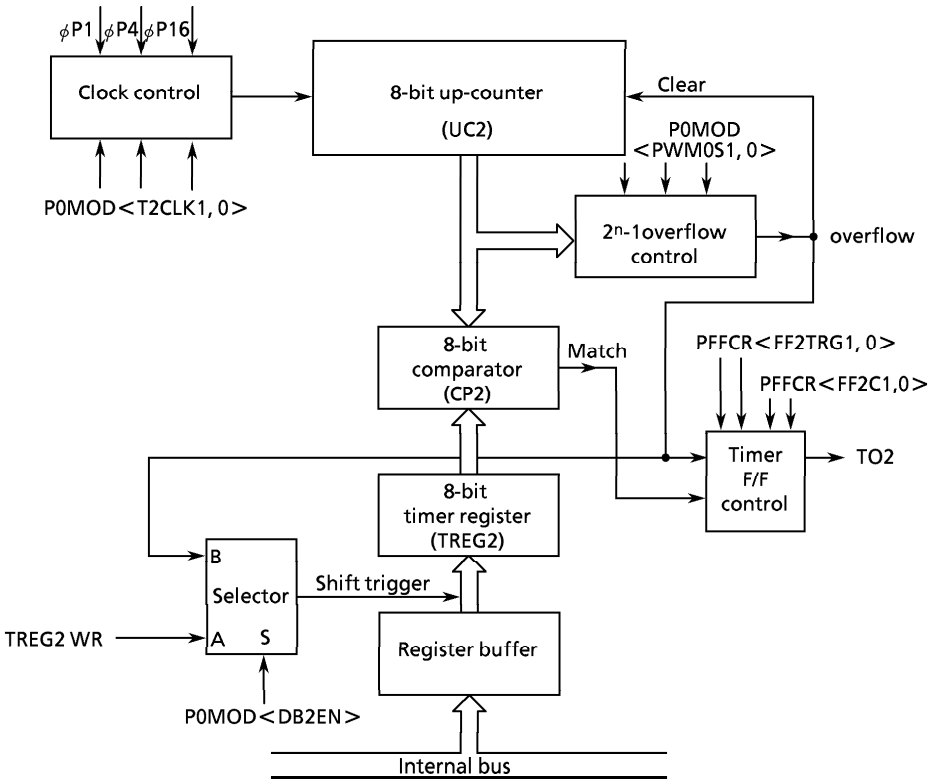


Figure 3.8.10 Block Diagram of PWM Timer Mode (PWM0)

In this mode, enabling double buffer is very useful. The register buffer value shifts into TREG2 when a 2^n-1 overflow is detected, when double buffer is enabled.
Using double buffer makes handling small duty waves easily.

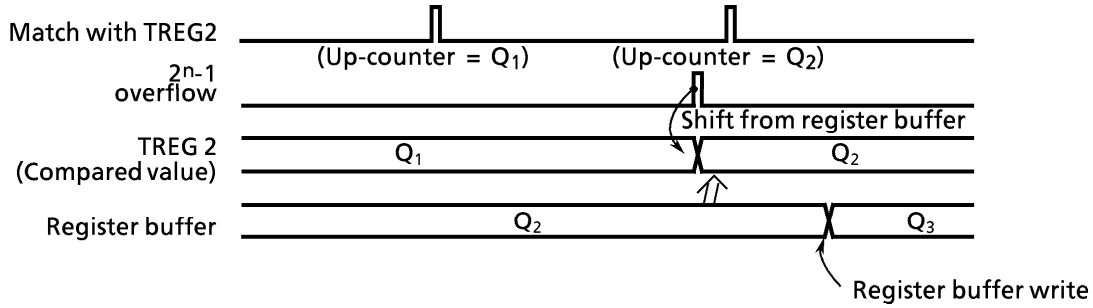
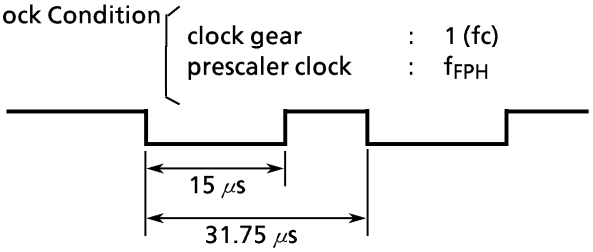


Figure 3.8.11 Register Buffer Operation

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



To implement 31.75 μs PWM cycle by φP1 = 0.25 μs (at fc = 16 MHz)
31.75 μs ÷ 0.25 μs = 127 = 2^7-1.
Consequently, set n to 7.
Since the low level cycle = 15 μs; for φP1 = 0.25 μs
15 μs ÷ 0.25 = 60 = 3CH
set the 3CH in TREG2.

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

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

Table 3.8.2 PWM Cycle at fc = 20 MHz

select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	PWM Cycle								
		2^6 - 1			2^7 - 1			2^8 - 1		
		φP1	φP4	φP16	φP1	φP4	φP16	φP1	φP4	φP16
00 (f_{FPH})	000 (fc)	12.6 μs	50.4 μs	201.6 μs	25.4 μs	101.6 μs	406.4 μs	51.0 μs	204.0 μs	816 μs
	001 (fc/2)	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
	010 (fc/4)	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
	011 (fc/8)	100.8 μs	403.2 μs	1.61 ms	203.2 μs	812.8 μs	3.25 ms	408.0 μs	1.63 ms	6.53 ms
	100 (fc/16)	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.50 ms	816.0 μs	3.26 ms	13.06 ms
10 (fc/16 clock)	XXX	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.50 ms	816.0 μs	3.26 ms	13.06 ms

XXX : Don't care

(2) 8-bit timer mode

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

① Generating a 50 % square wave

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

Example : To output a 2.4 μs square wave at fc = 20 MHz from TO2 pin, set registers as follows.

※ Cock Condition
- system clock : 1 (fc)
- prescaler clock : f_{FPH}

	7 6 5 4 3 2 1 0	
TRUN	← - X - - - 0 - -	Stops PWM0 and clears it to 0.
P0MOD	← X 0 1 1 0 0 X X	Sets 8-bit timer mode and selects ϕP1 (0.2 μs) as the input clock
TREG2	← 0 0 0 0 0 1 1 0	Sets 2.4 μs ÷ 0.2 μs ÷ 2 = 6 in the timer register.
PFFCR	← - - - - 1 0 0 1	Clears TFF2 to 0 and inverts using comparator output.
P7CR	← X X X X - 1 - -	Sets P72 as TO2 pin.
P7FC	← X X X X - 1 - X	
TRUN	← 1 X - - - 1 - -	Starts counting PWM0.

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

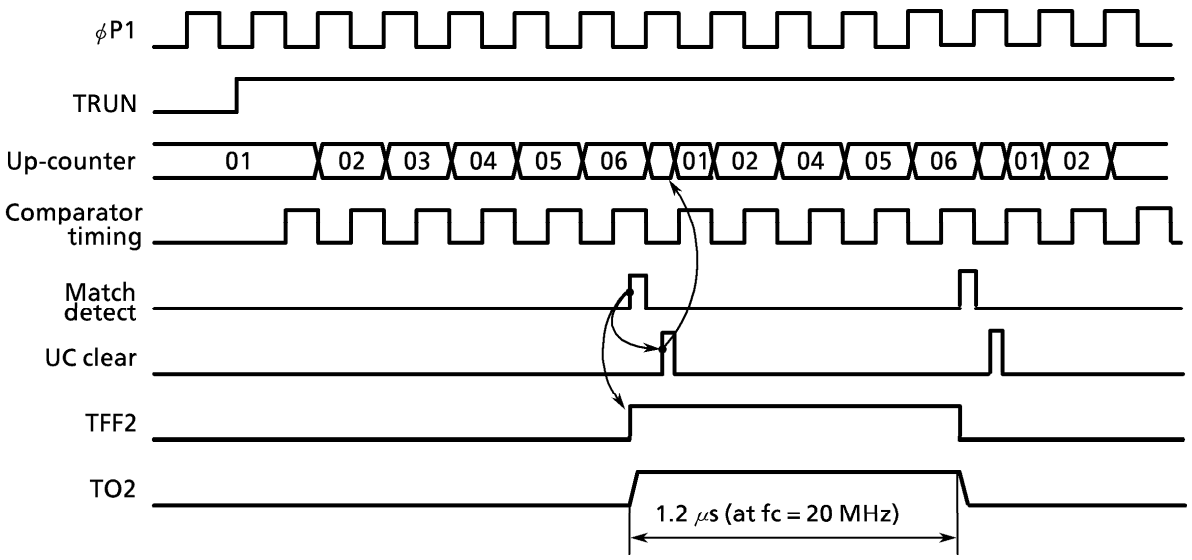


Figure 3.8.12 Square Wave (50 % Duty) Output Timing Chart

This mode is as shown in Figure 3.8.13 below.

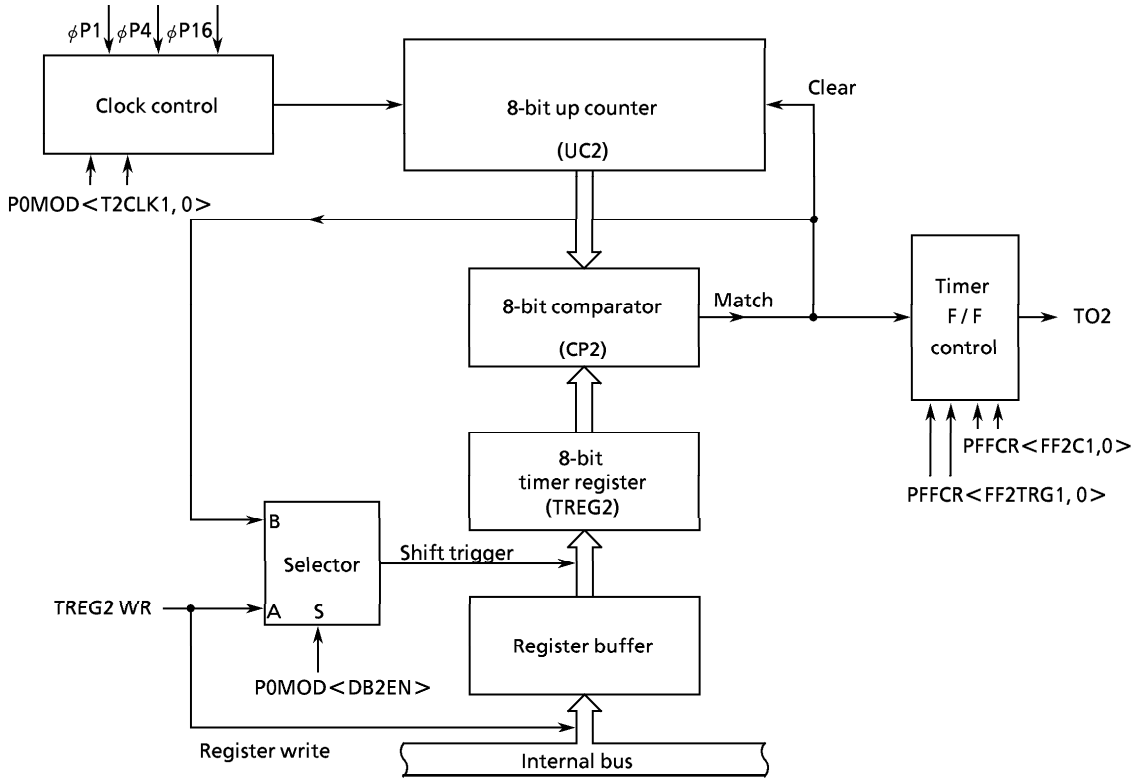


Figure 3.8.13 Block Diagram of 8-bit Timer Mode

3.9 16-bit Timers

TMP93CS42A contains two (timer 4 and timer 5) multifunctional 16-bit timer / event counters with the following operation modes.

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) mode
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

Timer / event counter consists of 16-bit up-counter, two 16-bit timer registers(One of them applies double-buffer), two 16-bit capture registers, two comparators, capture input controller, and timer flip-flop and the control circuit.
Timer / event counter is controlled by 4 control registers: T4MOD/T5MOD, T4FFCR / T5FFCR, TRUN and T45CR.

Figure 3.9.1, 3.9.2 shows the block diagram of 16-bit timer / event counter (timer 4 and timer 5).

Timer 4 and 5 can be used independently.
All timer operate in the same manner except the following points, and thus only the operation of Timer 4 will be explained below.

(Different Points between Timer 4 and 5)

	Timer 4	Timer 5
Timer Out Pin	TO4 pin (TFF4) TO5 pin (TFF5)	TO6 pin (TFF6) No TO7 pin
Different Phased Palse Output Mode	Yes	No (no TO7 pin)

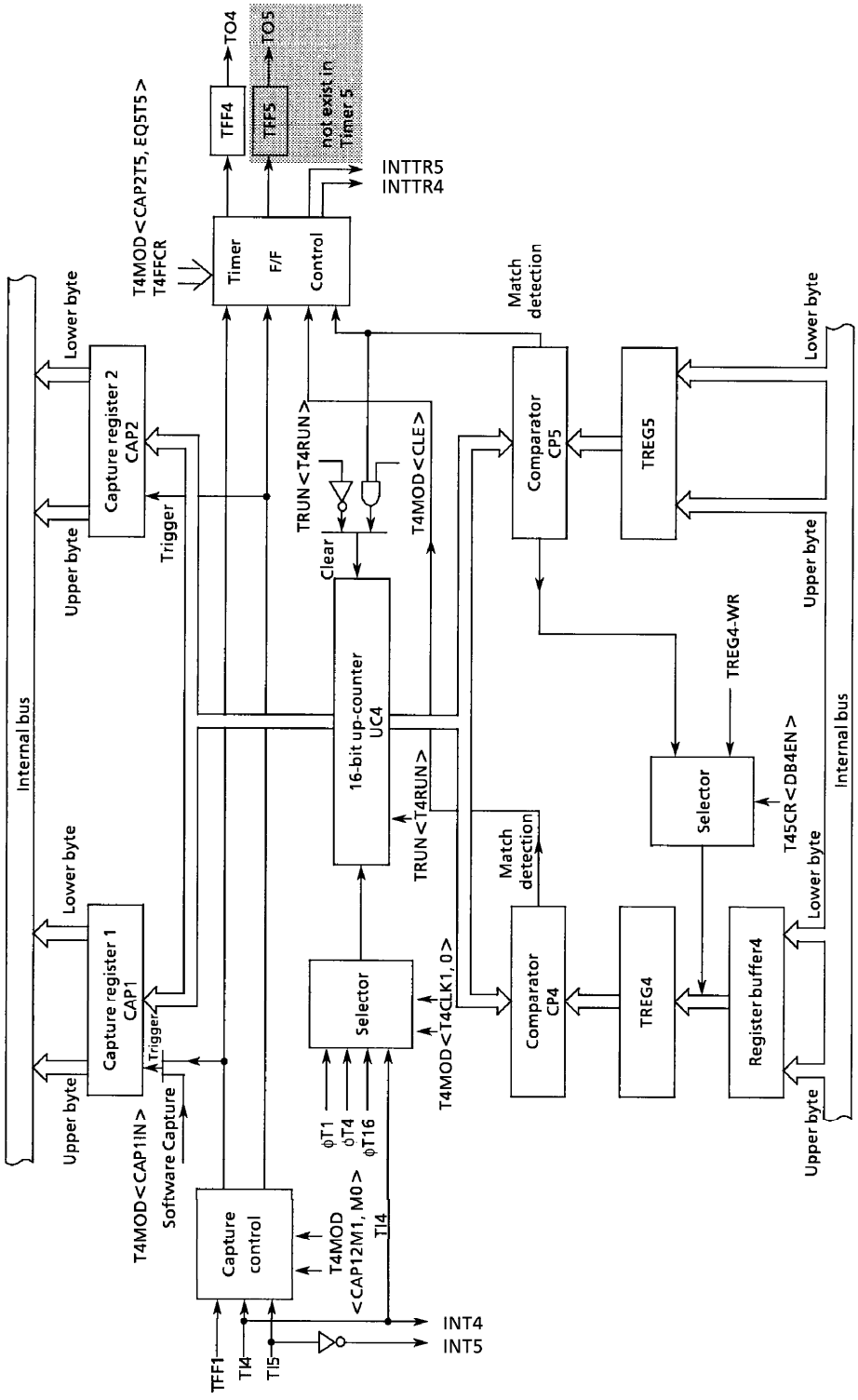


Figure 3.9.1 Block Diagram of 16-Bit Timer (Timer 4)

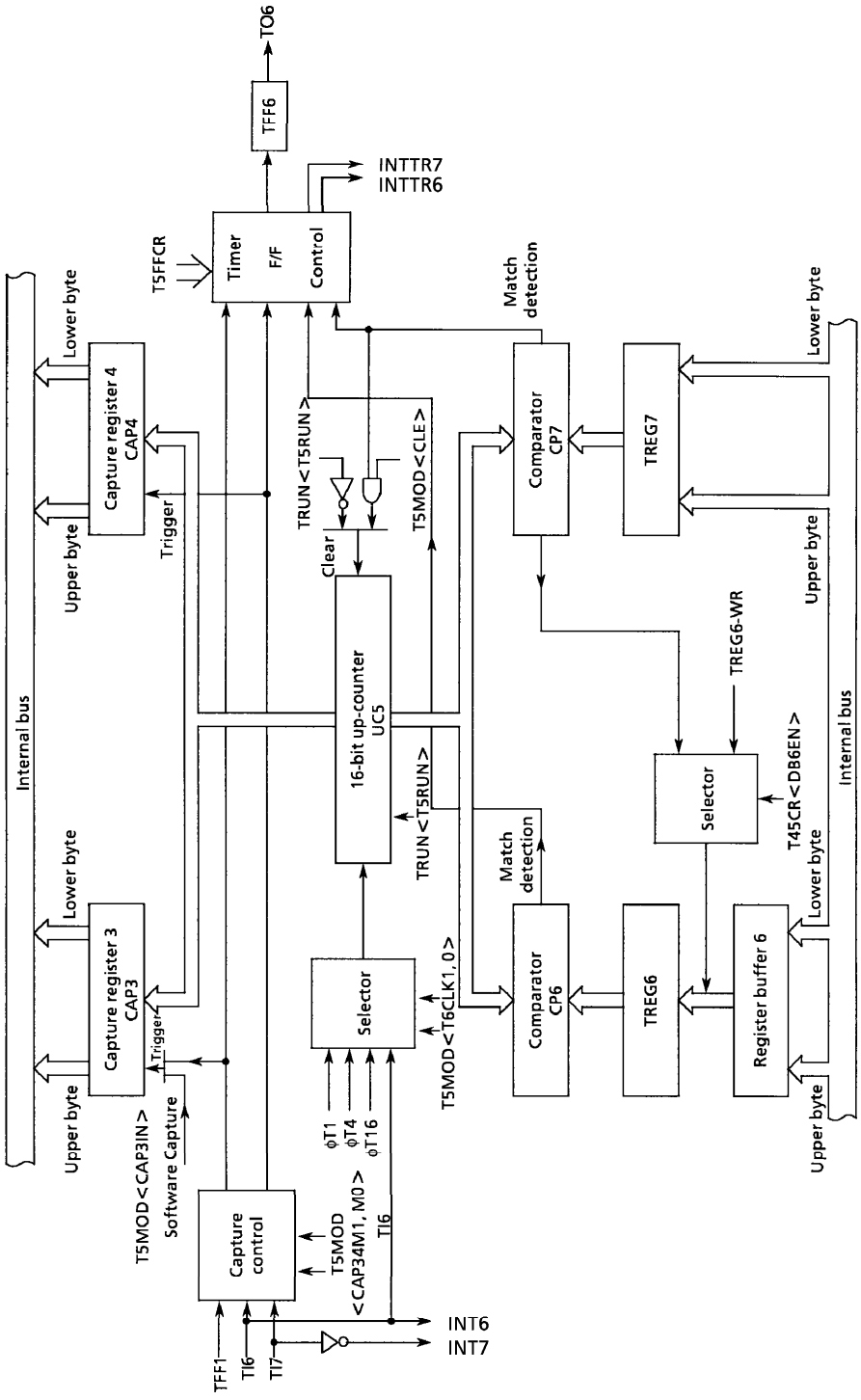


Figure 3.9.2 Block Diagram of 16-Bit Timer (Timer 5)

T4MOD (0038H)

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

Timer 4 input clock

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

Clearing the up-counter UC4

0	Clear disable
1	Clear by match with TREG5.

Figure 3.9.3 16-Bit Timer Mode Controller Register (T4MOD) (1/2)

T4MOD (0038H)

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

Capture timing of timer4

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

Software capture

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

Timer flip-flop 5 (TFF5) invert trigger

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

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

Figure 3.9.3 16-Bit Timer Mode Controller Register (T4MOD) (2/2)

T4FFCR (0039H)

	7	6	5	4	3	2	1	0
Bit symbol	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	1	1	0	0	0	0	1	1
Function	00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care Always read as "11".		TFF4 invert trigger 0: Disable trigger 1: Enable trigger 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	00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care ※ Always read as "11"	

Timer flip-flop 4 (TFF4) control

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

Timer flip-flop 4 (TFF4) invert trigger

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

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

Timer flip-flop 5 (TFF5) control

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

Figure 3.9.4 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			W	R/W		R/W	R/W	
After reset			1	0	0	0	0	0
Function			0: Soft-Capture 1: Don't care	Capture timing 00: Disable INT6 occurs at rise edge. 01: TI6↑ TI7↑ INT6 occurs at rise edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC5 Clear Enable	Timer 5 source clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 5 input clock

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

Clearing the up-counter UC5

0	Clear disable
1	Clear by match with TREG7

Figure 3.9.5 16-Bit Timer Mode Control Register (T5MOD) (1/2)

T5MOD (0048H)

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

Timer 5 Capture timing

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

Software capture

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

Figure 3.9.5 16-Bit Timer Mode Control Register (T5MOD) (2/2)

T5FFCR (0049H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
Read/Write			R/W	R/W	R/W	R/W	W	
After reset			0	0	0	0	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 ※ Always read as "11"	
			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 (Always read as "11") .

Timer flip-flop 6 (TFF6) invert trigger

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

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

Figure 3.9.6 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	Warming Up Timer input 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

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 : T45CR<bit6 to 4>is always read as "1".

Figure 3.9.7 16-Bit Timer Trigger Control Register (T45CR)

TRUN (0020H)

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

Count Operation

0	Stop and clear
1	Count

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

Note : TRUN<bit6> is always read as "1".

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol							PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	Set to 1.	Set to 0.	Set to 1.	Set to 0.	Set to 0.	Set to 0.	select prescaler clock 00 : f_{FPH} 01 : (reserved) 10 : fc/16 11 : (reserved)	

select prescaler clock

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

Clock divided by 4

Figure 3.9.8 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, 16 bit Timer 4,5 and serial Interface 0,1. Figure 3.9.9 shows the block diagram. Table 3.9.1 shows prescaler clock resolution into 8, 16 bit Timer.

