

**CMOS 16-bit Microcontrollers****TMP95C061F****1. Outline and Device Characteristics**

TMP95C061F is a high-speed advanced 16-bit microcontroller developed for controlling medium to large-scale equipment. The TMP95C061F is housed in an 100-pin flat package.

Device characteristics are as follows:

## (1) Original 16-bit CPU

- TLCS-90/900 instruction mnemonic upward compatible.
- 16M-byte linear address space
- General-purpose registers and register bank system
- 16-bit multiplication/division and bit transfer/arithmetic instructions
- High-speed DMA
  - 4 channels (640ns/2 bytes at 25MHz)
- (2) Minimum instruction execution time
  - 160ns at 25MHz
- (3) Internal RAM: None
- Internal ROM: None

## (4) External memory expansion

- Can be expanded up to 16M bytes (for both programs and data)
- AM8/ $\overline{16}$  pin (select external data bus width)
- Can mix 8- and 16-bit external data bus width.
  - ...Dynamic data bus sizing
- (5) DRAM Controller
- (6) 8-bit timer: 2 channels
- (7) 16-bit timer: 2 channels
- (8) Pattern generators: 4 bits, 2 channels
- (9) Serial interface: 2 channels
- (10) 10-bit A/D converter: 4 channels
- (11) Watchdog timer
- (12) Chip select/wait controller: 4 blocks
- (13) Interrupt functions
  - 2 CPU interrupts ... SWI instruction and illegal instruction
  - 18 internal interrupts
  - 6 external interrupts
  - 7-level priority can be set.
- (14) I/O ports: 56 pins
- (15) Standby function : 3 HALT modes (RUN, IDLE, STOP)

The information contained here is subject to change without notice.

The information contained herein is presented only as guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others. These TOSHIBA products are intended for usage in general electronic equipments (office equipment, communication equipment, measuring equipment, domestic electrification, etc.) Please make sure that you consult with us before you use these TOSHIBA products in equipments which require high quality and/or reliability, and in equipments which could have major impact to the welfare of human life (atomic energy control, spaceship, traffic signal, combustion control, all types of safety devices, etc.). TOSHIBA cannot accept liability to any damage which may occur in case these TOSHIBA products were used in the mentioned equipments without prior consultation with TOSHIBA.

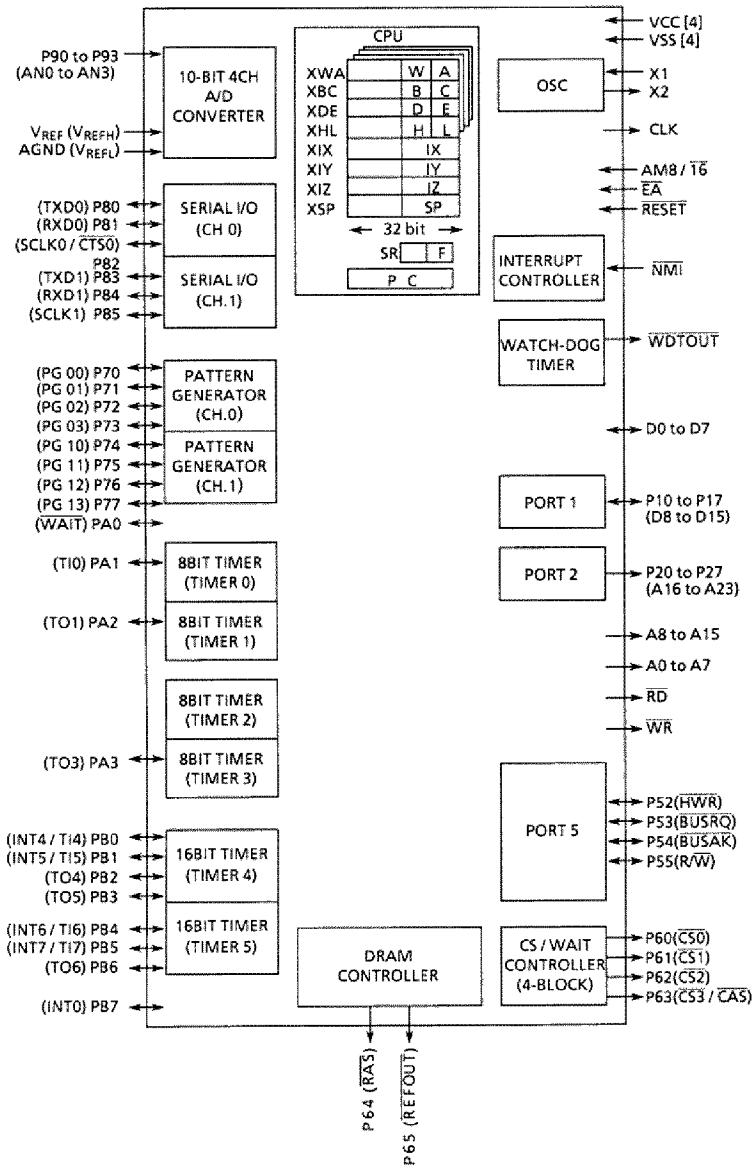


Figure 1. TMP95C061 Block Diagram

## 2. Pin Assignment and Functions

The assignment of input/output pins for TMP95C061, their name and outline functions are described below.