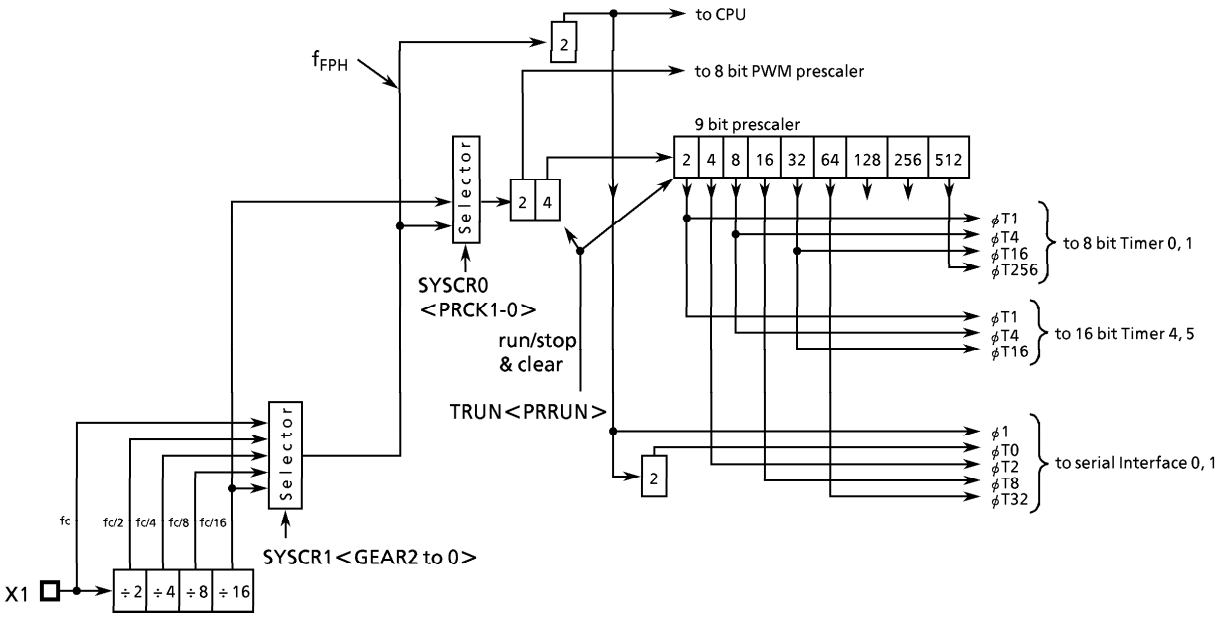


Figure 3.9.9 The Block Diagram of Prescaler

Table 3.9.1 Prescaler Clock Resalation to 8, 16 bit Timer

at fc = 20 MHz

select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Prescaler Clock Resolution			
		φT1	φT4	φT16	φT256
00 (f_{FPH})	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)	$fc/2^{15}$ (1.64 ms)
10 (fc/16 clock)	XXX	$fs/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.64 ms)

XXX : Don't care

16-bit timer

8-bit timer

The clock selected among f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0 <PRCK1, 0>.
Resetting sets <PRCK1, 0> to "00"selects the f_{FPH} clock input divided by 4.

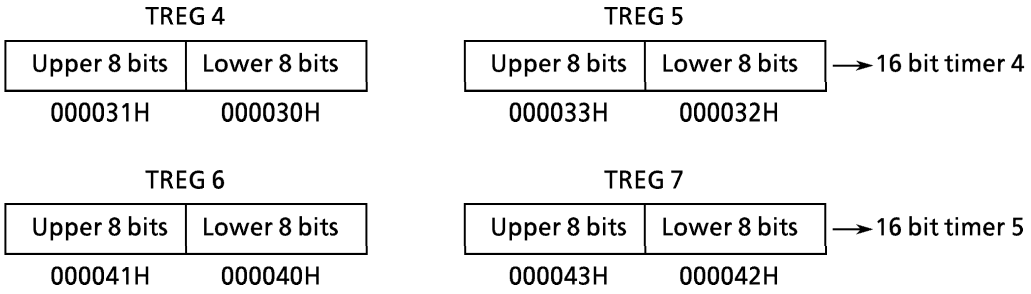
The 16 bit Timer selects between 3 clock inputs : ϕT1, ϕT4, and ϕT16 among the prescaler outputs.
This prescaler can be run or stopped by the timer operation control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to "1". The prescaler is cleared zero and stops operation when <PRRUN> is set to "0".
Resetting clear <PRRUN> to "0" and stops the prescaler.
When the IDLE1 mode (only the oscillator operates) is used, set TRUN <PRRUN> to '0' to reduce the power consumption of the prescaler before the "HALT" instruction is executed.

② Up-counter

The up counter 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 P80/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 UC will be cleared to zero each timer it coincides matches the timer register TREG5. The "clear enableADisable" is set by T4MOD<CLE>.
If clearing is disabled, the counter operates as a free-running counter.

③ Timer register

These two 16-bit registers are used to set the counter value. 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 data transfer instruction or using 1-byte data transfer instruction twice for lower 8 bits and upper 8 bits in order.

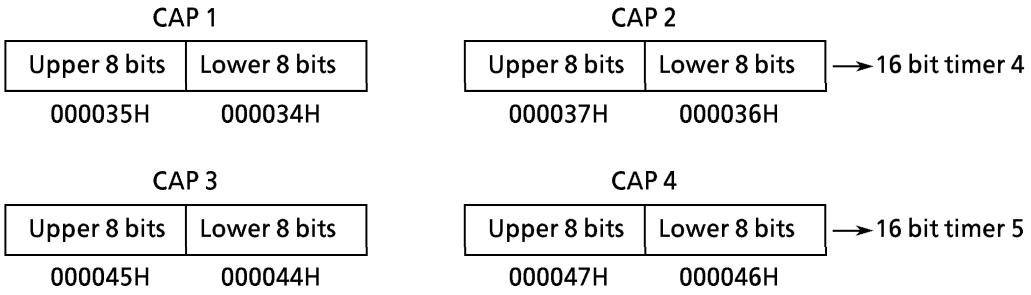


TREG4 timer register is a double buffer structure, which is paired with register buffer. The timer control register T45CR<DB4EN> controls whether the double buffer structure should be enabled or disabled. : disabled when <DB4EN> = 0, while enabled when <DB4EN> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up-counter (UC4) and timer register TREG5.
When reset, it will be initialized to <DB4EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN> = 1, and then write the following data in the register buffer.
TREG4 and register buffer are allocated to the same memory addresses 000030H/000031H. When <DB4EN> = 0, the same value will be written in both TREG4 and register buffer. When <DB4EN> = 1, the value is written into only the register buffer.

④ Capture Register

These 16-bit registers are used to hold the values of the up-counter.
Data in the capture registers should be read by a 2-byte data load instruction or two 1-byte data load instruction, from the lower 8 bits followed by the upper 8 bits.



⑤ Capture Input Control

This circuit controls the timing to latch the value of up-counter UC4 into the capture register(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 rising edge of TI4 pin (also used as P80/INT4) input, while data is loaded to CAP2 at the rising edge of TI5 pin (also used as P81/$\overline{\text{INT5}}$) input. (Time difference measurement)

- When T4MOD<CAP12M1, 0> = 10

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

- When T4MOD<CAP12M1, 0> = 11

  Data is loaded to CAP1 at the rising edge of timer flip-flop TFF1, while to CAP2 at the falling 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 (INTT4, INTT5) respectively.
The up-counter UC4 is cleared only when UC4 matches TREG5. (The clearing of up-counter UC4 can be disabled by setting T4MOD<CLE> = 0.)

⑦ Timer Flip-flop (TFF4)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable / enable of inversion can be set for each element by T4FFCR<CAP2T4, CAP1T4, EQ5T4, EQ4T4>.

TFF4 will be inverted when "00" is written in T4FFCR<TFF4C1,0>. Also it is set to "1" when "01" is written, and cleared to "0" when "10" is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P82). TFF4 is undefined on reset.

⑧ Timer Flip-flop (TFF5)

This flip-flop is inverted by the match detect signal between the up-counter (UC4) and the timer register TREG5 and the latch signal to the capture register CAP2. Disable/enable of inversion can be set for each element by T4MOD<CAP2T5, EQ5T5>. TFF5 will be inverted when "00" is written in T4FFCR<TFF5C1,0>. Also it is set to "1" when "01" is written, and cleared to "0" when "10" is written. The value of TFF5 can be output to the timer output pin TO5 (also used as P83). TFF5 is undefined on reset.

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

(1) 16-bit Timer Mode

Generating interrupts at fixed intervals

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

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

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

(2) 16-bit Event Counter Mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.
The counter counts at the rising edge of TI4 pin input.
TI4 pin can also be used as P80/INT4.

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

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

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

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

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

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

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

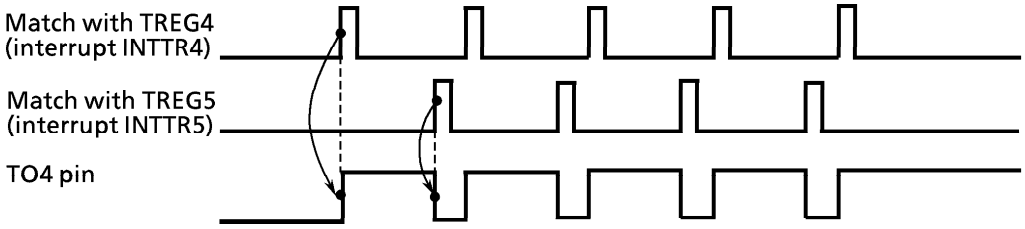


Figure 3.9.10 Programmable Pulse Generation (PPG) Output Waveforms

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.

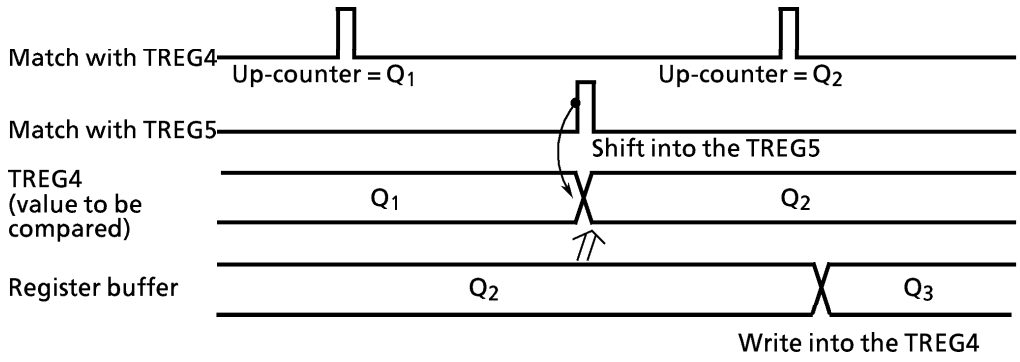


Figure 3.9.11 Operation of Register Buffer

Shows the block diagram of this mode.

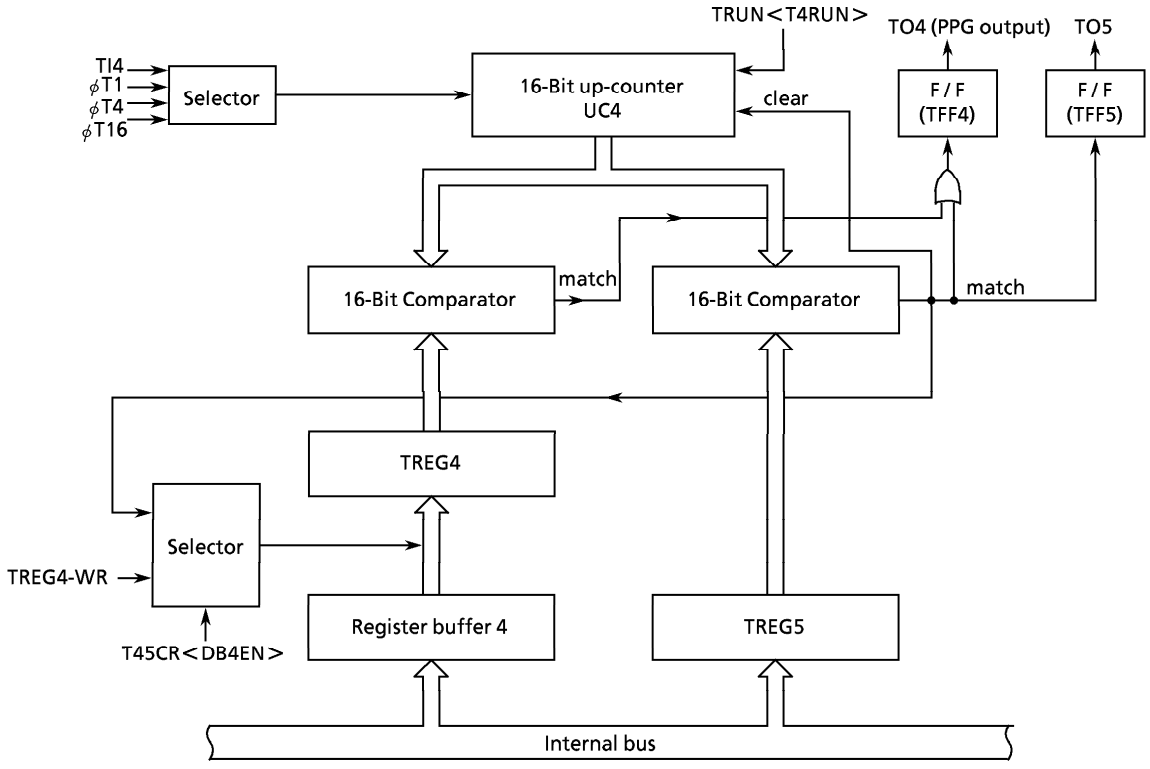


Figure 3.9.12 Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

The loading of up-counter (UC4) values into the capture registers CAP1 and CAP2, the timer flip-flop TFF4 inversion due to the match detection by comparators CP4 and CP5, and the output of the TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt function, they can be applied in many ways, for example:

① One-shot pulse output from external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot Pulse Output from External Trigger Pulse

Set the up-counter UC4 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up-counter into capture register CAP1 at the rising edge of the TI4 pin. Then set to T4MOD<CAP12M1, 0> = 01.
When the interrupt INT4 is generated at the rising edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 (c+d), and set the above set value (c+d) plus a one-shot pulse width (p) to TREG5 (c+d+p). When the interrupt INT4 occurs the T4FFCR<EQ5T4, EQ4T4>register should be set that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or TREG5. When interrupt INTTR5 occurs, this inversion will be disabled.

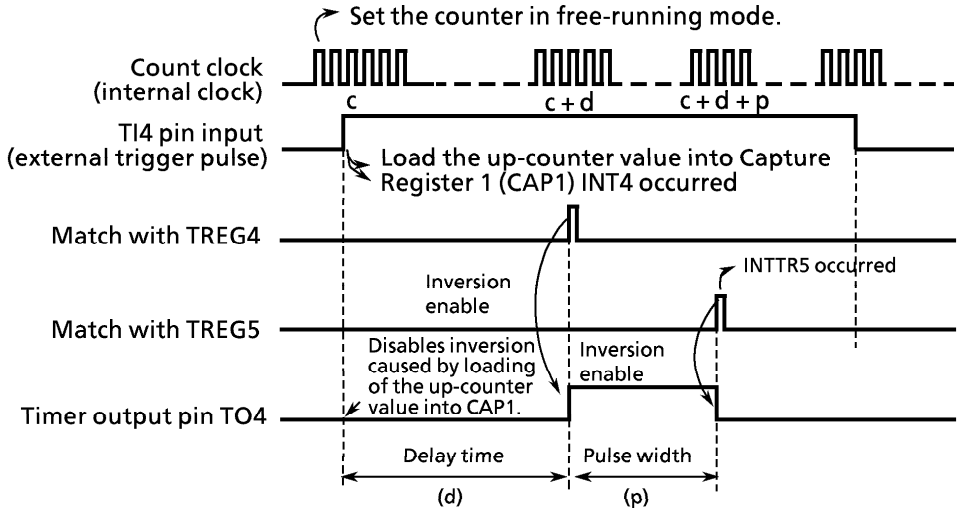


Figure 3.9.13 One-Shot Pulse Output (with Delay)

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

※ Cock Condition
- system clock : 1 (fc)
- prescaler clock : f_{FPH}

Main setting

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

- T4MOD: Keep counting (Free-runnig); Count with ϕT1.
- T4FFCR: Load the up-counter value into CAP1 at the rise edge of TI4 pin input; Clear TFF4 to zero; Disable TFF4 inversion.
- P8CR / P8FC: Select P82 as the TO4 pin.
- INTE45: Enable INT4, and disable INTTR4 and INTTR5.
- TRUN: Start timer 4.

Setting of INT4

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

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

Setting of INT5

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

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

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

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

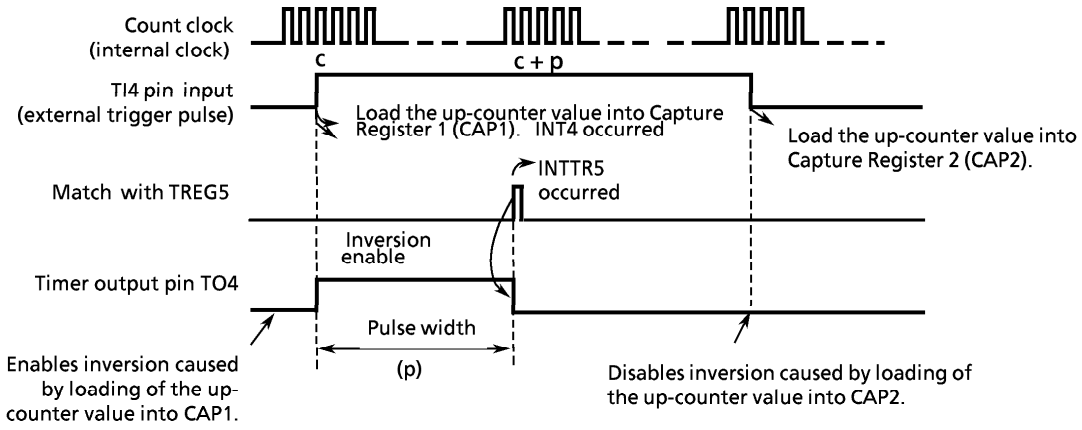


Figure 3.9.14 One-Shot Pulse Output (without Delay)

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. The value of the up-counter is loaded into the capture register CAP1 at the rising edge of the timer flip-flop TFF1 of 8-bit timers (Timer 0 and Timer 1), and into CAP2 at its falling edge.
The frequency is calculated by the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.

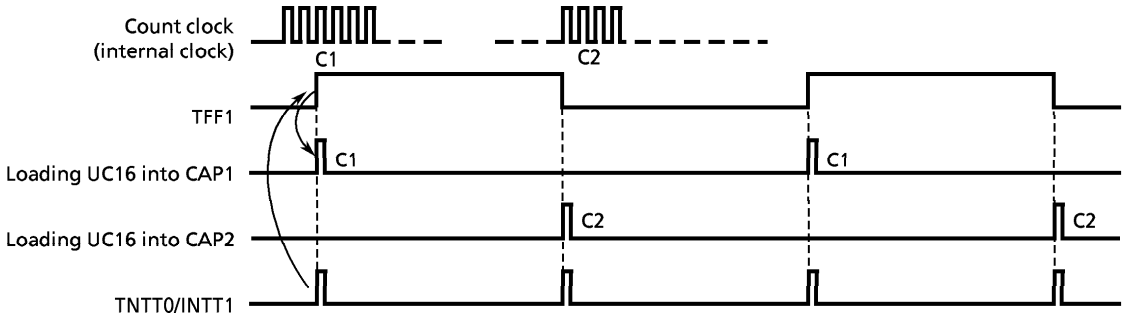


Figure 3.9.15 Frequency Measurement

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 $100 \div 0.5$ [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 $100 \times 0.8\ \mu s = 80\ \mu s$.

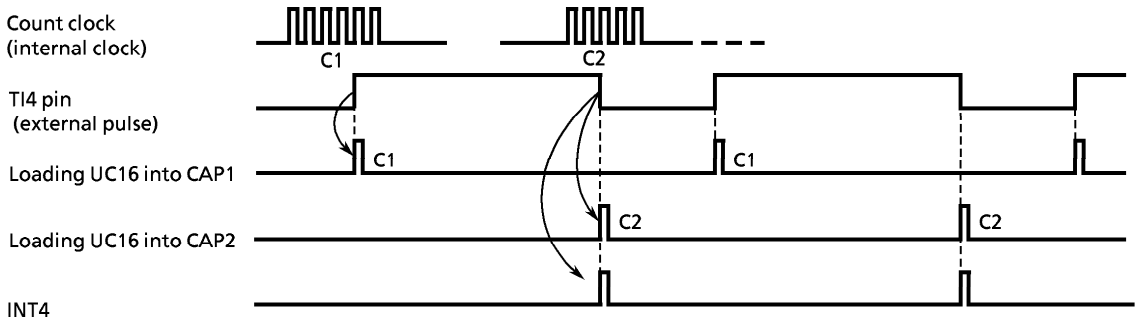


Figure 3.9.16 Pulse Width Measurement

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.

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

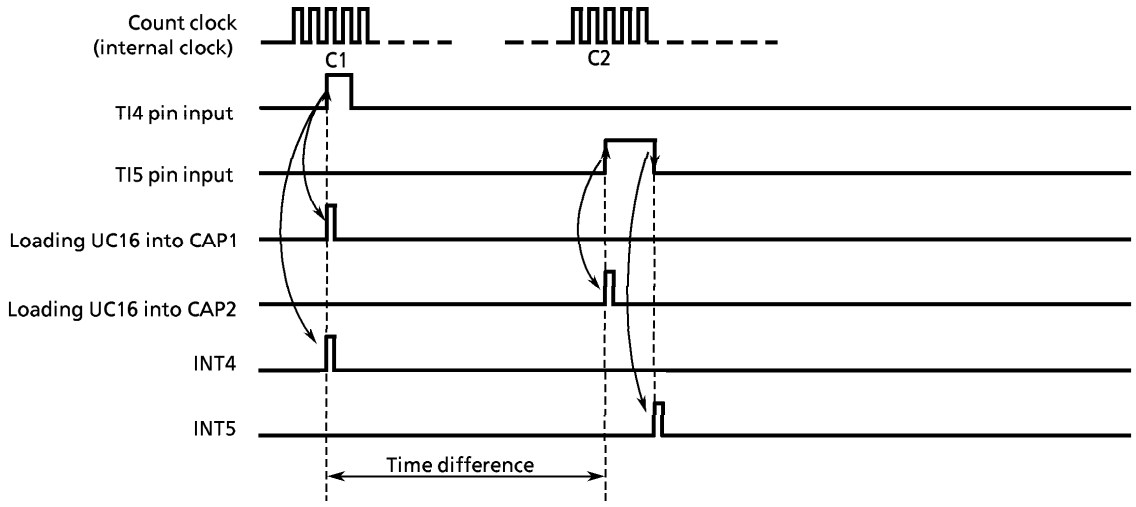


Figure 3.9.17 Time Difference Measurement

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

In this mode, signals with any different phase can be outputted by free-running up-counter UC4. When the value in up-counter UC4 and the value in TREG4 (TREG5) match, the value in TFF4 (TFF5) is inverted and output to TO4 (TO5).