### 2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP95C061.

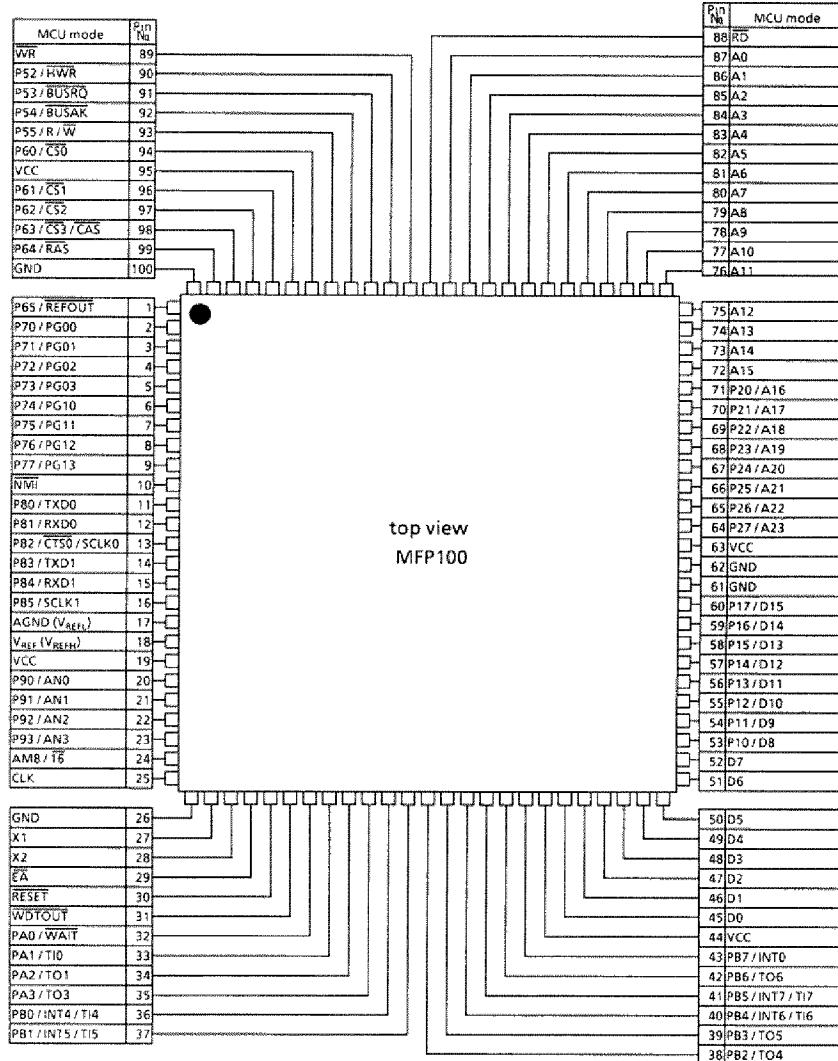


Figure 2.1. Pin Assignment (100-pin QFP)

## 2.2 Pin Names and Functions

The names of input/output pins and their functions are

**Table 2.2. Pin Names and Functions**

Pin name	Number of pins	I/O	Functions
D0 to D7	8	I/O	Data : 0 to 7 for data bus
P10 to P17 D8 to D15	8	I/O I/O	Port 1 : I/O ports that allow I/O to be selected on a bit basis Data : 8 to 15 for data bus
P20 to P27 A16 to A23	8	Output Output	Port 2 : Output ports Address : 16 to 23 for address bus
A8 to A15	8	Output	Address : 8 to 15 for address bus
A0 to A7	8	Output	Address : 0 to 7 for address bus
RD	1	Output	Read : Strobe signal for reading external memory
WR	1	Output	Write : Strobe signal for writing data on pins D0 to 7
P52 HWR	1	I/O Output	Port 52 : I/O port (with pull-up resistor) High Write : Strobe signal for writing data on pins D8 to 15
P53 BUSRQ	1	I/O Input	Port 53 : I/O port (with pull-up resistor) Bus request : Signal used to request high impedance for D0 to 15, A0 to 23, RD, WR, HWR, R/W, CS0 to CS3, RAS, CAS and REFOUT (+) pins. (for external DMAC)
P54 BUSAK	1	I/O Output	Port 54 : I/O port (with pull-up resistor) Bus Acknowledge : Signal indicating that D0 to 15, A0 to 23, RD, WR, HWR, R/W, CS0 to CS3, RAS, CAS and REFOUT (+) pins are at high impedance after receiving BUSRQ. (for external DMAC)
P55 R/W	1	I/O Output	Port 55 : Output port (with pull-up resistor) Read/Write : 1 : indicates read or dummy cycle 0 : indicates write cycle
P60 CS0	1	Output Output	Port 60 : Output port Chip Select 0 : Outputs 0 when address is within specified address area
P61 CS1	1	Output Output	Port 61 : Output port Chip Select 1 : Output 0 when address is within specified address area

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

(\*) RAS, CAS, and REFOUT are set to high impedance only when bus release mode is set using the DRAM controller. For details, see 3.7, Dynamic RAM (DRAM) Controller.

Pin name	Number of pins	I/O	Functions
P62 <u>CS2</u>	1	Output Output	Port 62 : Output port Chip Select 2 : Outputs 0 if address is within specified address area
P63 <u>CS3</u> <u>CAS</u>	1	Output Output	Port 63 : Output port Chip Select 3 : Outputs 0 if address is within specified address area Column address strobe : Outputs <u>CAS</u> strobe for DRAM if address is within specified address area
P64 <u>RAS</u>	1	Output Output	Port 64 : Output port Low address strobe : Output <u>RAS</u> strobe for DRAM if address is within specified address area
P65 <u>REFOUT</u>	1	Output Output	Port 65 : Output port Refresh Output : 0 : indicates period of refresh cycle
P70 to P73 <u>PG00 to PG03</u>	4	I/O Output	Port 70 to 73 : I/O port that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator Port : 00 to 03
P74 to P77 <u>PG10 to PG13</u>	4	I/O Output	Port 74 to 77 : I/O port that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator Port : 10 to 13
P80 <u>TXD0</u>	1	I/O Output	Port 80 : I/O port (with pull-up resistor) Serial send data 0
P81 <u>RXD0</u>	1	I/O Input	Port 81 : I/O port (with pull-up resistor) Serial receive data 0
P82 <u>CTS0</u> <u>SCLK0</u>	1	I/O Input I/O	Port 82 : I/O port (with pull-up resistor) Serial data send enable (clear to send) Serial Clock I/O 0
P83 <u>TXD1</u>	1	I/O Output	Port 83 : I/O port (with pull-up resistor) Serial send data 1
P84 <u>RXD1</u>	1	I/O Input	Port 84 : I/O port (with pull-up resistor) Serial receive data 1
P85 <u>SCLK1</u>	1	I/O I/O	Port 85 : I/O port (with pull-up resistor) Serial clock I/O 1
P90 to P93 <u>ANO to AN3</u>	4	Input Input	Port 9 : Input port Analog input : Input to A/D converter
PA0 <u>WAIT</u>	1	I/O Input	Port A0 : I/O port (with pull-up resistor) Wait : Pin used to request CPU us wait
PA1 <u>T10</u>	1	I/O Input	Port A1 : I/O port (with pull-up resistor) Timer input 0 : Timer 0 input
PA2 <u>TO1</u>	1	I/O Output	Port A2 : I/O port (with pull-up resistor) Timer output 1 : Timer 0 or 1 output

Pin name	Number of pins	I/O	Functions
PA3 TO3	1	I/O Output	Port A3 : I/O port (with pull-up resistor) Timer output3 : 8-bit timer 3 output
PB0 TI4 INT4	1	I/O Input Input	Port B0 : 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
PB1 TI5 INT5	1	I/O Input Input	Port B6 : I/O port (with pull-up resistor) Timer input 5 : Timer 4 count / capture trigger signal input Interrupt request pin 5 : Interrupt request pin with rising edge
PB2 TO4	1	I/O Output	Port B2 : I/O port (with pull-up resistor) Timer output4 : Timer4 output
PB3 TO5	1	I/O Output	Port B3 : I/O port (with pull-up resistor) Timer output5 : Timer4 output
PB4 TI6 INT6	1	I/O Input	Port B4 : 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
PB5 TI7 INT7	1	I/O Input Input	Port B5 : I/O port (with pull-up resistor) Timer input 7 : Timer 5 count / capture trigger signal input Interrupt request pin 7 : Interrupt request pin with rising edge
PB6 TO6	1	I/O Output	Port B6 : I/O port (with pull-up resistor) Timer output6 : Timer5 output pin
PB7 INT0	1	I/O Input	Port B7 : I/O port (with pull-up resistor) Interrupt request pin 0 : Interrupt request pin with programmable level / rising edge
V <sub>REF</sub> (V <sub>REFH</sub> )	1	Input	Pin for reference voltage input to A/D connector
AGND (V <sub>REFL</sub> )	7	Input	Ground pin for A/D converter
WDTOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin : Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output : Outputs [ System Clock + 4 ] Clock. Pulled-up during reset.
EA	1	Input	fixed GND
AM8/16	1	Input	Address mode : Selects external Data Bus width "0" should be inputted with fixed 16 bit Bus width or 16 bit Bus interlocked with 8 bit Bus. "1" should be inputted with fixed 8 bit Bus width
RESET	1	Input	Reset : Initializes LSI (with pull-up resistor)
X1 / X2	2	I/O	Oscillator connecting pin
VCC	4		Power supply pin (+ 5 V)
VSS	4		GND pin (0 V)

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

### 3. Operation

This section describes in blocks the functions and basic operations of TMP95C061 device.