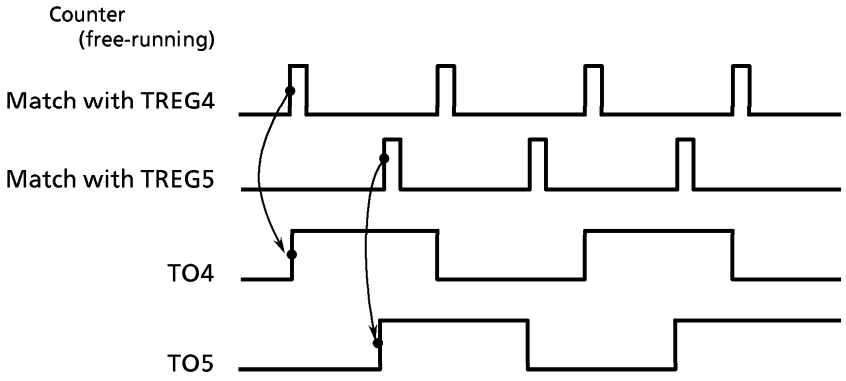


Figure 3.9.18 Phase Output

Cycles (counter overflow time) of the above output waves are listed on table 3.9.2.

Table 3.9.2 Timer Output Cycle on the Different Phased Pulse Output Mode

at fc = 20 MHz

select prescaler clock <PRCK1, 0>	Gear value <GEAR2 to 0>	Counter Overflow Time		
		ϕT1	ϕT4	ϕT16
00 (f_{FPH})	000 (fc)	26.214 ms	104.858 ms	419.430 ms
	001 (fc/2)	52.429 ms	209.715 ms	838.861 ms
	010 (fc/4)	104.858 ms	419.430 ms	1.678 s
	011 (fc/8)	209.715 ms	838.861 ms	3.355 s
	100 (fc/16)	419.430 ms	1.678 s	6.711 s
10 (fc/16 clock)	XXX	419.430 ms	1.678 s	6.711 s

XXX : Don't care

3.10 Serial Channel

TMP93CS42A contains 2 serial I/O channels. Channel 0 is used only in UART mode. Channel 1 selects UART mode (asynchronous transmission) or I/O interface mode (synchronous transmission).
The serial channel has the following operation modes.

- I/O interface mode (channel 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

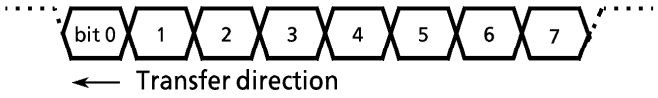
In mode 1 and mode 2, a parity bit can be added. Mode 3 has a wake-up function for making the master controller start slave controllers in a serial link (multi-controller) system.
Figure 3.10.1 shows the data format in each mode.
Serial Channels 0 and 1 can be used independently.
Channel 1 can use mode 0 to 3, and thus only the operation of channel 1 will be explained below.

Different Points between Channel 0 and 1

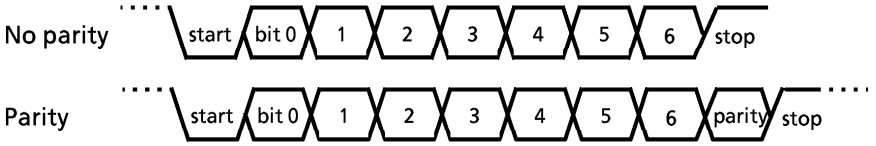
	Channel 0	Channel 1
Pin Name	$\overline{\text{TXD0}}$ (P90), RXD0 (P55) $\overline{\text{CTS0}}$ (P56)	TXD1 (P93), RXD1 (P57) SCLK1 (P95)
Handshake Function	Yes	No (no $\overline{\text{CTS1}}$ pin)
I/O interface mode	No	Yes
UART mode	Yes	Yes

Note : Using the handshake function can transmit in units of one data format. Thus over run error is prevented. See "Handshake function" for details.

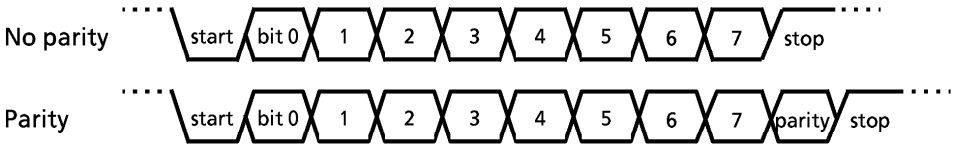
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

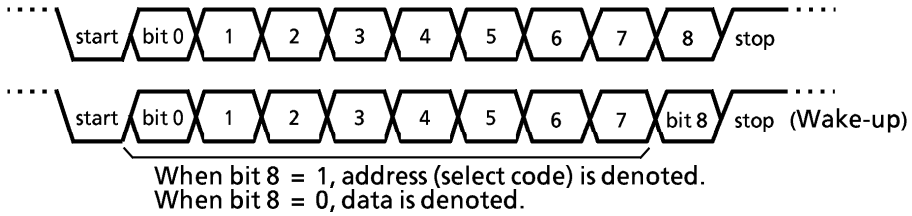


Figure 3.10.1 Data Formats

The serial channel has buffer registers for transmitting and receiving operations in order to temporarily store transmitted or received data. This is done so that transmitting and receiving operations can be done independently (full duplex).
However, in I/O interface mode, the SCLK (serial clock) pin is used for both transmitting and receiving, the channel becomes half-duplex.
The receiving data register is a double buffer structure to prevent the occurrence of an overrun error and it provides one data format of margin before CPU reads the received data. The receiving data register stores the previously received data while the buffer register receives the next frame data.
By using $\overline{CTS}$ and $\overline{RTS}$ (there is no $\overline{RTS}$ pin, so any single port must be controlled by software) at channel 0, 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 to not start the receiving operation by erroneous 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 of the start bit.
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(transmit interrupt) or INTRX(receive interrupt) interrupt occurs. If an overrun error, parity error, or framing error occurs during receiving operation, flag SC1CR<OERR, PERR, FERR> will be set.
The serial channel 0 / 1 includes a special baud rate generator, which can set to any baud rate by dividing the frequency of 4 clocks (ϕT0, ϕT2, ϕT8, and ϕT32 from the 9-bit prescaler shared by the 8-bit / 16-bit timers) by the value 2 to 16.
In I/O interface mode, it is possible to input synchronous signals as well as to transmit or receive data by using an external clock.

3.10.1 Control Registers

The serial channels are controlled by 3 control registers SC1CR, SC1MOD and BR1CR. Transmitted and received data are stored in register SC1BUF.

Note : The number of the control register name is equaled to the channel number.

SC0MOD (0052H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data Bit 8	Hand shake 0: CTS disable 1: CTS enable	Receiving Function 0: Receive disable 1: Receive enable	Wake up Function 0: disable 1: Enable	Serial transmission mode 00: (rserved) 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 Trigger 01: baud rate generator 10: Internal clock ϕ1 11: Don't care	

Serial transmission clock source (UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	Don't care

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

Serial transmission mode

00	(rserved)	
01	UART	7-bit length
10		8-bit length
11		9-bit length

Wake-up Function

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

Receiving Function

0	Receive Disable
1	Receive Enable

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

0	Disable (always Transferable)
1	Enable

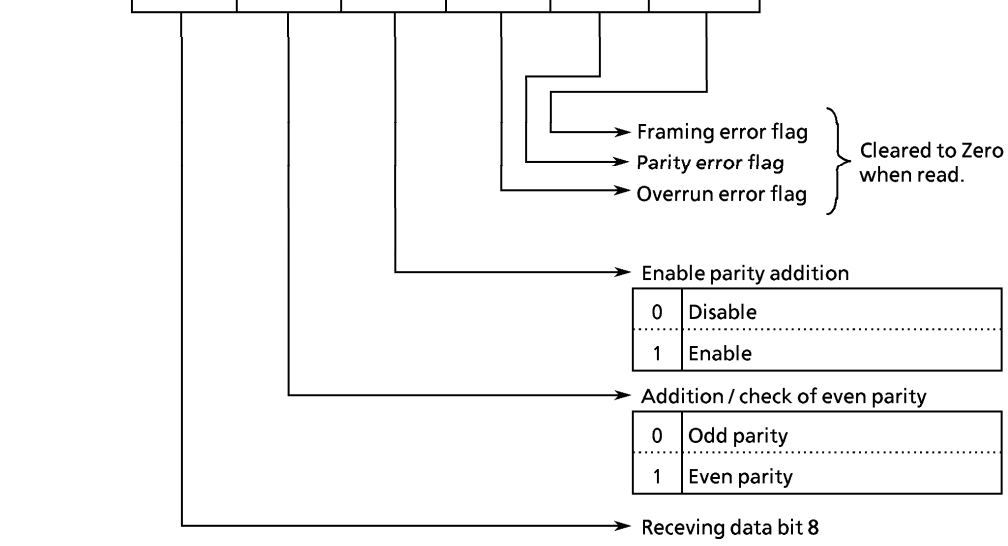
Transmission data bit 8

Note : There is SC1MOD (56H) in Channel1

Figure 3.10.2 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		
Read/Write	R	R/W		R (cleared to Zero when read)				
After reset	Undefined	0	0	0	0	0		
Function	Received data Bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error (Overrun)	1: error (Parity)	1: error (Framing)	Fixed to "0".	Fixed to "0".



Note 1: Serial control register for channel 1 is SC1CR (55H).
Note 2: As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.10.3 Serial Control Register (channel 0, SC0CR)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
Bit symbol			BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Note: Always fixed to "0".		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

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

Selecting the input clock of baud rate generator

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

Note 1: Serial contorl register for channel 1 is BR1CR(57H).
Note 2: Set TRUN<PRRUN> to "1" when the baud rate generator is used.
Note 3: BR0CR<bit6> is always read as "1".
Note 4: Don't read from or write to BR0CR register during sending or receiving.

Figure 3.10.4 Serial Channel Control(channel 0, BR0CR)

SC0BUF (0050H)

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

Note: Read-modify-write is prohibited for SC0BUF.

Figure 3.10.5 Serial Transmission/Receiving Buffer Registers (channel 0, SC0BUF)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8		RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfered data Bit 8	Note: Always fixed to "0".	Receiving Function 0: Receive disable 1: Receive Enable	Wake up Function 0: disable 1: enable	Serial Transmission mode 00: I/O interface mode 01: 7-Bit UART 10: 8-Bit UART 11: 9-Bit UART		Serial Transmission clock (UART) 00: TO0 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	

Serial transmission clock source (For UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	Don't care

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

Serial transmission mode

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

Wake up Function

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

Receiving control

0	Receive disable
1	Receive enable

Transmission data bit 8

Figure 3.10.6 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	Undfened	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

Select I/O interface input clock

0	Baud rate generate
1	SCLK1 Pin input

Edge selection for SCLK pin (input mode only)

0	Transmits and receives (↑) data at rise edge of SCLK1
1	Transmits and receives (↓) data at fall edge of SCLK1

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

Enable parity addition

0	Disable
1	Enable

Addition / check of even parity

0	Odd Parity
1	Even Parity

Receiving data bit 8

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

Figure 3.10.7 Serial Control Register (Channel 1, SC1CR)

BR1CR (0057H)

	7	6	5	4	3	2	1	0
Bit symbol			BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Note: Always fixed to "0".		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

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

Selecting the input clock of baud rate generator

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

Note 1 : To use baud rate generator, set TRUN<PRRUN> to "1", putting the prescaler in RUN.
Note 2 : BR1CR<bit6> is always read as "1".
Note 3 : Don't read from or write to BR1CR register during sending or receiving.

Figure 3.10.8 Baud Rate Generator Control Register (channel 1, BR1CR)

SC1BUF (0054H)

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

Note : Read-modify-write is prohibited for SC1BUF.

Figure 3.10.9 Serial Transmission/Receiving Buffer Registers (channel 1, SC1BUF)

P9FC (001DH)

	7	6	5	4	3	2	1	0
Bit symbol			P95F		P93F			P90F
Read/Write			W		W	W		W
After reset			0		0	0		0
Function			0: PORT 1: SCLK1		0: PORT 1: TxD1	Fix to "0".		0: PORT 1: TxD0

Prohibit Read modify write

Setting TXD0 output of P90

0	Port
1	TxD0 output (channel 0)

Setting TXD1 output of P93

0	Port
1	TxD1 output (channel 1)

Setting SCLK1 output of P95

0	Port
1	SCLK1 output (channel 1)

Figure 3.10.10 Port 9 Function Register (P9FC)

ODE (0058H)

	7	6	5	4	3	2	1	0
Bit symbol							ODE1	ODE0
Read/Write							R/W	
After reset							0	0
Function							P93 0: CMOS 1: OPEN Drain	P90 0: CMOS 1: OPEN Drain

Setting P90 as Open-drain output

0	CMOS output
1	Open-drain output

Setting P93 as Open-drain

0	CMOS output
1	Open-drain output

Note : ODE<bit7 to 2> is always read as "1".

Figure 3.10.11 Port 9 Open Drain Enable Register (ODE)

3.10.2 Configuration

Figure 3.10.12 shows the block diagram of the serial channel 0.

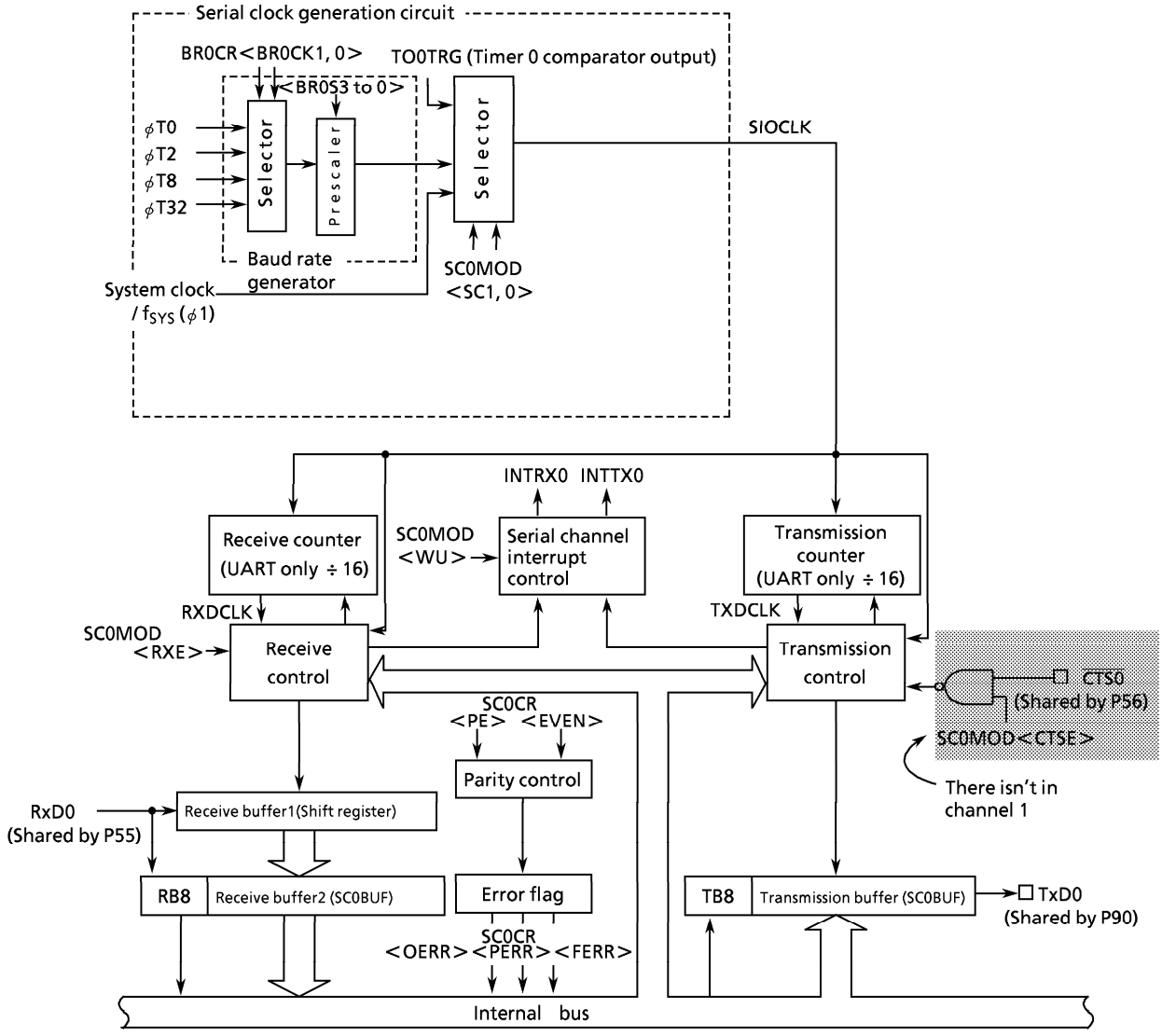
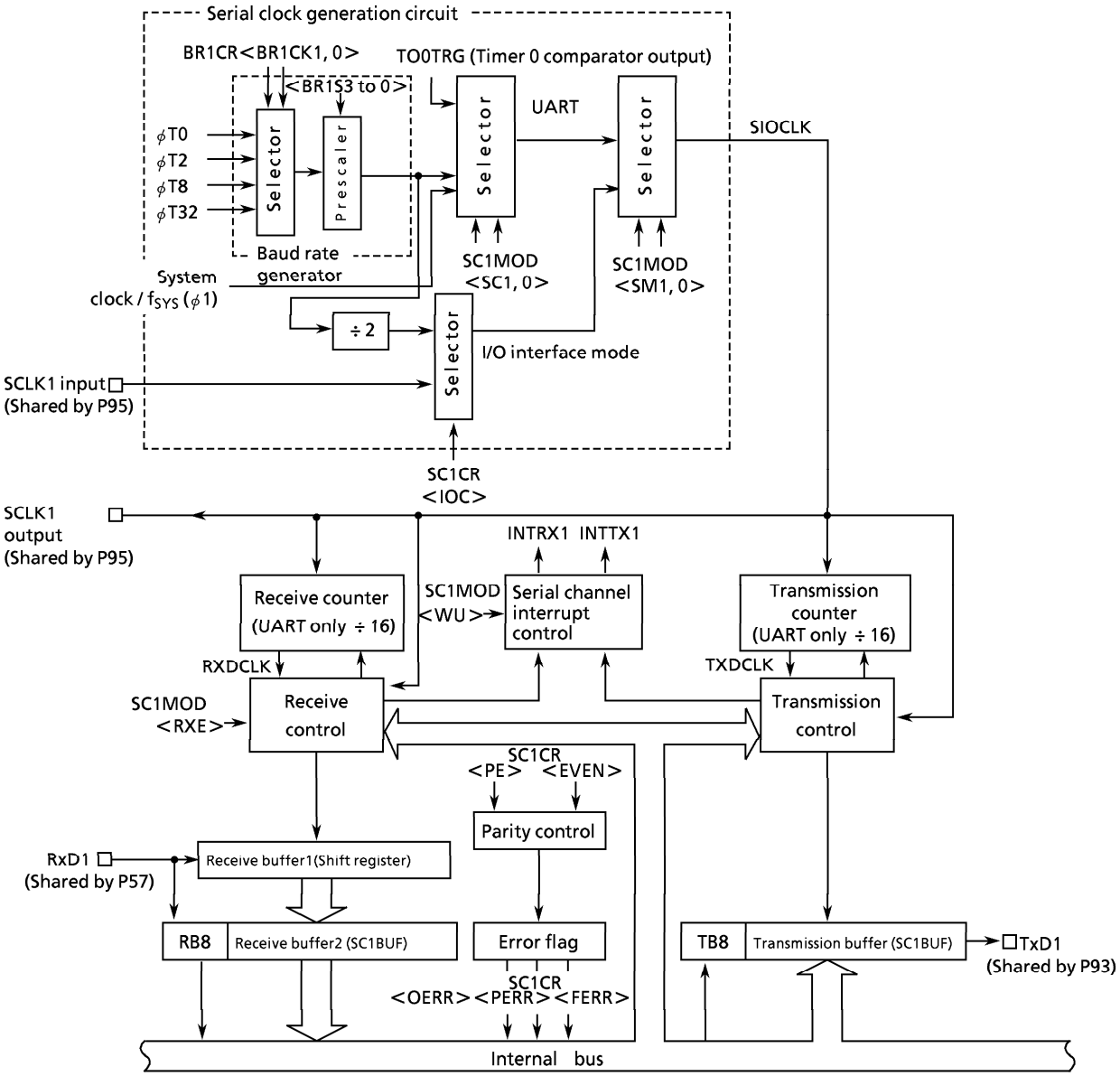


Figure 3.10.12 Block Diagram of the Serial Channel 0

Figure 3.10.13 shows the block diagram of the serial channel 1.



Note : SCLK1 input/output is used only in I/O interface mode.

Figure 3.10.13 Block Diagram of the Serial Channel 1

① Prescaler, Prescaler clock select

There are 9 bit prescaler and prescaler clock selection registers to generate input clock for 8 bit Timer0, 1, 16 bit Timer4, 5 and Serial Interface0, 1.
Figure 3.10.14 shows the block diagram. Table 3.10.1 shows prescaler clock resolution into the baud rate generator.