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

#### 3.1 CPU

TMP95C061 device has a built-in high-performance 16-bit CPU (900/H CPU). (For CPU operation, see TLCS-900 CPU in the previous section.)

This section describes CPU functions unique to TMP95C061 that are not described in the previous section.

##### 3.1.1 Reset

To reset the TMP95C061, the RESET input must be kept at 0 for at least 10 system clocks (10 states: 800ns with a 25MHz system clock) within an operating voltage range and with a stable oscillation.

When reset is accepted, the CPU sets as follows:

- Program Counter (PC) to 8000H.  
 PC (7 : 0) ← stored data to 0FFFF00H  
 PC (15 : 8) ← stored data to 0FFFF01H  
 PC (23 : 16) ← stored data to 0FFFF02H
- Stack pointer (XSP) for system mode to 100H.
- IFF2 to 0 bits of status register to 111. (Sets mask register to interrupt level 7.)
- MAX bit of status register to 1. (Sets to minimum mode.)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

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

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

- Initializes built-in I/O registers as per specifications.
- Sets port pins (including pins also used as built-in I/Os) to general-purpose input/output port mode.
- Sets the WDTOUT pin to 0. (Watchdog timer is set to enable after reset.)
- Pulls up the CLK pin to 1.

##### 3.1.2 External Data Width Selection Pin (AM18/16)

After reset operation, TMP95C061, the operates 8 bits or 16 bits external data width according to input to AM8/16 pin.

- Fixed 16 bit bus or 16 bit bus interlarded with 8 bit bus

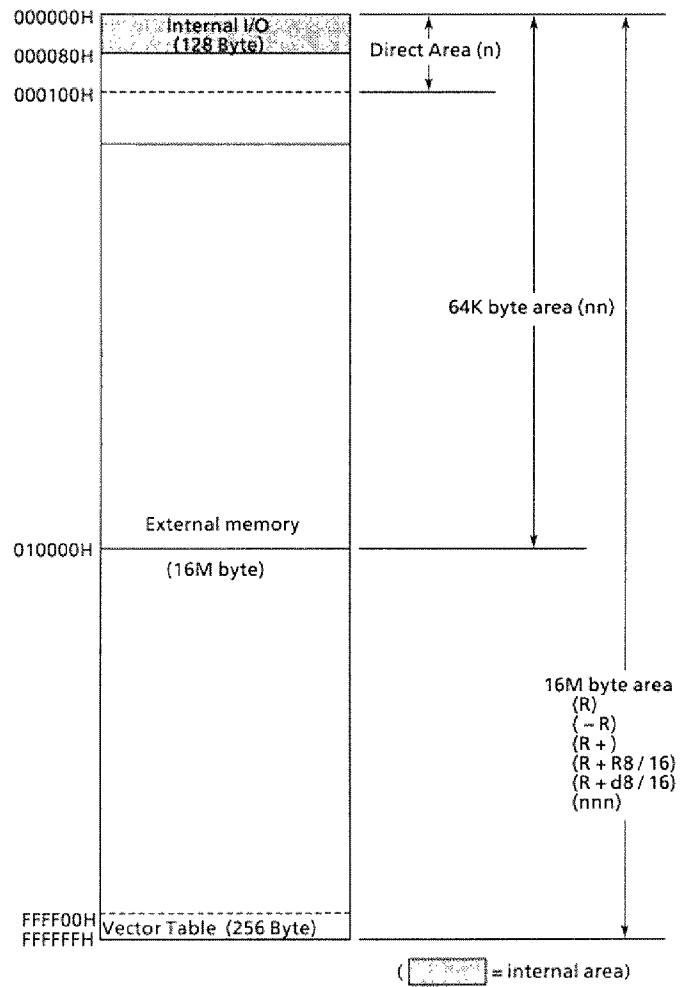
"0" should be input. Port 1 (P10 to P17) operate as data bus D8 to 15. The data bus width for external access is set by Chip Select/Wait Control resistor.

- Fixed 8 bit bus

"1" should be input. Port 1 (P10 to P17) operate as 8 bit data I/O ports. The value set in Chip Select/Wait Control resistor <B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> and <BEXBUS> are neglected.