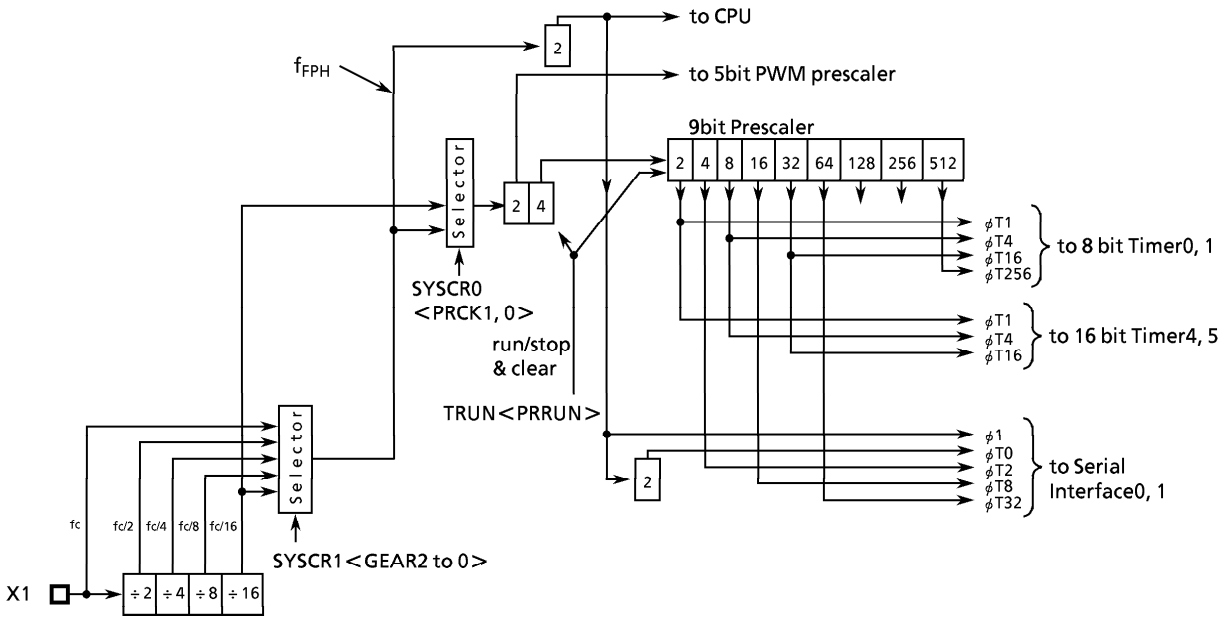


Figure 3.10.14 The Block Diagram of Prescaler

Table 3.10.1 Prescaler Clock Resolution to Baud Rate Generator

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

XXX : Don't care — : Can not use

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

Resetting sets <PRCK1, 0> to "00" 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". The prescaler is cleared to zero and stops operation when<PRRUN> is set to "0". Resetting clears <PRRUN> to "0", and the prescaler is cleared and stops operation.
When the IDLE1 mode (only the oscillator operates) is used, set TRUN <PRRUN> to '0' to stop this prescaler before the "HALT" instruction is executed.

② Baud Rate Generator

The baud rate generator is a circuit that generates transmission and receiving clocks to determine the transfer rate of the serial channel.
The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8, or ϕT32, is generated by the 9-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate generator control register BR1CR<BR1CK1, 0>.
The baud rate generator includes a 4-bit frequency divider, which divides frequency by 2 to 16 values to determine the transfer rate.
How to calculate a transfer rate when the baud rate generator is used is explained below.

- UART mode

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

- I/O interface mode

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

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

※ Clock Condition { clock gear : 1 (fc); prescaler clock : f_{FPH}

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

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

Table 3.10.2 shows an example of the transfer rate in UART mode.

Also with 8-bit timer 0, the serial channel can get a transfer rate. Table 3.10.3 shows an example of baud rate using timer 0.

Table 3.10.2 Selection of Transfer Rate (1) (When Baud Rate Generator Is Used)

Unit (Kbps)

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

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

Note2 : This table is calculated when fc is selected as a system clock. The clock gear is set for fc and the system clock as the prescaler clock input.

Table 3.10.3 Selection of Transfer Rate (2) (When timer 0 (input Clock ϕT1) is used)

Unit (Kbps)

TREG0 \ fc	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

How to calculate the transfer rate (when timer 0 is used):

$$\text{Transfer rate} = \frac{\text{SYSCR0<PRCK1 to 0>}}{\text{TREG0} \times \underset{\uparrow}{8} \times 16}$$

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

Note1 : Timer 0 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. The clock gear is set for fc and the f_{FPH} as the prescaler clock input.

③ Serial Clock Generation Circuit

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

- I/O interface mode
  When in SCLK output mode with the setting of SC1CR<IOC> = "0", the basic clock will be generated by dividing the output of the baud rate generator by 2 as described before. When in SCLK input mode with the setting of SC1CR<IOC> = "1", the rising edge or falling edge will be detected according to the setting of SC1CR<SCLKS> register to generate the basic clock.
- UART mode
  The setting of SC1MOD <SC1, 0> will select between the baud rate generator clock, internal clock $\phi 1$ (max 625Kbps at fc = 20 MHz), or the match detect signal from timer 0 to generate the basic clock SIOCLK.

④ Receiving Counter

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

⑤ Receiving Control

- I/O interface mode
  When in SCLK output mode with the setting of SC1CR<IOC> = "0", the RxD1 signal will be sampled at the rising edge of the shift clock which is output to the SCLK1 pin.
  When in SCLK input mode with the setting SC1CR<IOC> = "1", the RxD1 signal will be sampled at the rising edge or falling edge of the SCLK input according to the setting of the SC1CR<SCLKS> register.
- UART mode
  The receiving control block has a circuit for detecting the start bit by the rule of majority. When two or more "0" are detected during the 3 samples, it is recognized as start bit and the receiving operation is started.
  The data being received is also evaluated by the majority rule.

⑥ Receiving Buffer

To prevent an overrun error, the receiving buffer has a double buffer structure.
Received data is stored bit by bit in receiving buffer 1 (shift register type). When 7 bits or 8 bits of data is stored in the receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC1BUF) generating an interrupt INTRX1. The CPU reads only receiving buffer 2 (SC1BUF). Even before the CPU reads receiving buffer 2 (SC1BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC1BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC1CR<RB8> are still preserved.
The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SC1CR<RB8>.
When in 9-bit UART mode, the wake-up function of the slave controller is enabled by setting SC1MOD<WU> to "1", and interrupt INTRX1 occurs only when SC1CR<RB8> is set to "1".

⑦ Transmission Counter

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

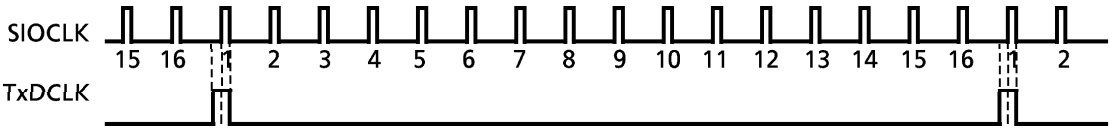


Figure 3.10.15 Generation of Transmission Clock

⑧ Transmission Controller

- I/O interface mode
  In SCLK output mode with the setting of SC1CR<IOC> = "0", the data in the transmission buffer is output bit by bit to the TxD1 pin at the rising edge of the shift clock which is output from the SCLK1 pin.
  In SCLK input mode with the setting of SC1CR<IOC> = "1", the data in the transmission buffer is output bit by bit to the TxD1 pin at the rising edge or falling edge of the SCLK input according to the setting of the SC1CR<SCLKS> register.

- UART mode
  When transmission data is written to the transmission buffer from the CPU, transmission starts at the rising edge of the next TxDCLK, generating a transmission shift clock TxDSFT.

Handshake function

Serial channel 0 has a $\overline{CTS0}$ pin. Using this pin, data can be sent in units of one frame ; thus, overrun errors can be avoided. The handshake function is enabled/ disabled by SC1MOD<CTSE>.

When the $\overline{CTS0}$ pin goes high, after completion of the current data send, data send is halted until the $\overline{CTS0}$ pin goes low again. When the INTTX1 Interrupt is generated, it requests the next data send to the CPU.

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

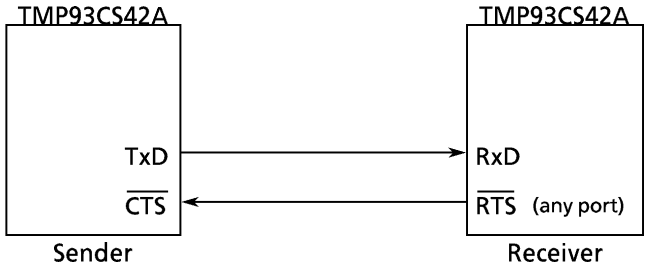
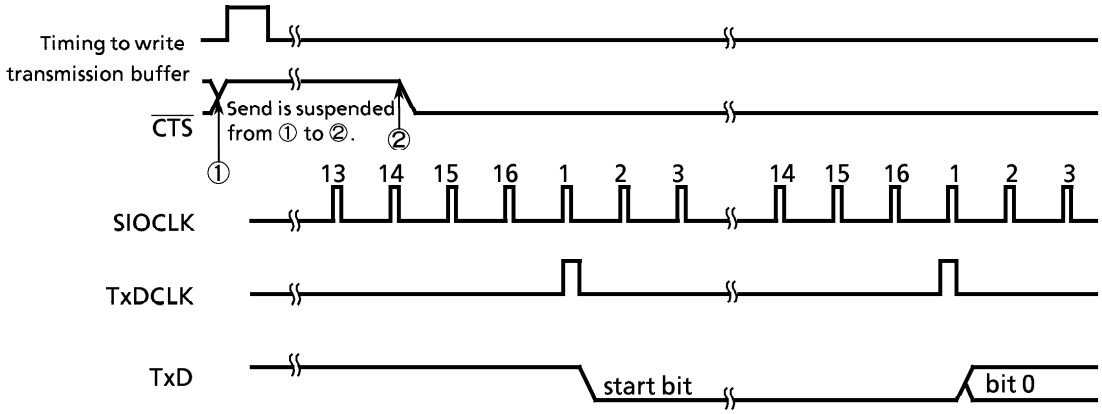


Figure 3.10.16 Handshake Function



Note 1 : If the $\overline{CTS}$ signal rises during transmission, the next data is not sent after the completion of the current transmission.

Note 2 : Transmission starts at the first TxDCLK clock falling edge after the $\overline{CTS}$ signal falls.

Figure 3.10.17 Timing of $\overline{CTS}$ (Clear to send)

⑨ Transmission Buffer

The transmission buffer (SC1BUF) shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX1 interrupt.

⑩ Parity Control Circuit

When the serial channel control register SC1CR<PE>is set to "1", it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART modes. With SC1CR <EVEN> register, even or odd parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer SC1BUF. The data is transmitted after the parity bit is stored in SC1BUF<TB7> when in 7-bit UART mode or in SC1MOD <TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before the transmission data is written to the transmission buffer.
For receiving, data are shifted in the receiving buffer 1, the parity is added after the data is transferred to receiving buffer 2 (SC1BUF), and then compared with SC1BUF<RB7> when in 7-bit UART mode and with SC1MOD<RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs and SC1CR<PERR> flag is set.

⑪ Error Flag

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

1. Overrun error <OERR>

   If all bits of the next data are received in receiving buffer 1 while valid data is stored in receiving buffer 2 (SC1BUF), an overrun error will occur.

2. Parity error <PERR>

   The parity generated for the data shifted in receiving buffer 2 (SC1BUF) 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 Generating Timing

1) 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 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 timing	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	←	←

2) In I/O Interface mode

Timing for send interrupt generation	SCLK0 output mode	Immediately after rise of last SCLK0 signal (See Figure 3.10.20)
	SCLK0 input mode	Immediately after rise (rising mode) or fall (falling mode) of last SCLK0 signal (See Fiugre 3.10.21)
Timing for receive interrupt generation	SCLK0 output mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.10.22)
	SCLK0 input mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.10.23)

3.10.3 Operational Description

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

This mode is able to use in channel 1.
This mode is used to increase the number of I/O pins for transmitting or receiving data to or from an external shifter register.
This mode includes the SCLK output mode to output synchronous clock SCLK and SCLK input mode to input external synchronous clock SCLK.

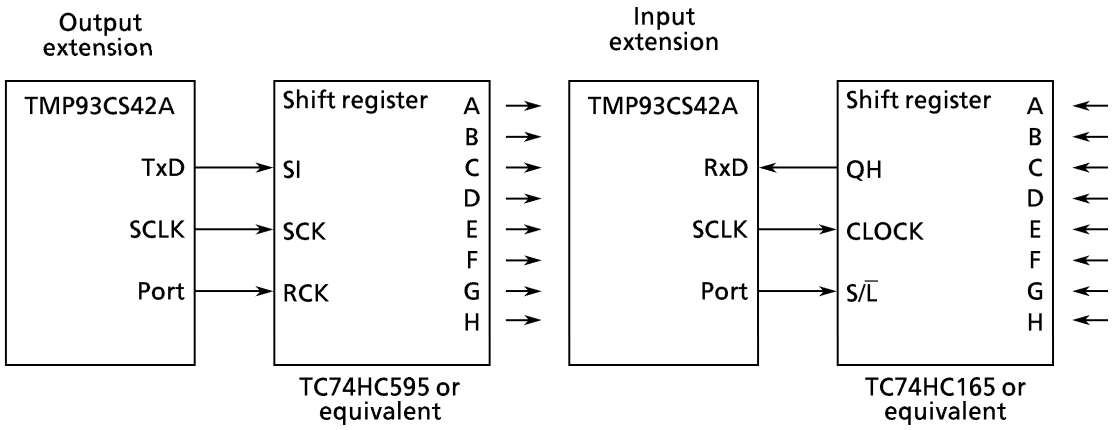


Figure 3.10.18 Example of SCLK Output Mode Connection

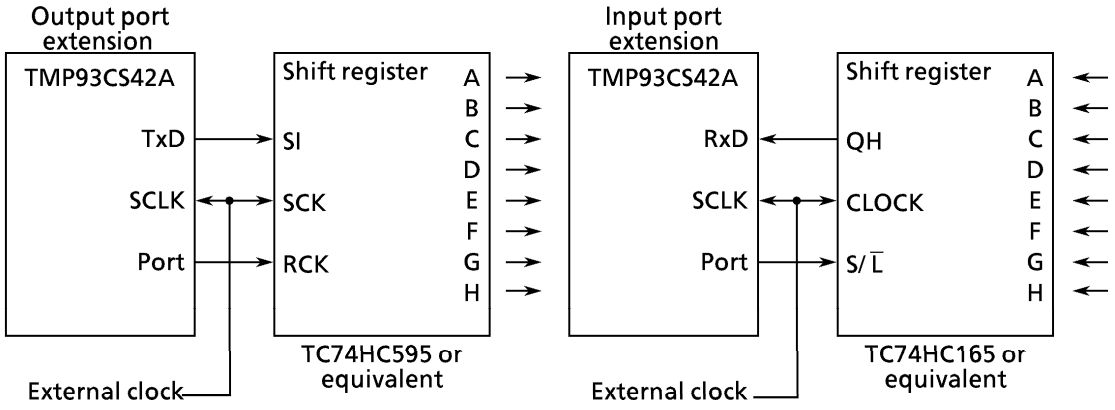


Figure 3.10.19 Example of SCLK Input Mode Connection

① Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TxD1 pin and SCLK1 pin respectively, each time the CPU writes data to the transmission buffer. When all data is output, INTES1<ITX1C> will be set to generate the INTTX1 interrupt.

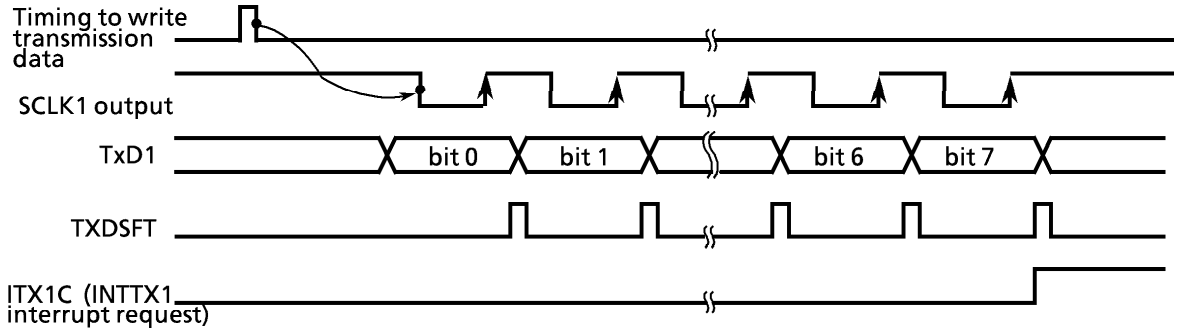


Figure 3.10.20 Transmitting Operation in I/O interface Mode (SCLK Output Mode)

In SCLK input mode, 8-bit data is output from TxD1 pin when SCLK1 input becomes active after data is written to the transmission buffer by CPU.
When all data is output, INTES1<ITX1C> will be set to generate INTTX1 interrupt.

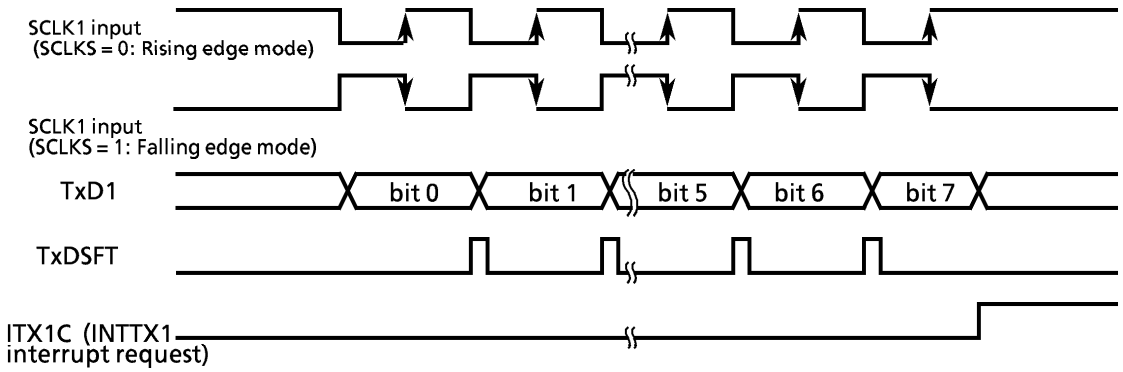


Figure 3.10.21 Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK1 pin and the data is shifted to receiving buffer 1. This starts when the receive interrupt flag INTES1<IRX1C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC1BUF according to the timing shown below) and INTES1<IRX1C> will be set again to generate INTRX1 interrupt.

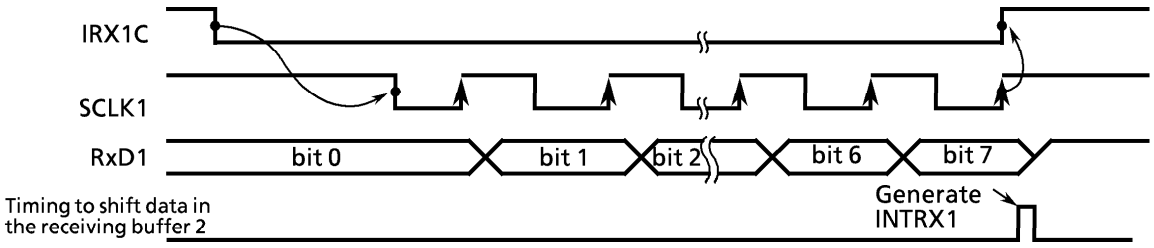


Figure 3.10.22 Receiving Operation in I/O interface Mode (SCLK Output Mode)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES1 <IRX1C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC1BUF according to the timing shown below) and INTES1 <IRX1C> will be set again to generate INTRX1 interrupt.

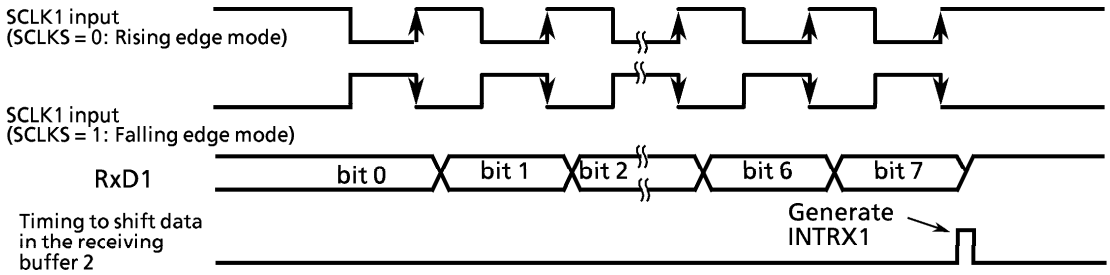


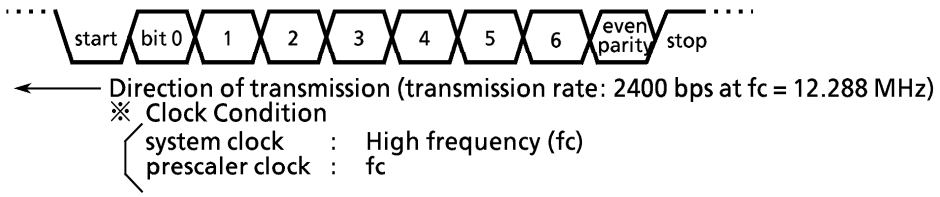
Figure 3.10.23 Receiving Operation in I/O interface Mode (SCLK Input Mode)

Note : For data receiving, the system must be placed in the receive enable state (SC1MOD<RXE> = "1")

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

The 7-bit mode can be set by setting serial channel mode register SC1MOD <SM1,0> to "01".
In this mode, a parity bit can be added, and the addition of the parity bit can be enabled or disabled by serial channel control register SC1CR<PE>, and even parity or odd parity is selected by SC1CR <EVEN> when <PE> is set to "1" (enable).

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



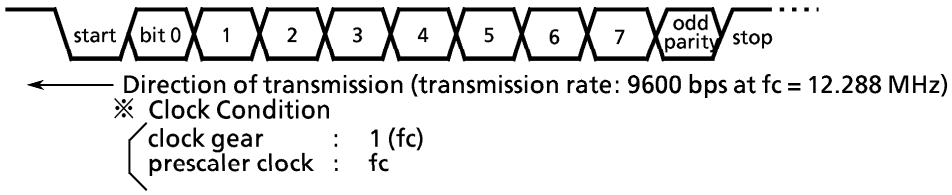

```
          7 6 5 4 3 2 1 0
P9CR    ← X X - - - - - 1   }
P9FC    ← X X - X - 0 X 1   }  Select P90 as the TxD0 pin.
SC0MOD  ← X 0 - X 0 1 0 1      Set 7-bit UART mode.
SC0CR   ← X 1 1 X X X 0 0      Add even parity.
BR0CR   ← 0 X 1 0 0 1 0 1      Set transfer rate at 2400 bps.
TRUN    ← 1 X - - - - - -      Start the prescaler for the baud rate generator.
INTES0  ← 1 1 0 0 - - - -      Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF  ← * * * * * * * *      Set data for transmission.
```

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

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

The 8-bit UART mode can be specified by setting SC1MOD<SM1,0> to "10". In this mode, the parity bit can be added (the addition of a parity bit is enabled or disabled by SC1CR<PE>), and even parity or odd parity is selected by SC1CR<EVEN> when <PE> is set to "1" (enable).