### 3.2 Memory Map

Figure 3.2 shows a memory map of the TMP95C061.



(Note) After reset operation, Stack point (XSP) is set to 100H.

**Figure 3.2. Memory Map**

### 3.3 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask

flip-flop (IFF2 to 0) and the built-in interrupt controller.

TMP95C061F has the following 26 interrupt sources:

- Interrupts from the CPU···2  
(Software interrupts, privileged violations, and Illegal (undefined) instruction execution)
- Interrupts from external pins (NMI, INT0, and INT4 to 7)···6
- Interrupts from built-in I/Os···14
- Interrupts from high-speed DMA (HDMA)···4

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

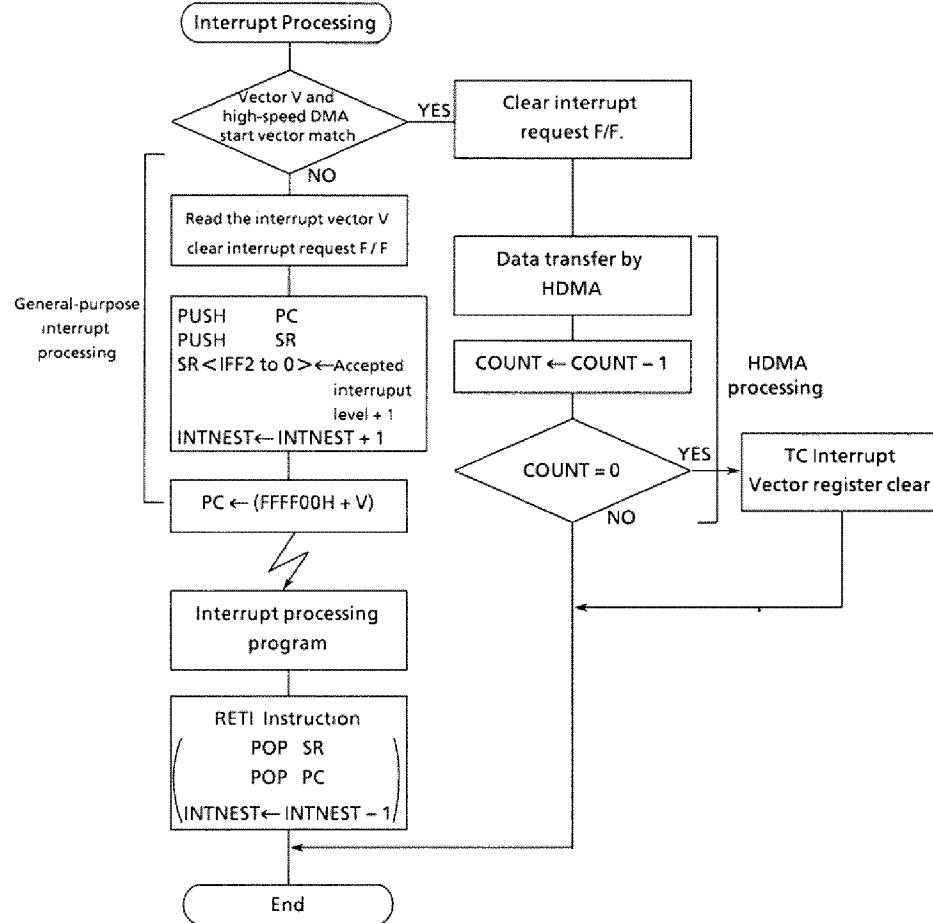
When an interrupt is generated, the interrupt controller sends the value of the priority of the interrupt source to the CPU. When more than one interrupt is generated simultaneously, the interrupt controller sends the value of the highest priority (7 for non-maskable interrupts is the highest) to the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

The CPU compares the value of the priority sent with the value in the CPU interrupt mask register (IFF2 to 0). If the value is greater than that of the CPU interrupt mask register, the interrupt is accepted. The value in the CPU interrupt mask register (IFF2 to 0) can be changed using the EI instruction (contents of the EI num/IFF<2:0> = num). For example, programming EI 3 enables acceptance of maskable interrupts

with a priority of 3 or greater, and non-maskable interrupts which are set in the interrupt controller. The DI instruction (IFF<2:0> = 7) operates in the same way as the EI 7 instruction. Since the priority values for maskable interrupts are 0 to 6, the DI instruction is used to disable maskable interrupts to be accepted. The EI instruction becomes effective immediately after execution. (With the TLCS-90, the EI instruction becomes effective after execution of the subsequent instruction.)

In addition to the general-purpose interrupt processing mode described above, there is also a high-speed DMA (HDMA) processing mode. HDMA is a mode used by the CPU to automatically transfer byte, word and 4-byte data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

Figure 3.3 (1) is a flowchart showing overall interrupt processing.



Note : In case of read-only mode, always branches to "NO" without conditional branch.

**Figure 3.3 (1). Interrupt Processing Flowchart**

### 3.3.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows:

- (1) The CPU reads the interrupt vector from the interrupt controller. When more than one interrupt with the same level is generated simultaneously, the interrupt controller generates interrupt vectors in accordance with the default priority (which is fixed as follows: the smaller the vector value, the higher the priority), then clears the interrupt request.
- (2) The CPU pushes the program counter and the status register to the system stack area (area indicated by the system mode stack pointer(XSP)).
- (3) The CPU sets a value in the CPU interrupt mask register <IFF2 to 0> that is higher by 1 than the value of the accepted interrupt level. However, if the value is 7, 7 is set without an increment.
- (4) The CPU increments the INTNEST (Interrupt Nesting Counter).
- (5) The CPU jumps to address FFFF00H + interrupt vector, then starts the interrupt processing routine.

Bus Width of Stack Area	Bus Width Interrupt Vector Area	Interrupt Processing State Number	
		MAX mode	Min mode
8 bit	8 bit	23	24
	16 bit	24	20
16 bit	8 bit	22	20
	16 bit	18	16

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

If an interrupt generated while the CPU is performing processes (1) to (5) for an earlier interrupt, the new interrupt is sampled immediately after the start instruction of the interrupt processing is executed. Setting DI as the start instruction disables maskable interrupt nesting. (Note: With the 900 and 900/L, an interrupt is sampled before the start instruction is executed.)

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

The addresses 0FFF00H to 0FFFFFFH (256 bytes) of the TMP95C061 are assigned for interrupt processing entry area.

**Table 3.3 (1) TMP95C061 Interrupt Table**

Default priority	Type	Interrupt source	Vector value "V"	Address refer to vector	HDMA start vector
1	Non-maskable	Reset, or SWI0 instruction	0 0 0 0 H	FFFF00H	-
2		SWI1 instruction	0 0 0 4 H	FFFF04H	-
3		INTUNDEF : Illegal instruction, or SWI2	0 0 0 8 H	FFFF08H	-
4		SWI3 instruction	0 0 0 C H	FFFF0CH	-
5		SWI4 instruction	0 0 1 0 H	FFFF10H	-
6		SWI5 instruction	0 0 1 4 H	FFFF14H	-
7		SWI6 instruction	0 0 1 8 H	FFFF18H	-
8		SWI7 instruction	0 0 1 C H	FFFF1CH	-
9		NMI Pin	0 0 2 0 H	FFFF20H	-
10		INTWD : Watchdog timer	0 0 2 4 H	FFFF24H	-
-	Maskable	(HDMA)	-	-	-
11		INT0 pin	0 0 2 8 H	FFFF28H	0AH
12		INT4 pin	0 0 2 C H	FFFF2CH	0BH
13		INT5 pin	0 0 3 0 H	FFFF30H	0CH
14		INT6 pin	0 0 3 4 H	FFFF34H	0DH
15		INT7 pin	0 0 3 8 H	FFFF38H	0EH
-		(Reserved)	0 0 3 C H	FFFF3CH	-
16		INTT0 : 8-bit timer0	0 0 4 0 H	FFFF40H	10H
17		INTT1 : 8-bit timer1	0 0 4 4 H	FFFF44H	11H
18		INTT2 : 8-bit timer2	0 0 4 8 H	FFFF48H	12H
19		INTT3 : 8-bit timer3	0 0 4 C H	FFFF4CH	13H
20		INTTR4 : 16-bit timer4 (TREG4)	0 0 5 0 H	FFFF50H	14H
21		INTTR5 : 16-bit timer4 (TREG5)	0 0 5 4 H	FFFF54H	15H
22		INTTR6 : 16-bit timer5 (TREG6)	0 0 5 8 H	FFFF58H	16H
23		INTTR7 : 16-bit timer5 (TREG7)	0 0 5 C H	FFFF5CH	17H
24		INTRX0 : Serial receive (Channel.0)	0 0 6 0 H	FFFF60H	18H
25		INTTX0 : Serial send (Channel.0)	0 0 6 4 H	FFFF64H	19H
26		INTRX1 : Serial receive (Channel.1)	0 0 6 8 H	FFFF68H	1AH
27		INTTX1 : Serial send (Channel.1)	0 0 6 C H	FFFF6CH	1BH
28		INTAD : A/D conversion completion	0 0 7 0 H	FFFF70H	1CH
29		INTTC0 HDMA completion (channel.0)	0 0 7 4 H	FFFF74H	-
30		INTTC1 HDMA completion (channel.1)	0 0 7 8 H	FFFF78H	-
31		INTTC2 HDMA completion (channel.2)	0 0 7 C H	FFFF7CH	-
32		INTTC3 HDMA completion (channel.3)	0 0 8 0 H	FFFF80H	-
-		(Reserved)	0 0 8 4 H	FFFF84H	-
to		to	to	to	to
-		(Reserved)	0 0 F C H	FFFFFC	-

## Setting to Reset/Interrupt Vector

### ① Reset Vector

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

### ② Interrupt Vector (except Reset Vector)

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

XX : don't care

### (Setting Example)

Reset Vectir : 8100H, NMI Vector : 9ABCH, INTAD Vector : 123456H.