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



Main setting

```
                7 6 5 4 3 2 1 0
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 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.
```

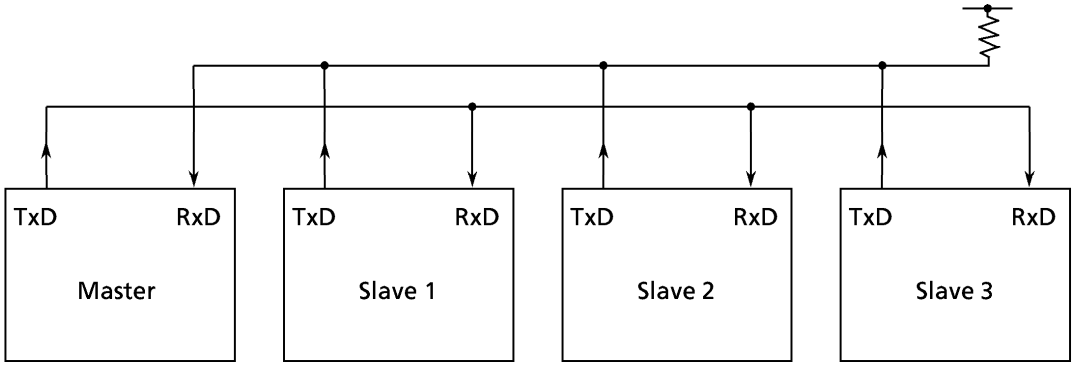
Note : X ; Don't care - ; No change

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

9-bit UART mode can be specified by setting SC1MOD<SM1,0> to "11". In this mode, parity bit cannot be added.
For transmission, the MSB (9th bit) is written in SC1MOD <TB8>. For receiving it is stored in SC1CR<RB8>. For writing and reading of the buffer, the MSB is read or written first then the rest of the data from SC1BUF.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SC1MOD<WU> to "1". The interrupt INTRX1 occurs only when <RB8> = 1.



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

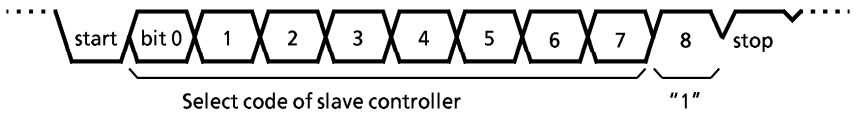
Figure 3.10.24 Serial Link Using Wake-Up Function

Protocol

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

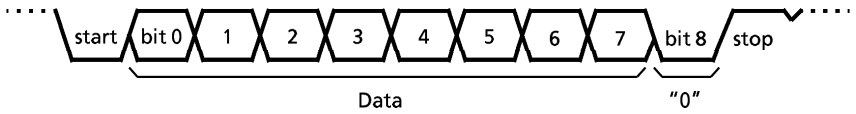
② Set SC1MOD<WU> bit of each slave controller to "1" to enable data receiving.

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".



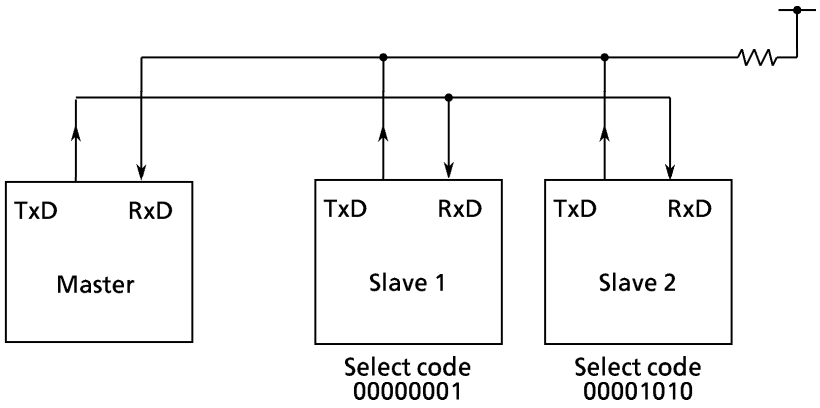
④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.

⑤ The master controller transmits data to the specified slave controller whose SC1MOD<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 INTRX1.
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

```
P9CR    ← X X - - - - X 1   }
P9FC    ← X X - X - 0 X 1   }  Select P90 as TxD0 pin.
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

Main

```
P9CR    ← X X - - - - X 1   }
P9FC    ← X X - X - 0 X 1   }  Select P90 as TxD (open drain output).
ODE     ← X X X X X X - 1   }
INTES0  ← 1 1 0 1 1 1 1 0      Enable INTRX0 and INTTX0.
SC0MOD  ← 0 0 1 1 1 1 1 0      Set <WU> to "1" in the 9-bit UART transmission mode
                               with transfer clock φ1.
```

INTRX0 interrupt

```
Acc ← SC0BUF
if Acc = Select code
Then SC0MOD← - - - 0 - - - -   Clear <WU> to "0".
```

3.11 Analog / Digital Converter

TMP93CS42A contains an analog / digital converter (AD converter) with 5-channel analog input that features 10-bit successive approximation.
Figure 3.11.1 shows the block diagram of the AD converter. 5-channel analog input pins (AN4 to AN0) are shared by input-only port P5 which also can be used as a general purpose input port.

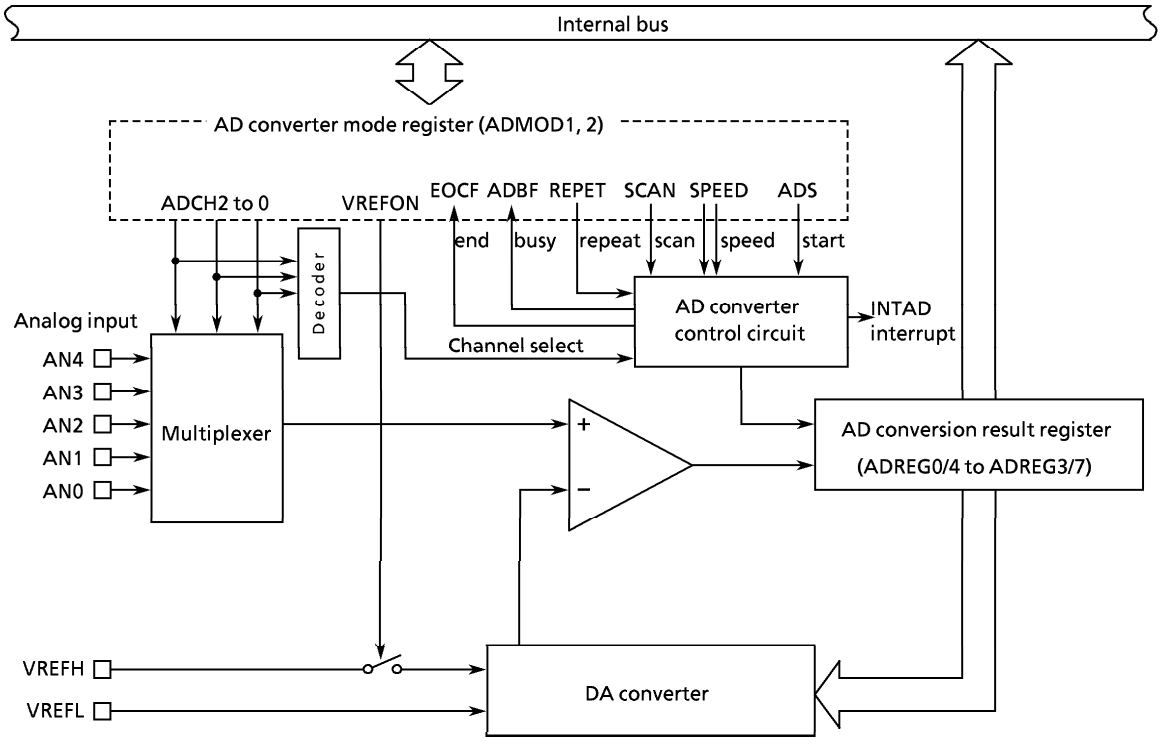


Figure 3.11.1 Block Diagram of AD Converter

Note 1 : This AD converter does not have a built-in sample and hold circuit.

Note 2 : When the power supply current is reduced in IDLE2, IDLE1, STOP mode, there is possible to set a standby enabling the internal comparator. Stop operation of AD converter before execution of "HALT" instruction. And set ADMOD2<SPEED1, 0> ="00"

Note 3 : The operation above is guaranteed with $f_{FPH} \geqq 4$ MHz.

ADMOD1 (005EH)

	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	REPET	SCAN		ADS		
Read/Write	R		R/W			R/W		
After reset	0	0	0	0		0		
Function	AD Conversion End Flag 1: END	AD Conversion BUSY Flag 1: BUSY	Repeat Mode 0: Single Mode 1: Repeat Mode	Scan Mode 0: Fixed channel mode 1: Channel Scan Mode		AD Conversion start 1: START Always read as "0"		

AD Conversion Start

0	—
1	Start Conversion

Note) Always "0" when data is read

AD Scan Mode

0	Fixed channel mode
1	Channel scan mode

AD Repeat Mode

0	Single mode
1	Repeat mode

AD BUSY flag

0	NOT BUSY
1	BUSY

AD END flag

0	NOT END
1	END

Figure 3.11.2 AD Control Register (1/2)

ADMOD2 (005FH)

	7	6	5	4	3	2	1	0
Bit symbol	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	1		0	0		0	0	0
Function	String Resistance ON / OFF		Conversion Spead 00 : 160 states 01 : 320 states 10 : 640 states 11 : 1280 sates			Analog Input Channel Selecti		

Analog Input Channel Selection

<ADCH2 to 0> \ <SCAN>	0	1
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011	AN3	AN0→AN1→AN2→AN3
100	AN4	(reserved)
101	–	(reserved)
110	–	(reserved)
111	–	(reserved)

Conversion Speed

00	160 states (16 μs at 20 MHz)
01	320 states (32 μs at 20 MHz)
10	640 states (64 μs at 20 MHz)
11	1280 states (128 μs at 20 MHz)

Ladder Resistance ON / OFF selection

0	Ladder Resistance OFF
1	Ladder Resistance ON

Note : Set the <VREFON> register to "1" before starting (write <ADS> to "1") conversion.

Figure 3.11.2 AD Control Register (2/2)

ADREG04L (0060H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR01	ADR00						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN0 or AN4 are stored.							

ADREG04H (0061H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN0 or AN4 are stored.							

ADREG15L (0062H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR11	ADR10						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN1 are stored.							

ADREG15H (0063H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN1 are stored.							

Note : The result registers (ADREG04L/H) are used both as AN0 and AN4.
They are stored into ADREG04.

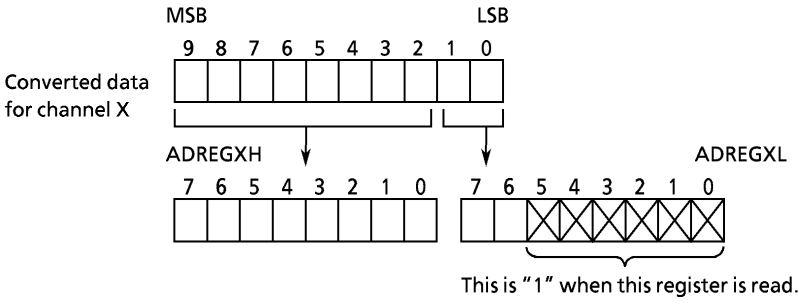


Figure 3.11.3 AD Conversion Result Register (ADREG04, 15) (1/2)

ADREG26L (0064H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN2 are stored.							

ADREG26H (0065H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN2 are stored.							

ADREG37L (0066H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN3 are stored.							

ADREG37H (0067H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN3 are stored.							

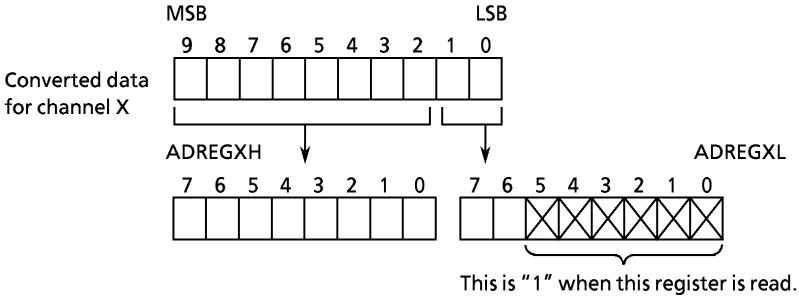


Figure 3.11.3 AD Conversion Result Register (ADREG26, 37) (2/2)

3.11.1 Operation

(1) Analog Reference Voltage

The high analog reference voltage is applied to the VREFH pin, and the low analog reference voltage is applied to VREFL pin.
The reference voltage between VREFH and VREFL is divided by 1024 (using string resistance) and compared with the analog input voltage for AD conversion.
The switch between VREFH and VREFL can be cut (OFF) by writing "0" to <VREFON>.
When the conversion can be started when <VREFON> = "0", a "1" must be written to <VREFON> and wait for 3 μs that the internal reference voltage is stable (regardless to fc) before writing "1" to <ADS>.

(2) Analog Input Channels

The analog input channel is selected by ADMOD2<ADCH2 to 0>. However, which channel to select depends on the operation mode of the AD converter.
In fixed analog input mode, one channel is selected by <ADCH2 to 0> among five pins: AN0 to AN4.
In analog input channel scan mode, the number of channels to be scanned is specified by <ADCH2 to 0>, such as AN0, AN0→AN1, AN0→AN1→AN2, AN0→AN1→AN2→AN3.
When reset, the AD conversion channel register will be initialized to ADMOD2 <ADCH2 to 0> = 000, so that the AN0 pin will be selected.
The pins which are not used as analog input channels can be used as ordinary input port pins on port P5.

(3) Starting AD Conversion

AD conversion starts when AD conversion register ADMOD1<ADS> is written "1". When conversion starts, conversion busy flag ADMOD1<ADBF> which indicates "conversion is in progress" will be set to "1".

(4) AD Conversion Mode

Both fixed AD conversion channel mode and conversion channel scan mode have two conversion modes, single and repeat conversion modes.
In fixed channel repeat mode, conversion of the specified single channel is executed repeatedly.
In scan repeat mode, scanning from AN0, ···→AN3 is executed repeatedly.
The AD conversion mode is selected by ADMOD1<REPET, SCAN>.

(5) AD Conversion Speed Selection

There are four AD conversion speed modes. The selection is determined by ADMOD2<SPEED1, 0> register.
When reset, <SPEED1, 0> will be initialized to "00", so that the 160 state conversion mode will be selected. (16 μs at 20 MHz)

(6) AD Conversion End and Interrupt

- AD conversion single mode

  ADMOD1<EOCF> for AD conversion end will be set to "1", ADMOD1 <ADBF> busy flag will be reset to "0", and INTAD interrupt will be enabled when AD conversion of specified channel ends in fixed conversion channel mode or when AD conversion of the last channel ends in channel scan mode.

- AD conversion repeat mode

  For both fixed conversion channel mode and conversion channel scan mode, INTAD should be disabled when in repeat mode. Always set the INTE0AD at "000" to disable the interrupt request. Write "0" to ADMOD1<REPET> to end the repeat mode. Then the repeat mode will be exited as soon as the conversion in progress completes.

(7) Storing the AD Conversion Result

The results of AD conversion are stored in ADREG04 to ADREG37 registers for each channel. The result register ADREG04 is used both as AN0 and AN4. However, the current conversion data can not determine which channels. In repeat mode, the registers are updated whenever conversion ends. ADREG04 to ADREG37 are read-only registers.

(8) Reading the AD Conversion Result

The results of AD conversion are stored in ADREG04 to ADREG37 registers. When the contents of one of the lower 2 bits register (ADREGxL) are read, ADMOD1 <EOCF> will be cleared to "0". <EOCF> is not cleared to "0" when the contents of one of the upper 8 bits register (ADREGxH) are read, where x corresponds to the channel number.

Setting example: ① When the analog input voltage of the AN3 pin is AD converted by 160 states speed and the result is transfered to the memory address 0100H by AD interrupt INTAD routine

Main setting

```
INTE0AD  ← 1 1 0 0 - - - -    Enable INTAD and set interrupt level 4.
ADMOD2   ← 1 X 0 0 X 0 1 1    Specify AN3 pin as an analog input channel and
ADMOD1   ← X X 0 0 X 1 X X    starts AD conversion in 160 states speed mode.
```

Interrupt processing

```
WA          ← (ADREG37)       Read ADREG37L and ADREG37H values and write
                              to WA (16 bit)
WA          >> 6              Right-shifts WA six times and writes 0 in upper
                              bits.
(000100H)←   WA               Writes contents of WA in memory at 0100H
```

② When the analog input voltage of the AN0 to AN3 pins (4 pins) are AD converted by 320 states speed and set the channel scan and repeat mode.

```
INTE0AD  ← 1 0 0 0 - - - -    INTAD disenable.
ADMOD2   ← 1 X 0 1 X 0 1 1    Specify AN0 to AN3 pins as input channel and scan
ADMOD1   ← X X 1 1 X 1 0 0    and repeat mode and starts AD conversion.
```

Note : X : Don't care - : No change

3.12 Watchdog Timer (Runaway Detecting Timer), Warming Up Timer

TMP93CS42A contains a watchdog timer for Runaway detection.
The watchdog timer (WDT) is used to return the CPU to a normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt to notify the CPU of the malfunction.
Connecting the watchdog timer detect signal 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 warming up timer for the internal oscillator stabilization. This is used for STOP releasing.

3.12.1 Configuration

Figure 3.12.1 shows the block diagram of the watchdog timer (WDT).

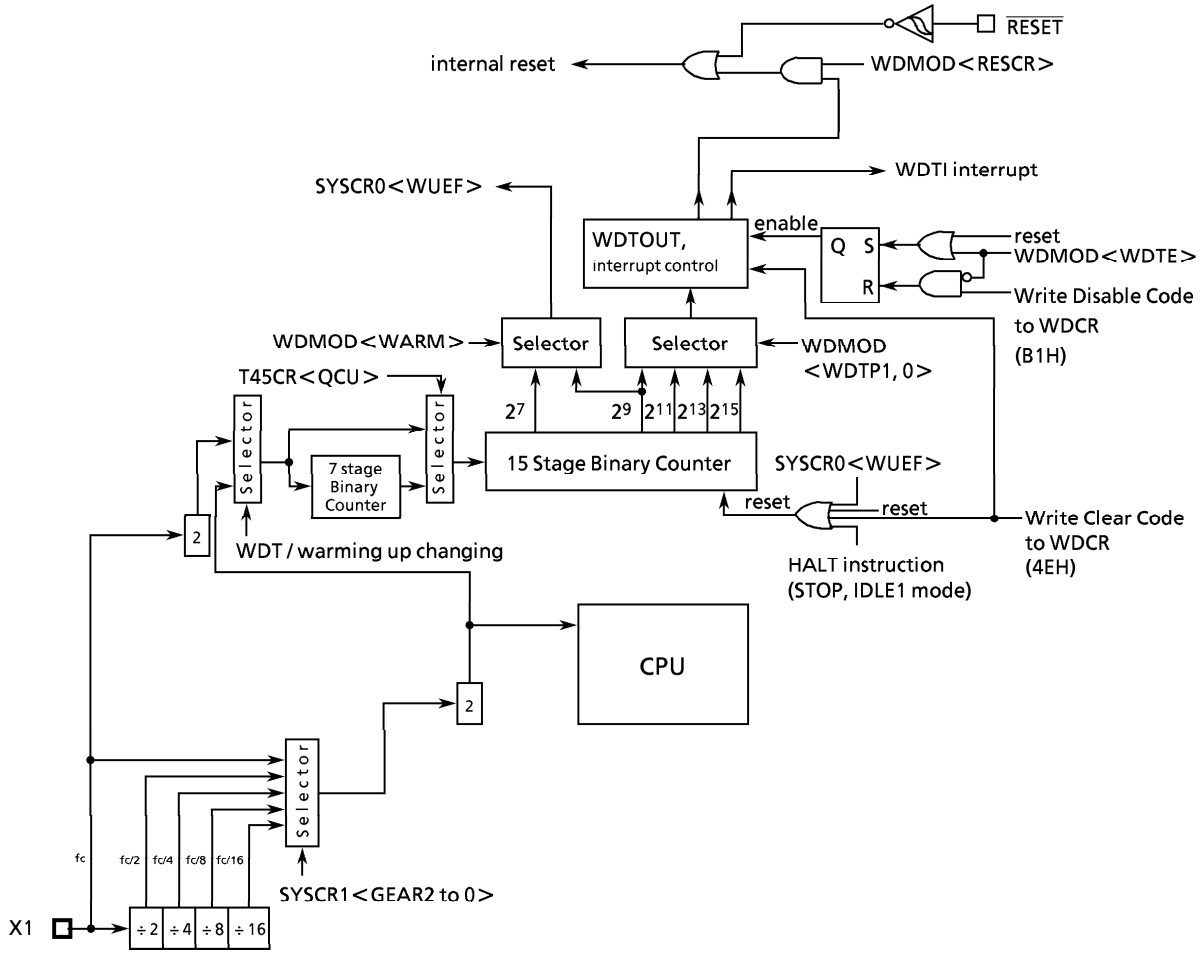


Figure 3.12.1 Block Diagram of Watchdog Timer / Warming up Timer

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 these outputs with in the WDMOD register generates a watchdog interrupt, and outputs watchdog timer out when an overflow occurs.

For the warming-up counter, 2^7 and 2^9 outputs of 15-stage binary counter can be selected using WDMOD<WARM> register. When a stable-external oscillator is used, shorter warming-up time is available using T45CR<QCU> register. When <QCU> = 1, counting value of 2^7 is selected.

When the watchdog timer is in operation, this shorter warming-up time function cannot be selected. The warming-up timer function can be available by setting <QCU> = 0.

Example:

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

The watchdog timer out pin can also be connected to the reset pin internally.

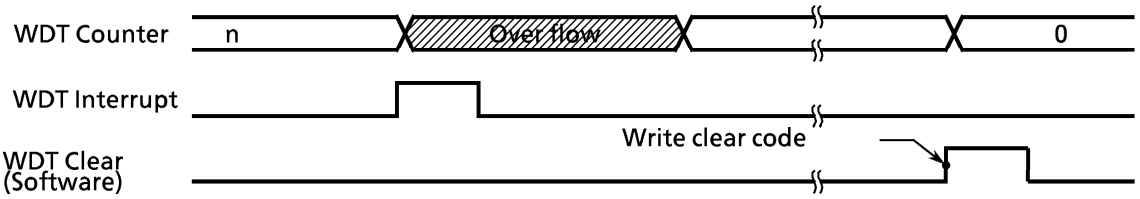


Figure 3.12.2 Normal Mode

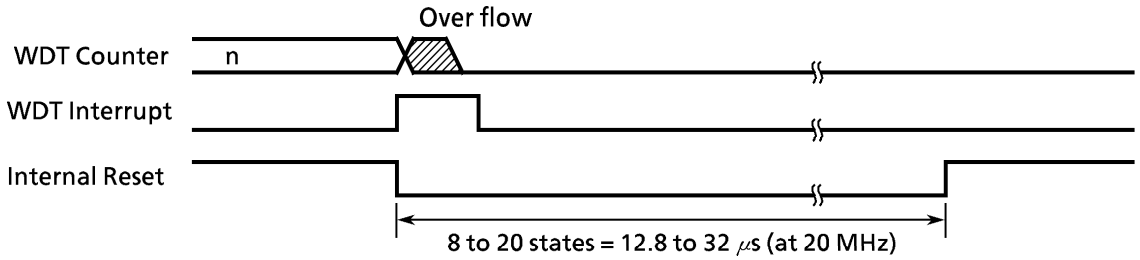


Figure 3.12.3 Reset Mode

3.12.2 Control Registers

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

(1) Watchdog Timer Mode Register (WDMOD)

① Setting the detecting time of watchdog timer <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD<WDTP1, 0> = "00" when reset.
The detecting time of WDT is shown Figure 3.12.6.

② Watchdog timer enable/disable control <WDTE>

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

③ Watchdog timer out reset connection <RESCR>

This bit is used to connect the output of the watchdog timer with RESET 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 of 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 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 Figure 3.12.6		Warming Up time	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Reset Internally connects WDT out to the reset pin	1: Drive the pin even in STOP mode

DRVE (explanation by stop mode)

Watchdog timer out control

0	—
1	Connects WDT out to reset

Select the standby mode HALT instruction

00	RUN mode (Only the CPU stops)
01	STOP mode (All circuits stop)
10	IDLE1 mode (Only the oscillator operates)
11	IDLE2 mode (CPU and AD stop)

Select the Warming-up time at fc = 20 MHz

Gear value <GEAR2 to 0>	Warming-up time		
	T45CR<QCU> = 0		T45CR<QCU> = 1
	WARM = 0	WARM = 1	WARM = X
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

Watchdog timer Enable / Disable control

0	Disable
1	Enable

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	—

Figure 3.12.5 Watchdog Timer Control Register

at fc = 20 MHz

Gear value <GEAR2 to 0>	Watch Dog Timer Detecting Time WDMOD<WDTP1, 0>			
	00	01	10	11
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

Note : When using as the watchdog timer, write "0" to T45CR<QCU> bit .

Figure 3.12.6 Watchdog Timer Detecting Time

3.12.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1, 0> and T45CR<QCU> registers. For normal operation, 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 use of recovery program.
The watchdog timer restarts operation immediately after reset is released.
The watchdog timer stops its operation in the IDLE1 and STOP modes. In the RUN mode, the watchdog timer is operational. When the bus is released ($\overline{BUSAK}$ = "L"), WDT continues counting.
The watchdog timer is enabled in IDLE2 mode, but the over flow interrupt is disabled. Disable the watchdog timer before entering 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 $2^{18} / f_{SYS}$

```
WDMOD ← 1 0 1 - - - X X
```

③ Disable the watchdog timer.

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

④ Set IDLE1 mode.

```
WDMOD ← 0 - - - 1 0 X X         Disables WDT and sets IDLE1 mode.
WDCR  ← 1 0 1 1 0 0 0 1
Executes HALT command           Set the standby mode
```

⑤ Set the STOP mode (warming up time: $2^{16} / f_{SYS}$)

```
WDMOD ← - - - 1 0 1 X X         Set the STOP mode.
Executes HALT command.          Execute HALT instruction. Set the standby
                                mode.
```

4. Electrical Characteristics

4.1 Absolute Maximum Ratings (TMP93CS42AF)

"X" used in an expression shows a frequency of clock f_{PH} selected by SYSCR1<SYSCK>. If a clock gear 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").

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC}	− 0.5 to 6.5	V
Input Voltage	V_{IN}	− 0.5 to V_{CC} + 0.5	V
Output Current (total)	ΣI_{OL}	120	mA
Output Current (total)	ΣI_{OH}	− 80	mA
Power Dissipation (Ta = 85°C)	P_D	600	mW
Soldering Temperature (10 s)	T_{SOLDER}	260	°C
Storage Temperature	T_{STG}	− 65 to 150	°C
Operating Temperature	T_{OPR}	− 40 to 85	°C

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition	Min	Typ. (Note)	Max	Unit
Power Supply Voltage ($AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS} = 0$ V)		V_{CC}	fc = 4 to 20 MHz	4.5		5.5	V
Input Low Voltage	AD0 to 15	V_{IL}	V_{CC} = 5 V ± 10%	−0.3		0.8	V
	Port2 to A (except P87, P5)	V_{IL1}				0.3 V_{CC}	
	RESET, NMI	V_{IL2}				0.25 V_{CC}	
	EA	V_{IL3}				0.3	
	X1	V_{IL4}				0.2 V_{CC}	
	INT0, INT5, INT7, INT8, INT9	V_{IL5}				0.8	
Input High Voltage	AD0 to 15	V_{IH}	V_{CC} = 5 V ± 10%	2.2		V_{CC} + 0.3	
	Port2 to A (except P87)	V_{IH1}		0.7 V_{CC}			
	RESET, NMI	V_{IH2}		0.75 V_{CC}			
	EA	V_{IH3}		V_{CC} − 0.3			
	X1	V_{IH4}		0.8 V_{CC}			
	INT0, INT5, INT7, INT8, INT9	V_{IH5}		2.8			
Output Low Voltage		V_{OL}	I_{OL} = 1.6 mA (V_{CC} = 5 V ± 10%)			0.45	V
Output High Voltage		V_{OH}	I_{OH} = − 400 μA (V_{CC} = 5 V ± 10%)	4.2			

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

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition	Min	Typ.(Note1)	Max	Unit
Darlington Drive Current (8 Output Pins Max)	I_{DAR} (Note2)	V_{EXT} = 1.5 V R_{EXT} = 1.1 kΩ V_{CC} = 5 V ± 10%	– 1.0		– 3.5	mA
Input Leakage Current	I_{LI}	$0.0 \leqq V_{IN} \leqq V_{CC}$		0.02	± 5	μA
Output Leakage Current	I_{LO}	$0.2 \leqq V_{IN} \leqq V_{CC} - 0.2$		0.05	± 10	
Power Down Voltage (at STOP, RAM Back up)	V_{STOP}	V_{IL2} = 0.2 V_{CC}, V_{IH2} = 0.8 V_{CC}	2.0		6.0	V
$\overline{\text{RESET}}$ Pull Up Resister	R_{RST}	V_{CC} = 5 V ± 10%	50		150	kΩ
Pin Capacitance	C_{IO}	fc = 1 MHz			10	pF
Schmitt Width $\overline{\text{RESET}}$, NMI	V_{TH}		0.4	1.0		V
Programmable Pull Down Resistor	R_{KL}	V_{CC} = 5 V ± 10%	10		80	kΩ
Programmable Pull Up Resistor	R_{KH}	V_{CC} = 5 V ± 10%	50		150	
NORMAL (Note3)	I_{CC}	V_{CC} = 5 V ± 10% fc = 20 MHz		19	25	mA
NORMAL2 (Note4)				24	30	
RUN				17	25	
IDLE2				10	15	
IDLE1				3.5	5	
STOP		V_{CC} = 5 V ± 10%		0.2	10	μA

Note 1 : Typical values are for Ta = 25°C and Vcc = 5 V unless otherwise noted.
Note 2 : I_{DAR} is guaranteed for total of up to 8 ports.
Note 3 : The condition of measurement of Icc (NORMAL).
Operates only CPU, output ports are open and input ports fixed.
Note 4 : The condition of measurement of Icc (NORMAL2).
Operates all functions, output ports are open and input port fixed.

4.3 AC Characteristics

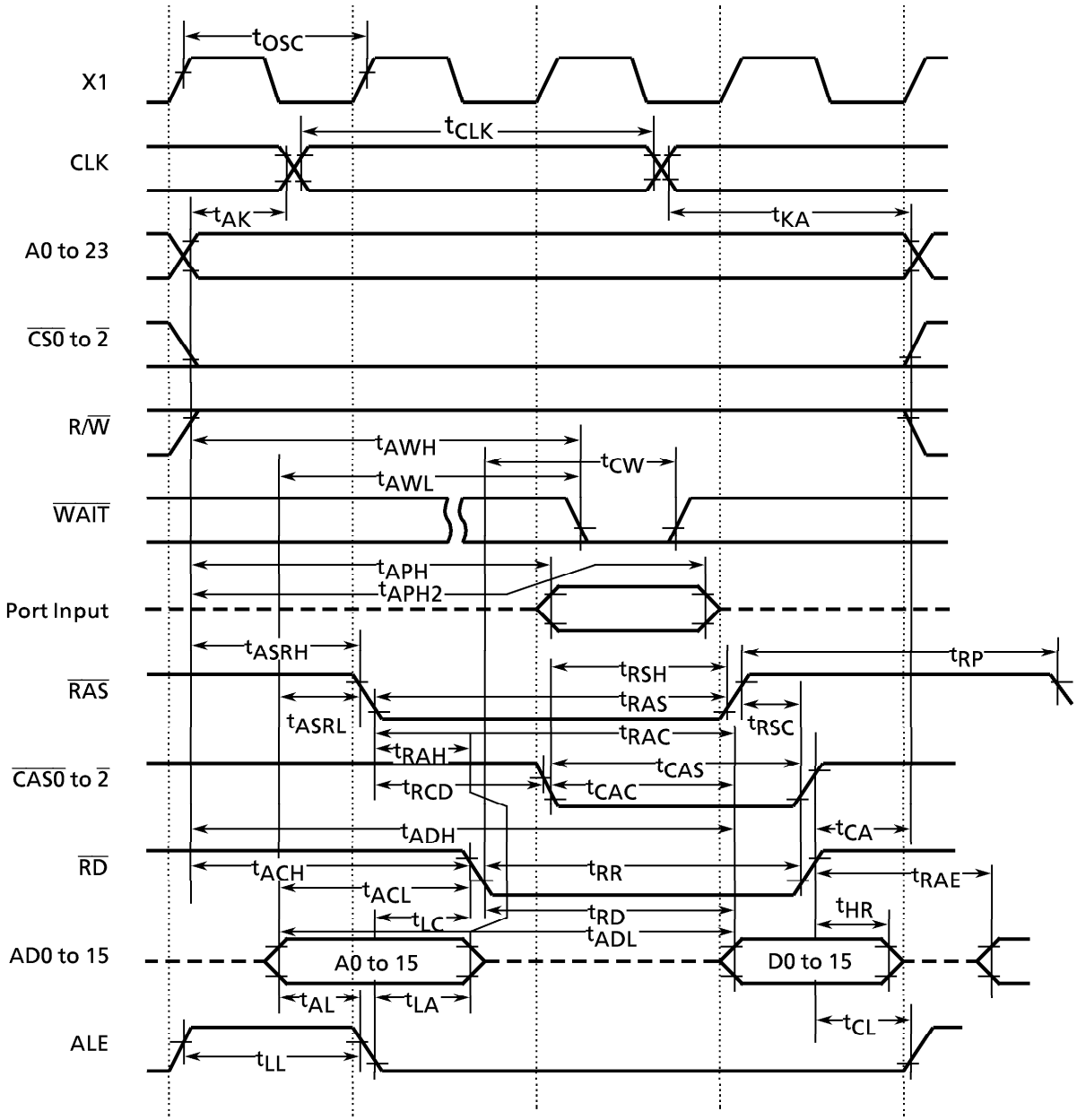
V_{CC} = 5 V ± 10%

No.	Parameter	Symbol	Variable		16 MHz		20 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Osc. Period (= x)	t_{OSC}	50	31250	62.5		50		ns
2	CLK pulse width	t_{CLK}	2x – 40		85		60		ns
3	A0 to 23 Valid→ CLK Hold	t_{AK}	0.5x – 20		11		5		ns
4	CLK Valid→ A0 to 23 Hold	t_{KA}	1.5x – 70		24		5		ns
5	A0 to 15 Valid→ ALE fall	t_{AL}	0.5x – 15		16		10		ns
6	ALE fall→ A0 to 15 Hold	t_{LA}	0.5x – 20		11		5		ns
7	ALE High pulse width	t_{LL}	x – 40		23		10		ns
8	ALE fall→ $\overline{RD}$/$\overline{WR}$ fall	t_{LC}	0.5x – 25		6		0		ns
9	$\overline{RD}$/$\overline{WR}$ rise→ ALE rise	t_{CL}	0.5x – 20		11		5		ns
10	A0 to 15 Valid→ $\overline{RD}$/$\overline{WR}$ fall	t_{ACL}	x – 25		38		25		ns
11	A0 to 23 Valid→ $\overline{RD}$/$\overline{WR}$ fall	t_{ACH}	1.5x – 50		44		25		ns
12	$\overline{RD}$/$\overline{WR}$ rise→ A0 to 23 Hold	t_{CA}	0.5x – 25		6		0		ns
13	A0 to 15 Valid→ D0 to 15 input	t_{ADL}		3.0x – 55		133		95	ns
14	A0 to 23 Valid→ D0 to 15 input	t_{ADH}		3.5x – 65		154		110	ns
15	$\overline{RD}$ fall → D0 to 15 input	t_{RD}		2.0x – 60		65		40	ns
16	$\overline{RD}$ Low pulse width	t_{RR}	2.0x – 40		85		60		ns
17	$\overline{RD}$ rise→ D0 to 15 Hold	t_{HR}	0		0		0		ns
18	$\overline{RD}$ rise→ A0 to 15output	t_{RAE}	x – 15		48		35		ns
19	$\overline{WR}$ Low pulse width	t_{WW}	2.0x – 40		85		60		ns
20	D0 to 15 Valid→ $\overline{WR}$ rise	t_{DW}	2.0x – 55		70		45		ns
21	$\overline{WR}$ rise →D0 to 15 Hold	t_{WD}	0.5x – 15		16		10		ns
22	A0 to 23 Valid→ $\overline{WAIT}$ input $\left(\begin{smallmatrix}2WAIT\\+n\ mode\end{smallmatrix}\right)$	t_{AWH}		5.5x – 90		254		185	ns
23	A0 to 15 Valid→ $\overline{WAIT}$ input $\left(\begin{smallmatrix}2WAIT\\+n\ mode\end{smallmatrix}\right)$	t_{AWL}		5.0x – 80		223		170	ns
24	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ Hold $\left(\begin{smallmatrix}2WAIT\\+n\ mode\end{smallmatrix}\right)$	t_{CW}	4.0x + 0		250		200		ns
25	A0 to 23 Valid→ PORT input	t_{APH}		2.5x – 120		36		5	ns
26	A0 to 23 Valid→ PORT Hold	t_{APH2}	2.5x + 50		206		175		ns
27	$\overline{WR}$ rise→ PORT Valid	t_{CP}		200		200		200	ns
28	A0 to 23 Valid→ $\overline{RAS}$ fall	t_{ASRH}	1.0x – 40		23		10		ns
29	A0 to 15 Valid→ $\overline{RAS}$ fall	t_{ASRL}	0.5x – 15		16		10		ns
30	$\overline{RAS}$ fall→ D0 to 15 input	t_{RAC}		2.5x – 70		86		55	ns
31	$\overline{RAS}$ fall→ A0 to 15 Hold	t_{RAH}	0.5x – 15		16		10		ns
32	$\overline{RAS}$ Low pulse width	t_{RAS}	2.0x – 40		85		60		ns
33	$\overline{RAS}$ High pulse width	t_{RP}	2.0x – 40		85		60		ns
34	$\overline{CAS}$ fall→ $\overline{RAS}$ rise	t_{RSH}	1.0x – 40		23		10		ns
35	$\overline{RAS}$ rise→ $\overline{CAS}$ rise	t_{RSC}	0.5x – 25		6		0		ns
36	$\overline{RAS}$ fall→ $\overline{CAS}$ fall	t_{RCD}	1.0x – 40		23		10		ns
37	$\overline{CAS}$ fall→ D0 to 15 input	t_{CAC}		1.5x – 65		29		10	ns
38	$\overline{CAS}$ Low pulse width	t_{CAS}	1.5x – 30		64		40		ns

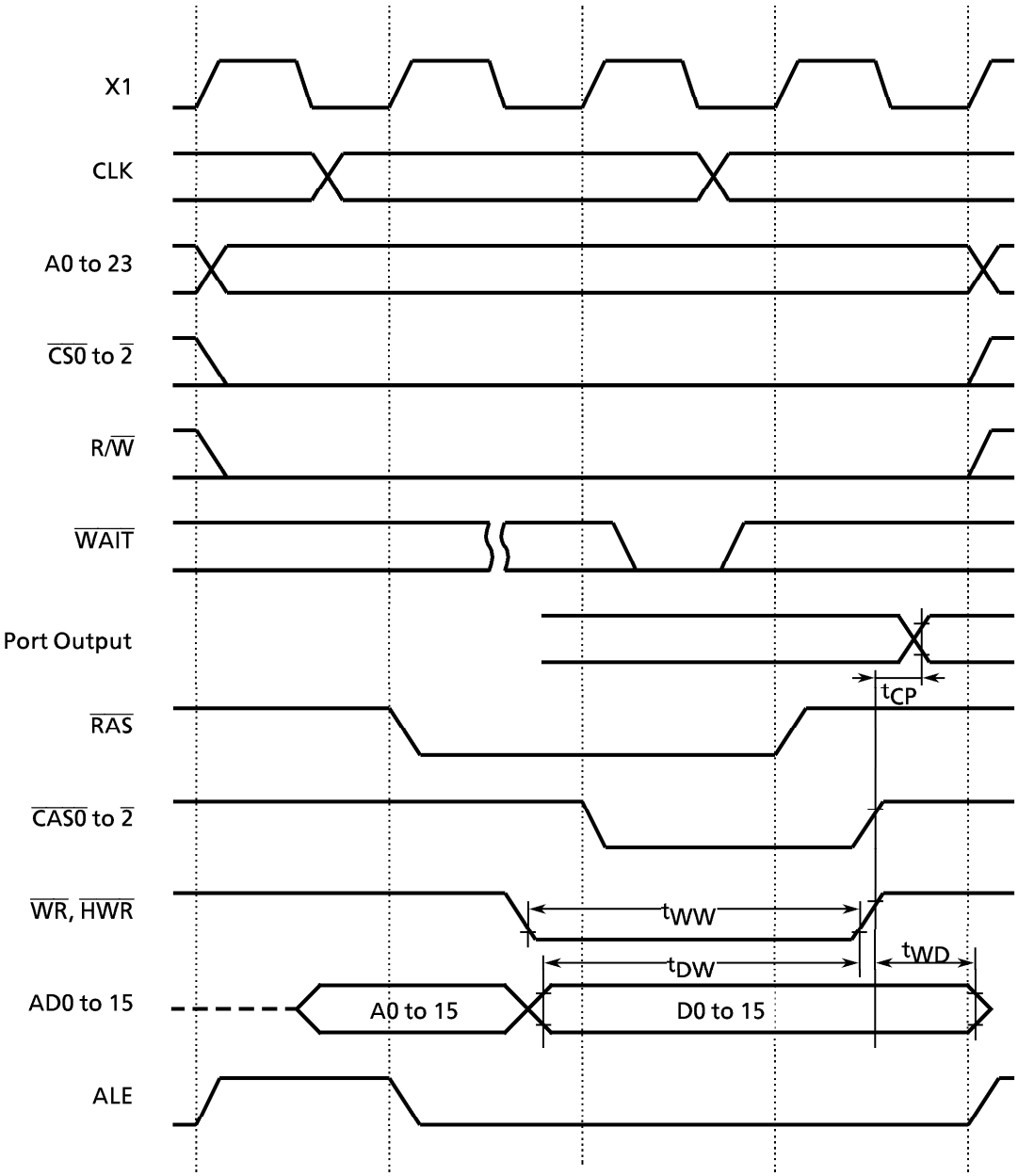
AC Measuring Conditions

- 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, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, CLK, $\overline{RAS}$, $\overline{CAS0}$ to $\overline{CAS2}$)
- Input Level : High 2.4 V / Low 0.45 V (AD0 to AD15)
  High 0.8 × V_{CC} / Low 0.2 × V_{CC} (Except for AD0 to AD15)

(1) Read Cycle


X1
CLK
A0 to 23
CS0 to 2
R/W
WAIT
Port Input
RAS
CAS0 to 2
RD
AD0 to 15
ALE
tOSC
tCLK
tAK
tKA
tAWH
tAWL
tCW
tAPH
tAPH2
tASRH
tASRL
tRSH
tRAS
tRAC
tRAH
tRCD
tCAS
tCAC
tRSC
tRP
tADH
tACH
tACL
tLC
tRR
tRD
tADL
tCA
tRAE
tHR
A0 to 15
D0 to 15
tAL
tLA
tLL
tCL

(2) Write Cycle



4.4 AD Conversion Characteristics

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

Parameter	Symbol	Power Supply	Min	Typ	Max	Unit
Analog reference voltage (+)	V_{REFH}	V_{CC} = 5 V ± 10%	V_{CC} – 1.5 V	V_{CC}	V_{CC}	V
Analog reference voltage (–)	V_{REFL}		V_{SS}	V_{SS}	V_{SS} + 0.2 V	
Analog input voltage range	V_{AIN}		V_{REFL}		V_{REFH}	
Analog current for analog reference voltage <VREFON> = 1	I_{REF} (V_{REFL} = 0 V)			0.5	1.5	mA
<VREFON> = 0				0.02	5.0	μA
Error (excluding quantizing error)	—			± 1.0	± 3.0	LSB

Note 1 : $1LSB = (V_{REFH} - V_{REFL}) / 2^{10}$

Note 2 : The operation above is guaranteed with $f_{FPH} \geqq 4$ MHz.

Note 3 : The value I_{CC} includes the current which flows through AVCC pin.

4.5 Serial Channel Timing

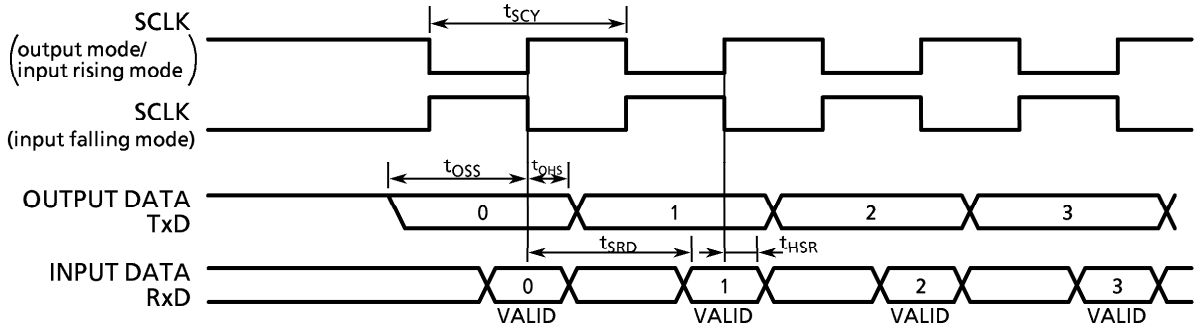
① SCLK Input Mode

Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle	t_{SCY}	16x		0.8		μs
Output Data → SCLK rising/falling timing*	t_{OSS}	$t_{SCY}/2 - 5x - 50$		100		ns
SCLK rising/falling timing* → Output Data hold	t_{OHS}	5x – 100		150		ns
SCLK rising/falling timing* → Input Data hold	t_{HSR}	0		0		ns
SCLK rising/falling timing* → effective data input	t_{SRD}		$t_{SCY} - 5x - 100$		450	ns

*) SCLK rising/falling timing ··· SCLK rising in the rising mode of SCLK, SCLK falling in the falling mode of SCLK.

② SCLK Output Mode

Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	16x	8192x	0.8	409.6	μs
Output Data → SCLK rising edge	t_{OSS}	$t_{SCY} - 2x - 150$		550		ns
SCLK rising edge → Output Data hold	t_{OHS}	2x – 80		20		ns
SCLK rising edge → Input Data hold	t_{HSR}	0		0		ns
SCLK rising edge → effective data input	t_{SRD}		$t_{SCY} - 2x - 150$		550	ns



4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7)

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

4.7 Interrupt and Capture

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

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

(2) INT4 to 9

INT4 to 9 input pulse width depends on the operation clock of CPU and Timer (9 bit prescaler). The following shows the pulse width for each clock.

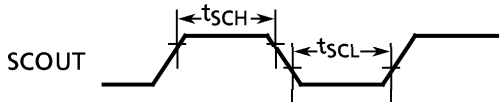
Clock selection for Prescaller <PRCK1, 0>	t_{INTBL} (INT4 to 9 Low level Pulse width)		t_{INTBH} (INT4 to 9 High level Pulse width)		Unit
	Variable	20 MHz	Variable	20 MHz	
	Min	Min	Min	Min	
00 (f_{FPH})	8X + 100	500	8X + 100	500	ns
10(fc/16)	128X + 0.1	6.5	128X + 0.1	6.5	μs

4.8 SCOUT pin AC characteristics

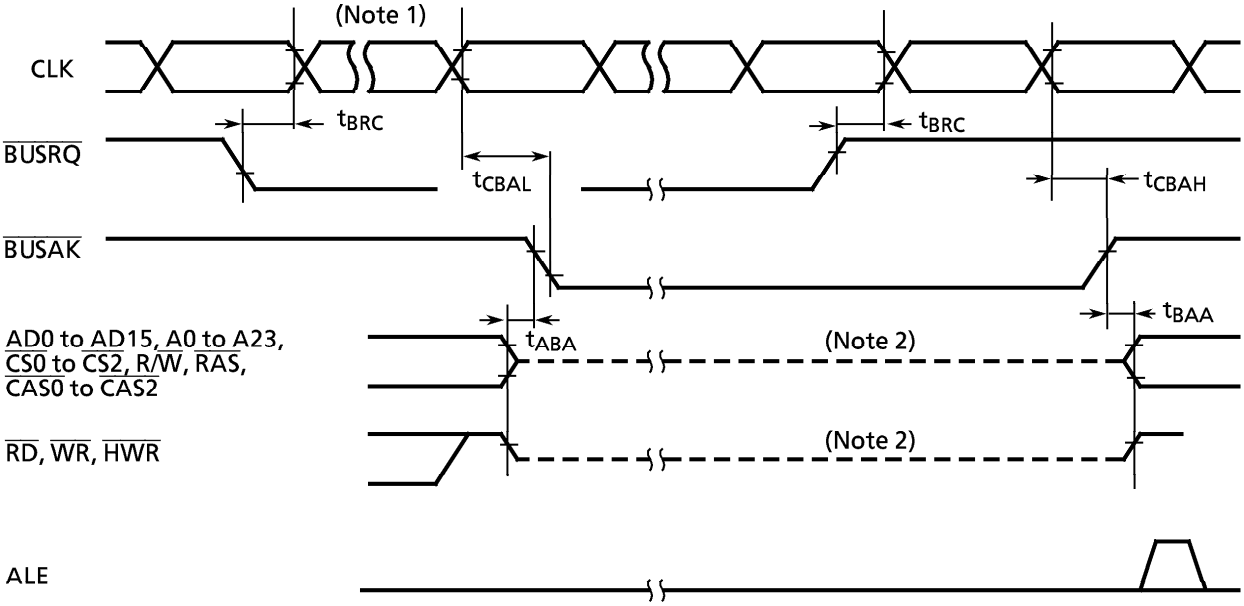
Parameter	Symbol	Variable		20 MHz	
		Min	Max	Min	Max
High-level pulse width VCC = 5V ± 10%	t_{SCH}	0.5X–10		15	
Low-level pulse width VCC = 5V ± 10%	t_{SCL}	0.5X–10		15	

Measurement condition

- Output level : High 2.2 V / Low 0.8 V, CL = 10 pF



4.9 Timing Chart for Bus Request/Bus Acknowledge



Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
$\overline{BUSRQ}$ set-up time to CLK	t_{BRC}	120		120		ns
CLK→ $\overline{BUSAK}$ falling edge	t_{CBAL}		1.5x + 120		195	ns
CLK→ $\overline{BUSAK}$ rising edge	t_{CBAH}		0.5x + 40		65	ns
Output Buffer is off to $\overline{BUSAK}$	t_{ABA}	0	80	0	80	ns
$\overline{BUSAK}$ to Output Buffer is on.	t_{BAA}	0	80	0	80	ns

Note 1: The Bus will be released after the $\overline{WAIT}$ request is inactive, when the $\overline{BUSRQ}$ is set to "0" during "Wait" cycle.

Note 2: This line only shows the output buffer is off-state.
It doesn't indicate the signal level is fixed.
Just after the bus is released, the signal level which is set before the bus is released is kept dynamically by the external capacitance. Therefore, to fix the signal level by an external resistor during bus releasing, designing is executed carefully because the level-fix will be delayed.
The internal programmable pull-up / pull-down resistor is switched active / non-active by an internal signal.

5. Table of Special Function Registers (SFRs)

(SFR ; Special Function Register)

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

(1) I/O port

(2) I/O port control

(3) Timer control

(4) Watch Dog Timer control

(5) Serial Channel control

(6) AD converter control

(7) Interrupt control

(8) Chip Select / Wait control

(9) Clock Control

Configuration of the table

Symbol	Name	Address	7	6			1	0	
									➔Bit symbol
									➔Read / Write
									➔Initial value afrer reset
									➔Remarks

Note : "Prohibit RMW" in table means that you cannot use RMW instructions to these registers.
(Example) In case of setting only the bit0 of register P0CR, you mustn't use "Set 0, (0002H)"

Table 5 I/O register address map

Address	Name	Address	Name	Address	Name	Address	Name
000000H	P0	20H	TRUN	40H	TREG6L	60H	ADREG04L
1H	P1	21H		41H	TREG6H	61H	ADREG04H
2H	P0CR	22H	TREG0	42H	TREG7L	62H	ADREG15L
3H		23H	TREG1	43H	TREG7H	63H	ADREG15H
4H	P1CR	24H	TMOD	44H	CAP3L	64H	ADREG26L
5H	P1FC	25H	TFFCR	45H	CAP3H	65H	ADREG26H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG37L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG37H
8H	P2CR	28H	P0MOD	48H	T5MOD	68H	B0CS
9H	P2FC	29H	P1MOD	49H	T5FFCR	69H	B1CS
AH	P3CR	2AH	PFFCR	4AH		6AH	B2CS
BH	P3FC	2BH		4BH		6BH	
CH	P4	2CH		4CH		6CH	
DH	P5	2DH		4DH		6DH	CKOCR