(The value '99H' as a vector is for the explaining  
and it does not have a special meaning.)

ORG	8100H	
LD	A, B	(cf)
		ORG, DL are the Assembler Directive.
ORG	9ABCH	ORG : control location counter
LD	B, C	DL : define the long word (32 bits) data
ORG	123456H	
LD	C, A	
ORG	0FFFF00H	
DL	<u>008100H</u>	; Reset = 8100H
ORG	0FFFF20H	
DL	<u>009ABCH</u>	; NMI = 9ABCH
ORG	0FFFF70H	
DL	<u>123456H</u>	; INTAD = 123456H

### 3.3.2 High-Speed DMA (HDMA)

In addition to the conventional interrupt processing, TMP95C061 also has a high-speed DMA (HDMA) function. Each interrupt request starts HDMA operation in level "6" irrelevant to the set interrupt level. Level "6" is the interrupt level which has the highest priority among maskable interrupts.

#### (1) HDMA Operation

When the interrupt is generated in the interrupt request source set by HDMA start vector resistors, the interrupt controller sends the HDMA request to the CPU in level "6" irrelevant to the set interrupt level.

The HDMA has four channels so that it can be set up for up to four types of interrupt source.

When an HDMA 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 not 0, HDMA processing is completed; if the value in the counter after decrementing is 0, the CPU notifies the interrupt controller of the HDMA transfer end interrupt (INTTCn), zero-clears the HDMA start vector register, disables re-start of the HDMA, and ends the HDMA processing.

The priority of the HDMA transfer end interrupt generated at this time is determined by its interrupt level and default priority, same as with other maskable interrupts.

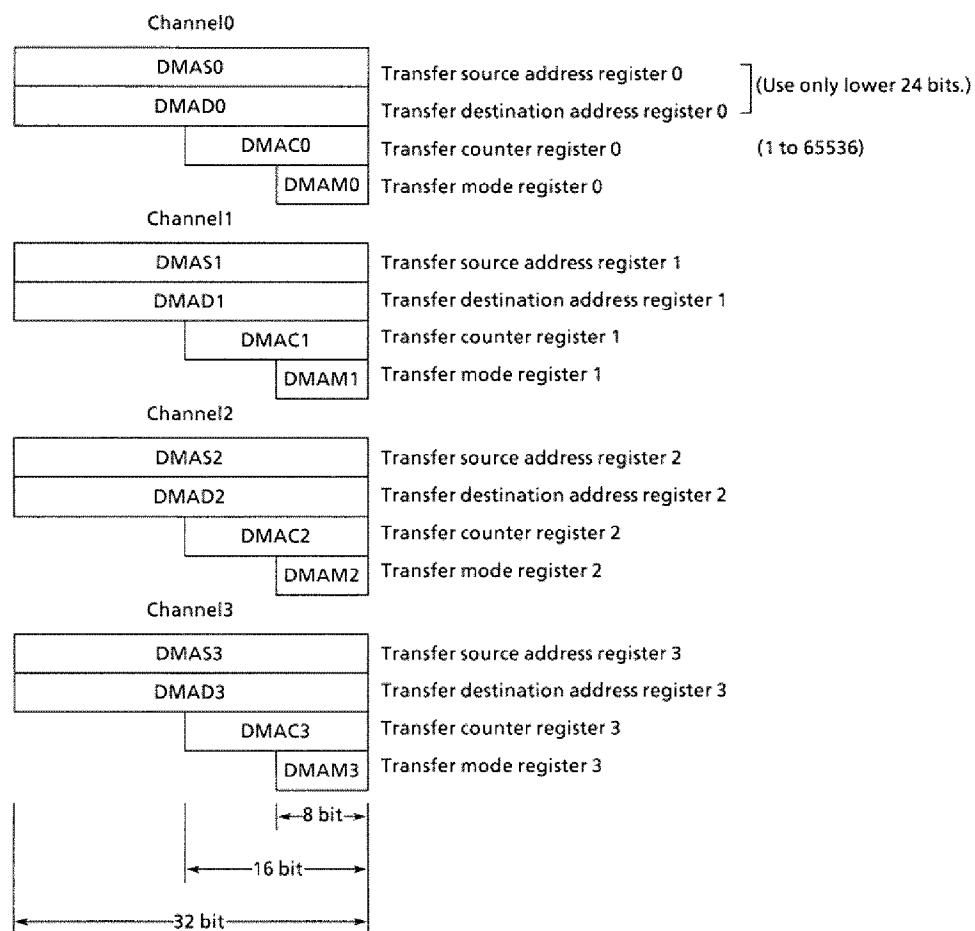
The priorities of HDMA requests generated simultaneously in multiple channels are irrelevant to their input levels. The HDMA request which is generated in the channel with the small number has a higher priority. (CH0: highest priority; CH3: lowest priority).