EH	P4CR	2EH		4EH		6EH	SYSCR0
FH		2FH		4FH		6FH	SYSCR1
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H		31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTE89
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	P7FC	37H	CAP2H	57H	BR1CR	77H	INTES0
18H	P8	38H	T4MOD	58H	ODE	78H	INTES1
19H	P9	39H	T4FFCR	59H		79H	
1AH	P8CR	3AH	T45CR	5AH		7AH	
1BH	P9CR	3BH		5BH		7BH	IIMC
1CH	P8FC	3CH		5CH	WDMOD	7CH	DMA0V
1DH	P9FC	3DH		5DH	WDCR	7DH	DMA1V
1EH	PA	3EH		5EH	ADMOD1	7EH	DMA2V
1FH	PACR	3FH		5FH	ADMOD2	7FH	DMA3V

Note : Do not access the addresses without allocated register names.

(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
			* R/W							
			0	0	0	0	0	0	0	0
			Input mode							
P3	PORT3	07H	P37	P36	P35	P34	P33	P32	P31	P30(Note1)
			* R/W							
			1	1	1	1	1	1	1	1
			Input mode						Output mode	
P4	PORT4	0CH						P42	P41	P40
								* R/W		
								0	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
			* R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P7	PORT7	13H					P73	P72	P71	P70
							* R/W			
							1	1	1	1
							Input mode			
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			* R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P9	PORT9 (note2)	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W	R/W	* R/W					
			1	1	1	1	1	1	1	1
			Output mode	Output mode	Input mode					
PA	PORTA	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			1							
			Input mode							

Note 1 : When P30 pin is defined as $\overline{RD}$ signal output mode (P30F = 1), the output latch of P30 is output to P61 pin.

Note 2 : Port 96, 97 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)
* R/W ; Read-modify-write is prohibited when controlling the PU/PD resistors.

(2) I/O Port Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	PORT0 Control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0 : IN 1 : OUT (When external access, set as AD7-0 and cleared to "0".)							
P1CR	PORT1 Control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P1FC">>							
P1FC	PORT1 Function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00 : IN, 01 : OUT, 10 : AD15-8, 11 : A15-8							
P2CR	PORT2 Control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P2FC">>							
P2FC	PORT2 Function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00 : IN, 01 : OUT, 10 : A7-0, 11 : A23-16							
P3CR	PORT3 Control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C		
			W							
			0	0	0	0	0	0		
			0 : IN			1 : OUT				
P3FC	PORT3 Function	0BH (Prohibit RMW)	P37F	P36F	P35F	P34F		P32F	P31F	P30F
			W							
			0	0	0	0		0	0	0
			0 : PORT 1 : $\overline{RAS}$	0 : PORT 1 : R/$\overline{W}$	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$		0 : PORT 1 : $\overline{HWR}$	0 : PORT 1 : $\overline{WR}$	0 : PORT 1 : $\overline{RD}$
P4CR	PORT4 Control	0EH (Prohibit RMW)						P42C	P41C	P40C
									W	
								0	0	0
									0 : IN	1 : OUT
P4FC	PORT4 Function	10H (Prohibit RMW)						P42F	P41F	P40F
									W	
								0	0	0
								0 : PORT		1 : $\overline{CS}$/$\overline{CAS}$

I/O Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	PORT6 Control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0 : IN				1 : OUT			
P7CR	PORT7 Control	15H (Prohibit RMW)					P73C	P72C	P71C	P70C
							W			
							0	0	0	0
							0 : IN		1 : OUT	
P7FC	PORT7 Function	17H (Prohibit RMW)					P73F	P72F	P71F	
							W			
							0	0	0	
							0 : PORT 1 : TO3	0 : PORT 1 : TO2	0 : PORT 1 : TO1	
P8CR	PORT8 Control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0 : IN				1 : OUT			
P9CR	PORT9 Control	1BH (Prohibit RMW)	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W	W	W					
			1	1	0	0	0	0	0	0
			0 : IN				1 : OUT			
P8FC	PORT8 Function	1CH (Prohibit RMW)		P86F			P83F	P82F		
				W			W	W		
				0			0	0		
				0 : PORT 1 : TO6			0 : PORT 1 : TO5	0 : PORT 1 : TO4		
P9FC	PORT9 Function	1DH (Prohibit RMW)			P95F		P93F			P90F
					W		W	W		W
					0		0	0		0
					0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	Set to 0		0 : PORT 1 : TxD0
PACR	PORTA Control	1FH (Prohibit RMW)	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0							
			0 : IN				1 : OUT			

(3) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
			Prescaler and Timer Run / Stop CONTROL 0 : Stop and Clear 1 : Run (Count up)							
TREG0	8-bit timer Register 0	22H (Prohibit RMW)	–							
			W							
			Undefined							
TREG1	8-bit timer Register 1	23H (Prohibit RMW)	–							
			W							
			Undefined							
TMOD	8-bit timer Source CLK & MODE	24H (Prohibit RMW)	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			W							
			0	0	0	0	0	0	0	0
			00 : 8-bit timer 01 : 16-bit timer 10 : 8-bit PPG 11 : 8-bit PWM		00 : – 01 : $2^6 - 1$ 10 : $2^7 - 1$ 11 : $2^8 - 1$	PWM	00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 入力 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TFFCR	8-bit timer Flip-Flop Control	25H				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS
						R/W	W		R/W	
						0	1	1	0	0
						1 : Double Buffer Enable	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	0 : Inverted by Timer 0
TREG2	PWM Timer Register 2	26H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
TREG3	PWM Timer Register 3	27H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
P0MOD	PWM0 Mode	28H (Prohibit RMW)	FF2RD	DB2EN		PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0
			R	W		W				
			–	0		0	0	0	0	0
			TFF2 output value	1 : Double Buffer Enable		0 : PWM Mode 1 : Timer Mode	00 : ϕP1 01 : ϕP4 10 : ϕP16 11 : Don't care		00 : $2^6 - 1$ 01 : $2^7 - 1$ 10 : $2^8 - 1$ 11 : Don't care	
P1MOD	PWM1 Mode	29H (Prohibit RMW)	FF3RD	DB3EN		PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0
			R	W		W				
			–	0		0	0	0	0	0
			TFF3 output value	1 : Double Buffer Enable		0 : PWM Mode 1 : Timer Mode	00 : ϕP1 01 : ϕP4 10 : ϕP16 11 : Don't care		00 : $2^6 - 1$ 01 : $2^7 - 1$ 10 : $2^8 - 1$ 11 : Don't care	

Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PFFCR	PWM Flip-Flop Control	2AH	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00 : Don't care 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		00 : Prohibit TFF3 invert 01 : Invert if matched 10 : Set if matched; clear if overflowed 11 : Clear if matched; set if overflowed		00 : Don't care 01 : Set TFF2 10 : Clear TFF2 11 : Don't care		00 : Prohibit TFF2 invert 01 : Invert if matched 10 : Set if matched; clear if overflowed 11 : Clear if matched; set if overflowed	
TREG4L	16-bit timer Register4L	30H (Prohibit RMW)	–							
			W							
			Undefined							
TREG4H	16-bit timer Register4H	31H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5L	16-bit timer Register5L	32H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5H	16-bit timer Register5H	33H (Prohibit RMW)	–							
			W							
			Undefined							
CAP1L	Capture Register1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register2H	37H	–							
			R							
			Undefined							
T4MOD	16-bit timer 4 Source CLK & MODE	38H	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
			R/W		W			R/W		
			0	0	0	0	0	0	0	0
			TFF5 INV TRG 0 : TRG Disable 1 : TRG Enable		0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI4 ↑ TI5 ↑ 10 : TI4 ↑ TI4 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC4 Clear Enable	Source Clock 00 : TI4 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T4FFCR	16-bit timer 4 Flip-Flop Control	39H	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00 : Invert TFF5 01 : Set TFF5 10 : Clear TFF5 11 : Don't care		TFF4 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF4 01 : Set TFF4 10 : Clear TFF4 11 : Don't care	

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
			Watchdog / Warming 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		
					0	0	0	0	0	0
					0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI6 ↑ TI7 ↑ 10 : TI6 ↑ TI6 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC5 Clear Enable	Source Clock 00 : TI6 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T5FFCR	16-bit timer 5 Flip-Flop Control	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	Undefined	
					TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(4) Watch Dog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watchdog Timer Mode	5CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT Enable	00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warming up Time 0: 2^{14}/inputted frequency 1: 2^{16}/inputted frequency	Standby Mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Connect internally WDT out to Reset Pin	1: Drive the pin in STOP mode
WDCR	Watchdog Timer Control Register	5DH	–							
			W							
			–							
			B1H: WDT Disable Code 4EH: WDT Clear Code							

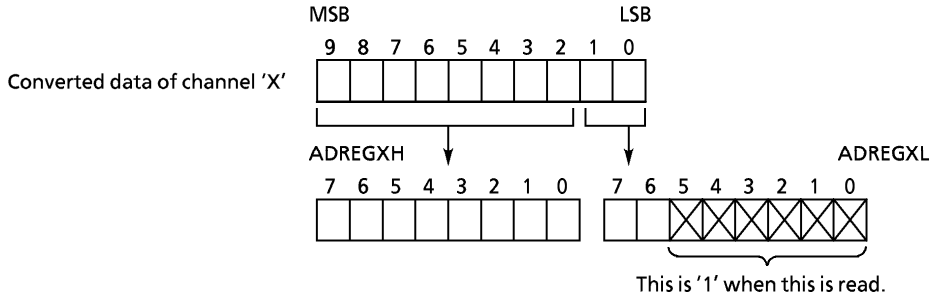
(5) Serial Channel

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	50H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	51H	RB8	EVEN	PE	OERR	PERR	FERR		
			R	R/W		R (Cleared to 0 by reading)				
			undefined	0	0	0	0	0		
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing		
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			undefined	0	0	0	0	0	0	0
			Transmisson data bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: (reserved) 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO0 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	
BR0CR	Baud Rate Control	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F ("1" prohibited)			
SC1BUF	Serial Channel 1 Buffen	54H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			undefined	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1  1: SCLK1	1: Input SCLK1 pin
SC1-MOD	Serial Channel 1 Mode	56H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmisson data bit 8	Fix at "0"	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO0 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	
BR1CR	Baud Rate Control	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F ("1" prohibited)			
ODE	Serial Open Drain Enable	58H							ODE1	ODE0
									R/W	
									0	0
									1:P93 Open-drain	1:P90 Open-drain

(6) AD Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD1	AD Mode Reg 1	5EH	EOCF	ADBF	RPT	SCAN		ADS		
			R		R/W			R/W		
			0	0	0	0		0		
			1: End	1: Busy	1: Repeat	1: Scan		1: Start		
ADMOD2	AD Mode Reg 2	5FH	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
			R/W		R/W			R/W		
			1		0	0		0	0	0
			String Resistance Switch ON/OFF		SPEED			Analog Input Channel Select		
AD REG04L *1)	AD Result Reg 0/4 low	60H	ADR01	ADR00						
			R							
			Undefined		1	1	1	1	1	1
AD REG04H	AD Result Reg 0/4 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG15L *1)	AD Result Reg 1 low	62H	ADR11	ADR10						
			R							
			Undefined		1	1	1	1	1	1
AD REG15H	AD Result Reg 1 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG26L *1)	AD Result Reg 2 low	64H	ADR21	ADR20						
			R							
			Undefined		1	1	1	1	1	1
AD REG26H	AD Result Reg 2 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG37L *1)	AD Result Reg 3 low	66H	ADR31	ADR30						
			R							
			Undefined		1	1	1	1	1	1
AD REG37H	AD Result Reg 3 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

*1 : Data to be stored in AD Conversion Result Reg Low are the lower 2 bits of the conversion result. The contents of the lower 6 bits of this register are always read as "1".



(7) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	INTerrupt Enable 0 & AD	70H (Prohibit RMW)	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5	71H (Prohibit RMW)	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE67	INTerrupt Enable 6/7	72H (Prohibit RMW)	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE-T10	INTerrupt Enable Timer 1/0	73H (Prohibit RMW)	INTT1 (Timer 1)				INTT0 (Timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE89	INTerrupt Enable 8/9	74H (Prohibit RMW)	INT9				INT8			
			I9C	I9M2	I9M1	I9M0	I8C	I8M2	I8M1	I8M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET54	INTerrupt Enable Treg 5/4	75H (Prohibit RMW)	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET76	INTerrupt Enable Treg 7/6	76H (Prohibit RMW)	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	
INTES0	INTerrupt Enable Serial 0	77H (Prohibit RMW)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES1	INTerrupt Enable Serial 1	78H (Prohibit RMW)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

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

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

Interrupt Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request Vector	7CH (Prohibit RMW)				Micro DMA0 start vector				
						DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
						W				
						0	0	0	0	0
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)				Micro DMA1 start vector				
						DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
						W				
						0	0	0	0	0
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)				Micro DMA2 start vector				
						DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
						W				
						0	0	0	0	0
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)				Micro DMA3 start vector				
						DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
						W				
						0	0	0	0	0
IIMC	Interrupt Input Mode Contorol	7BH (Prohibit RMW)						I0IE	I0LE	NMIREE
								W	W	W
								0	0	0
								1: $\overline{\text{INT0}}$ input enable	0: $\overline{\text{INT0}}$ edge mode 1: $\overline{\text{INT0}}$ level mode	1: Operate even at $\overline{\text{NMI}}$ rise edge

(8) Chip Select / Wait Controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	68H (Prohibit RMW)	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B0CS Master bit		0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 2000H to 3FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B1CS	Block 1 CS/WAIT control register	69H (Prohibit RMW)	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B1CS Master bit		0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 4000H to 5FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B2CS	Block 2 CS/WAIT control register	6AH (Prohibit RMW)	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1: B2CS Master bit		0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16-bit bus 1: 8-bit bus	00: 2WAIT 01: 1WAIT 10: 2WAIT + n 11: 0WAIT		00: 6000H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	

Note : After reset, only "Block 2" is set to enable.

(9) Clock Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CKOCR	Clock Output Control Register	006DH					SCOSEL	SCOEN	ALEEN	CLKEN
							R/W			
							0	0	0	0 (Note)
							SCOUT select 0 : f_{FPH} 1 : f_{SYS}	SCOUT Output control 0 : I/O port 1 : SCOUT output	ALE pin control 0 : High-Z output 1 : ALE output	CLK pin control 0 : High-Z output 1 : CLK出力 output
SYSCR0	System Clock Control Register 0	006EH							PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			Fix at "1"	Fix at "0"	Fix at "1"	Fix at "0"	Fix at "0"	Fix at "0"	select prescaler clock 00 : f_{FPH} 01 : (reserved) 10 : fc/16 11 : (reserved)	
SYSCR1	System Clock Control Register 1	006FH						GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							Fix at "0"	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 : During reset, CLK pin is pulled up internally regardless of the products.

6. Port Section Equivalent Circuit Diagram

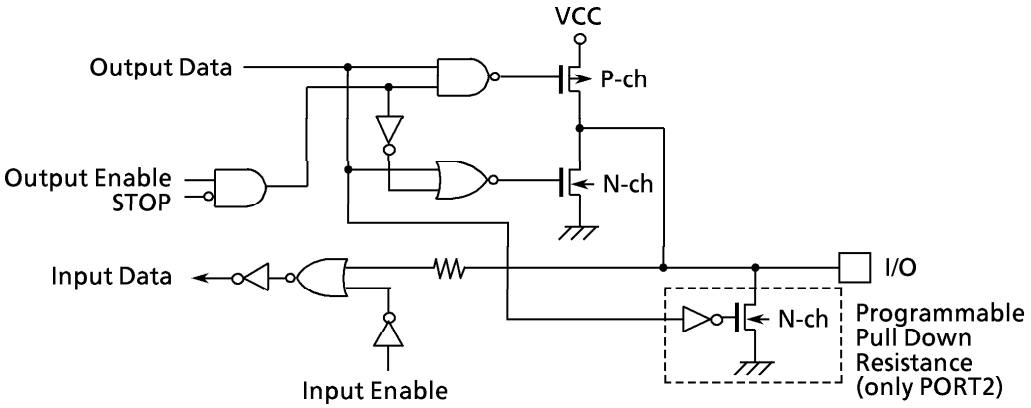
- Reading The Circuit Diagram

  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.

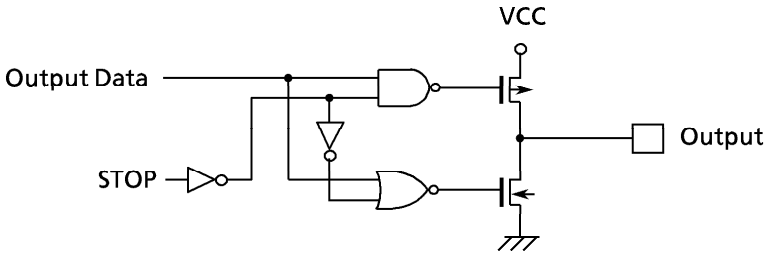
  STOP : This signal becomes active "1" when the halt mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however, STOP remains at "0".

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

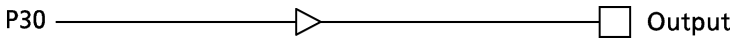
■ P0 (AD0 to AD7), P1 (AD8 to 15, A8 to 15), P2 (A16 to 23, A0 to 7)



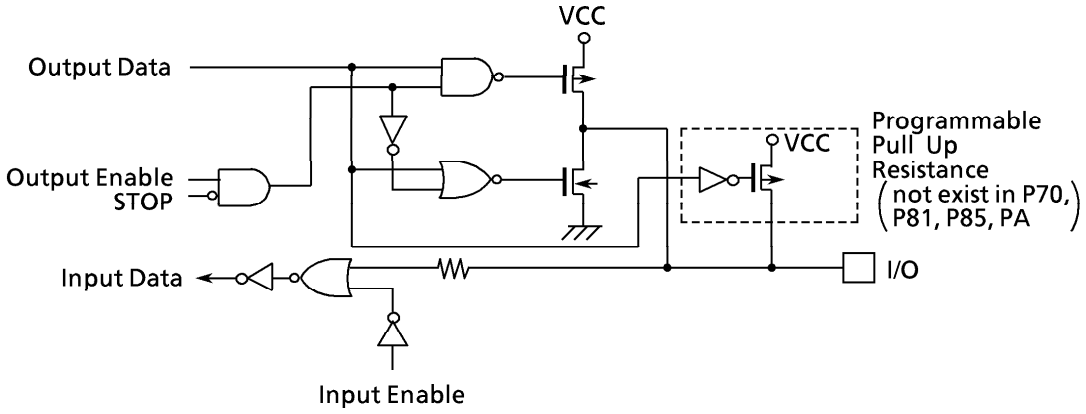
■ P30 ($\overline{RD}$: 72 pin), P31 ($\overline{WR}$)



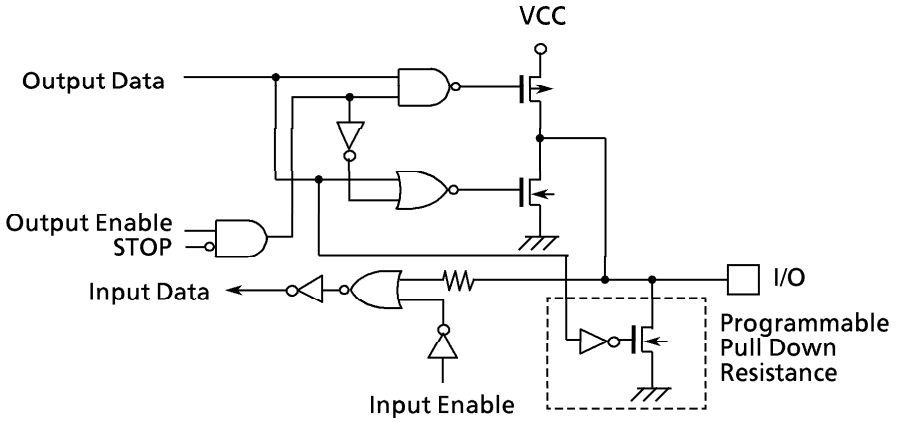
■ P30(61 pin)



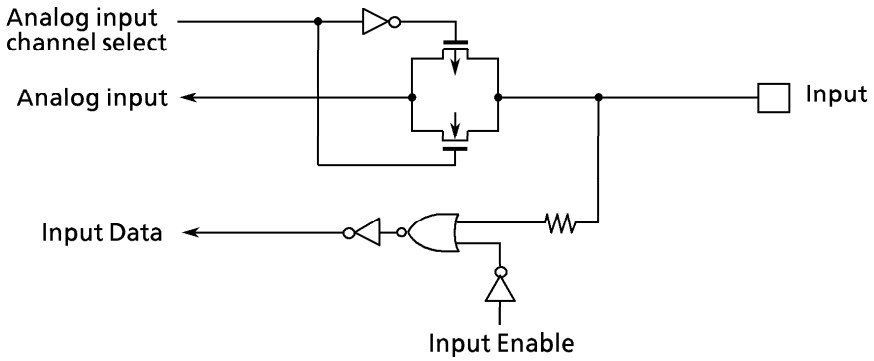
■ P32 to 37, P40 to 41, P6, P7, P80 to 86, P91 to 92, P94 to 95, PA



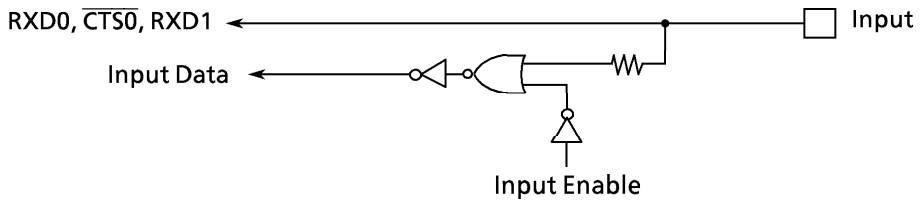
■ P42 ($\overline{CS2}$, $\overline{CAS2}$)



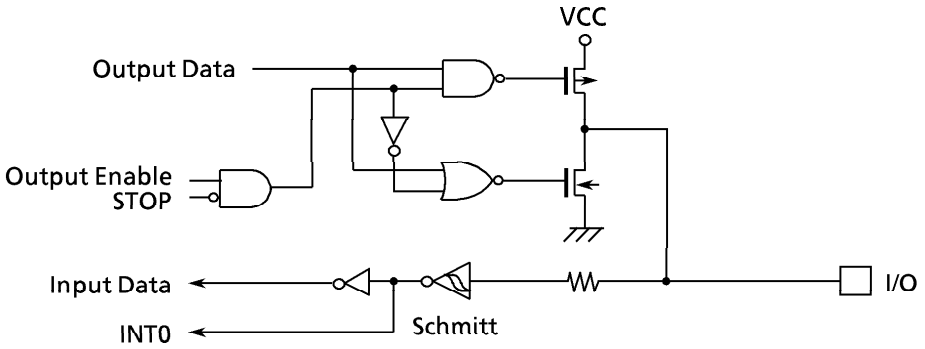
■ P50 to 54 (AN0 to 4)



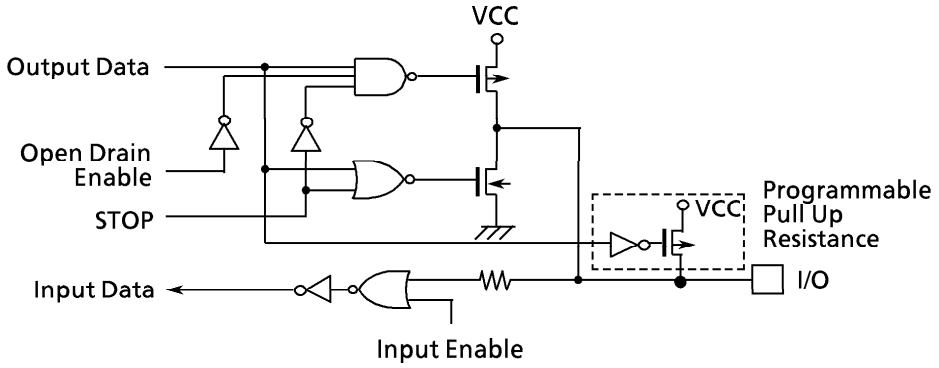
■ P55 to 57 (RXD0, $\overline{CTS0}$, RXD1)



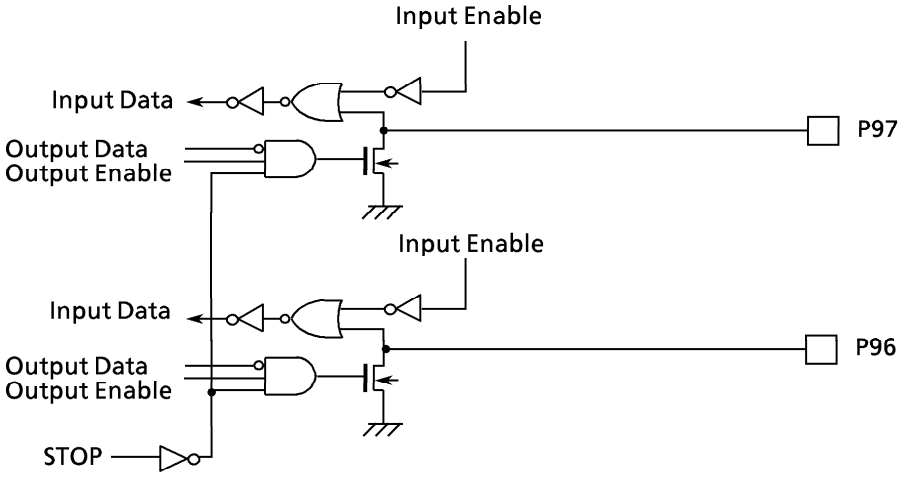
■ P87 ($\overline{INT0}$)



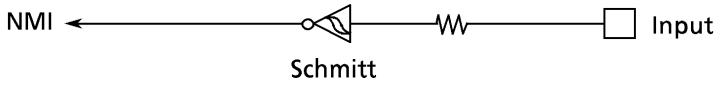
■ P90 (TXD0), P93 (TXD1)



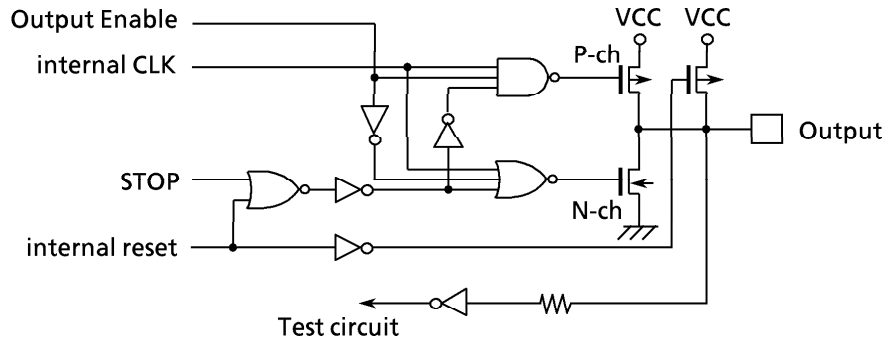
■ P96, P97



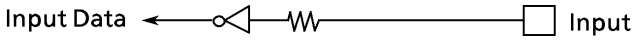
■ $\overline{NMI}$



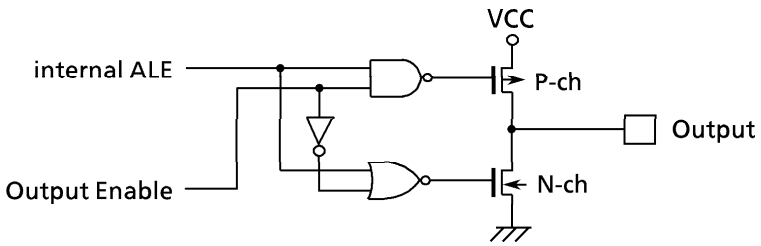
■ CLK



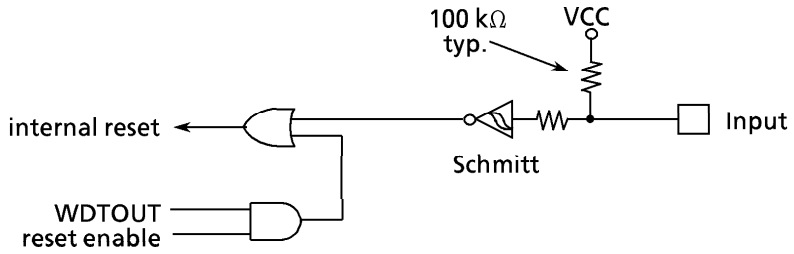
■ $\overline{EA}$, $\overline{INT8}$



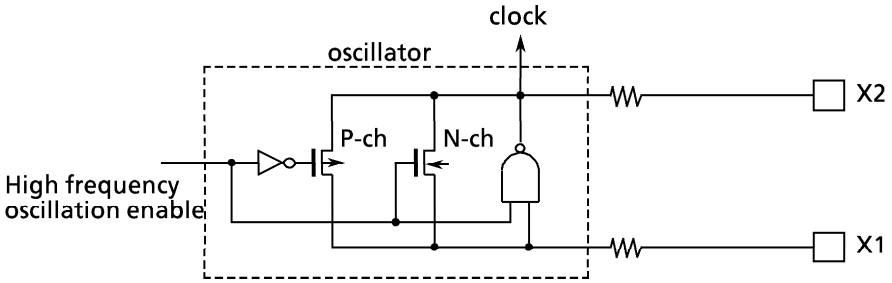
■ ALE



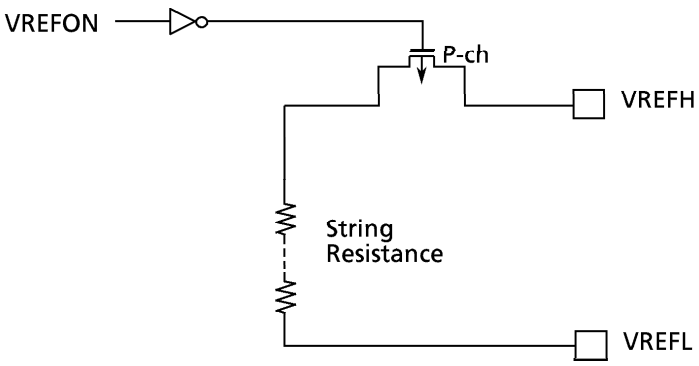
■ $\overline{\text{RESET}}$



■ X1, X2



■ VREFH, VREFL



7. Points of Concern and Restrictions

(1) Special Expression

① Explanation of a built-in I/O register : Register Symbol <Bit Symbol>
ex)TRUN<T0RUN> ··· Bit T0RUN of Register TRUN

② Read, Modify and Write Instruction
An instruction which CPU reads data from memory and writes the data to the same memory location by one instruction.
ex1) SET 3, (TRUN) ··· set bit 3 of TRUN
ex2) INC 1, (100H) ··· increment the data of 100H

- The representative Read, Modify and Write instructions in the TLCS-900

Exchange instruction
EX (mem), R

Arithmetic operation

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

Logic operation

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

Bit manipulation

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

Rotate, Shift

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

③ fc, f_{FPH}, f_{SYS}, 1 state
The clock frequency input from X1, X2 pins is called fc, and the clock selected by SYSCR1<GEAR2 to 0> is called f_{FPH}, and f_{FPH} divided by 2 clock frequency is called f_{SYS}. One cycle of f_{SYS} is called 1 state.

(2) Care Points

① $\overline{EA}$ pin
Fix these pins to Vcc unless changing voltage.

② TEST1, TEST2 pin
Connect the TEST1 pin with the TEST2 pin.

③ Reserved Area in memory space
The 256 bytes memory area between FFFF00H to FFFFFFH can not be used because they are reserved.

④ Standby Mode (IDLE1)
When the IDLE1 mode (operates only oscillator) is used, set TRUN <PRRUN> to "0" to stop prescaler before "HALT" instruction is executed.

⑤ Warmingup Counter
The warmingup counter operates when the STOP mode is released even if the system uses an external oscillator. As a result, it takes the warming up time from inputting the releasing request to outputting the system clock.

⑥ MicroDMA (DRAM refresh mode)
When the Bus is released (BUSAK= "0"), DRAM refresh can not be not performed because of the microDMA cannot access the bus.

⑦ Programmable Pull Up / Down Resistance
The programmable pull up / down resistors can be selected ON / OFF by the program when the ports are used as the input ports. In case where the ports are used as outputs, they can not be selected ON / OFF by program.
The data registers (ex. P2, P3, ...) are used for selecting ON/OFF of pull-up/down resistors. As a result, Read-modify-write instructions are prohibited.

⑧ Bus Releasing Function
Refer to the "Note about the Bus Release" in 3.5 Functions of Ports because the pin state, when the bus is released, is written.

⑨ WatchDog Timer
The watch dog timer starts operation immediately after the reset is released. When the watch dog timer is not used, disable watch dog timer.

⑩ WatchDog Timer
When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate. So, the watch dog timer continues to run. Therefore, be careful about the bus releasing time and set the detection timer of watch dog timer.

⑪ AD Converter
The ladder resistor between VREFH and VREFL pins can be cut by program to reduce the power consumption. When the standby mode is used, disable in the program before the "HALT" instruction is executed. And set ADMOD2<SPEED1, 0> ="00".

⑫ CPU (MicroDMA)
Only the "LDC cr, r", "LDC r, cr" instruction can be used to access the control registers like transfer source address register (DMASn) in the CPU.

⑬ POP SR instruction
Please execute POP SR instruction during DI condition.

⑭ $\overline{INT0}$ pin

This pin is always set to "0" in STOP mode. (See Table 3.3.4 in Standby function in section 3.3.) However $\overline{INT0}$ is used to release STOP mode, so set $\overline{INT0}$ to the edge mode before entering STOP mode or disable $\overline{INT0}$.
See Interrupt controller control 2) External interrupt control in section 3.4.3 for how to set it.

⑮ Internal ROM separation
Inputting "0" to $\overline{EA}$ pin separate a built-in ROM. In this case, the external area is accessed after reset and ROM is externally connected. Please note the following restrictions for using it.

A. No wait is set when accessing the external area after reset, as it is in TMP93CM41, etc. $\overline{CS2}$ is set to "L". That is because the chip select wait controller is different from the others, and the default decode area of CS2 is 6000H to 7FFFH.

B. ALE pin is set to "0" after reset. (At $\overline{EA}$ = 1, ALE pin is high impedance.)

C. In STOP mode, the states of P0, P1, P30 ($\overline{RD}$) and P31 ($\overline{WR}$) pins are as follows.

Pin name	Input/Output	$\overline{EA}$ = "0"		$\overline{EA}$ = "1"	
		DRVE = 0	DRVE = 1	DRVE = 0	DRVE = 1
P0	Input mode/AD0 to 7 Output mode	— ×	— ×	— —	— Output
P1	Input mode / AD8 to 15 Output mode / A8 to 15	— ×	— ×	— —	— Output
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	Output	—	"1" Output	—	Output

— : In the input mode, input is invalid. In the output mode, high impedance is set.
PD* : Programmable pull-down. When setting no pull-down, through current is not set because of input disabled.
× : Impossible to set.

D. The ports of P0, P1, P30 ($\overline{RD}$) and P31 ($\overline{WR}$) access the external areas, and fixed as follows.

	16-bit bus	8-bit bus
P00 to 07	AD0 to 7	AD0 to 7
P10 to 17	AD8 to 15	A8 to 15
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	$\overline{RD}$ / $\overline{WR}$ (Pin 61 is P30 output.)	

The 8-bit bus is set by the chip select wait controller. In the other cases, the 16-bit bus is set.

E. When the external ROM is accessed after reset, the 16-bit bus is set. To use the 8-bit bus ROM, use LDX instruction.

⑯ Pin states in STOP mode

Open-drain output state. Input gate in operation. Set output to "L" or attach pull-up on pin so that the input gate stays constant.

8. TMP93XX40/42A Different Points

Item			TMP93CS40F	TMP93PS40F	TMP93CS42AF	TMP93PS42AF
Operation voltage/frequency			5 V ± 10 % (fc = 4 to 20 MHz) 3 V ± 10 % (fc = 4 to 12.5 MHz)		5 V ± 10 % (fc = 4 to 20 MHz)	
Built-in ROM			64 Kbyte Mask ROM (8000H to 17FFFH)	64 Kbyte OTP (8000H to 17FFFH)	64 Kbyte Mask ROM (8000H to 17FFFH)	64 Kbyte OTP (8000H to 17FFFH)
Built-in RAM			2 Kbyte (0080H to 087FH)			
Dual clock			Available		None	
Clock gear			Available		Available	
I/O ports			79 pins		80 pins ($\overline{\text{WDTOUT}}$→P30)	
Interruption			9 CPU interrupts, 14 internal interrupts, 6 external interrupts		9 CPU interrupts, 12 internal interrupts, 8 external interrupts	
External data bus select pin (AM8/16)			Available		None (8 bit bus selected by CS/WAIT controller)	
Watchdog timer output pin			Available		None	
Port P55, P56, P57			AIN5 to 7		RXD0, $\overline{\text{CTS0}}$, RXD1	
P70, P81, P85, P87 Pull-up resistor			Available		None	
CS0 mapping area (B0CS<B0C1,0> = "00")			7F00H to 7FFFH		2000H to 3FFFH	
CS1 mapping area (B1CS<B1C1,0> = "00")			0880H to 7FFFH		4000H to 5FFFH	
CS2 mapping area (B2CS<B2C1,0> = "00")			8000H to		6000H to 7FFFH	
Pattern generator			4 bits, 2 channels		None	
10 bit AD converter			8 channels		5 channels	
General serial channel			UART, I/O Interface mode select 2ch.		UART, I/O Interface mode select 1ch. UART 1ch.	
External interrupt pins			Positive logic (rising edge)		Negative logic (falling edge)	
	Input level	INT0	Schmitt trigger		V_{IH} = 2.8 [V] MIN V_{IL} = 0.8 [V] MAX	
		INT5	CMOS			
		INT7	CMOS			
		INT8	None			
		INT9	None			