The 32-bit control registers are used for setting transfer source/destination addresses. However, the TLCS-900 has only 24 address pins for output. A 16M-byte space is available for the high-speed HDMA. There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes, see the description of transfer mode registers.

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

Interrupt sources processed by HDMA processing are those with the high-speed HDMA start vectors listed in Table 3.3 (1).

## (2) Register Configuration (CPU Control Register)



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

## (3) Transfer Mode Register Details

(DMAM0 to 3)

0 0 0 Z Z	Transfer destination address INC mode ... for I/O to memory (DMADn +) ← (DMASn) DMACn←DMACn - 1 if DMACn = 0 then INTTC.	8 states (640 ns) @ Byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 0 1 Z Z	Transfer destination address DEC mode ... for I/O to memory (DMADn -) ← (DMASn) DMACn←DMACn - 1 if DMACn = 0 then INTTC.	8 states (640 ns) @ Byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 1 0 Z Z	Transfer source address INC mode ... for memory to I/O (DMADn) ← (DMASn +) DMACn←DMACn - 1 if DMACn = 0 then INTTC.	8 states (640 ns) @ Byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 1 1 Z Z	Transfer source address DEC mode ... for memory to I/O (DMADn) ← (DMASn -) DMACn←DMACn - 1 if DMACn = 0 then INTTC.	8 states (640 ns) @ Byte / word transfer 12 states (960 ns) @ 4-byte transfer
1 0 0 Z Z	Fixed address mode ... I/O to I/O (DMADn) ← (DMASn) DMACn←DMACn - 1 if DMACn = 0 then INTTC.	8 states (640 ns) @ Byte / word transfer 12 states (960 ns) @ 4-byte transfer
1 0 1 0 0	Counter mode ... for interrupt counter DMASn←DMASn + 1 DMACn←DMACn - 1 if DMACn = 0 then INTTC.	5 states (400 ns)

(1 states = 80 ns at 25 MHz)

Note: n : corresponds to  $\mu$ HDMA channels 0 to 3.

DMADn + / DMASn + : Post-increment (Increments register value after transfer.)

DMADn - / DMASn - : Post-decrement (Decrements register value after transfer.)

"I/O" means the fixed address, "memory" means the increased or decreased address in this table.

Do not use undefined codes for transfer mode control.

### 3.3.3. Interrupt Controller

Figure 3.3.3 (1) is a block diagram of the interrupt circuits. The left half of the diagram shows the interrupt controller; the right half includes the CPU interrupt request signal circuit and the HALT release signal circuit.

Each interrupt channel (total of 24 channels) in the interrupt controller has an interrupt request flip-flop, interrupt priority setting register, and a register for storing the high-speed micro DMA start vector. The interrupt request flip-flop is used to latch interrupt requests from peripheral devices. The flip-flop is cleared to 0 at reset, when the CPU reads the interrupt channel vector after the acceptance of interrupt, or when the CPU executes an instruction that clears the interrupt of that channel (writes 0 in the clear bit of the interrupt priority setting register).

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

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

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

The interrupt priority can be set by writing the priority in the interrupt priority setting register (e.g., INTE0AD, INTE45, etc.) provided for each interrupt source. Interrupt levels to be set are from 1 to 6. Writing 0 or 7 as the interrupt priority dis-

ables the corresponding interrupt request. The priority of the non-maskable interrupt (NMI pin, watchdog timer, etc.) is fixed to 7. If interrupt requests with the same interrupt level are generated simultaneously, interrupts are accepted in accordance with the default priority (the smaller the vector value, the higher the priority).

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2 to 0> set in the Status Register by the interrupt request signal with the priority value sent; if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 in the CPU SR <IFF2 to 0>. Interrupt requests where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine. When interrupt processing is completed (after execution of the RETI instruction), the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR <IFF2 to 0>.

The interrupt controller also has four registers used to store the HDMA start vector. These are I/O registers; unlike other HDMA registers (DMAS, DMAD, DMAM, and DMAC), they can be accessed in either normal or system mode. Writing the start vector of the interrupt source for the HDMA processing (see Table 3.3 (1)), enables the corresponding interrupt to be processed by micro HDMA processing. The values must be set in the HDMA parameter registers (e.g., DMAS and DMAD) prior to the micro HDMA processing.

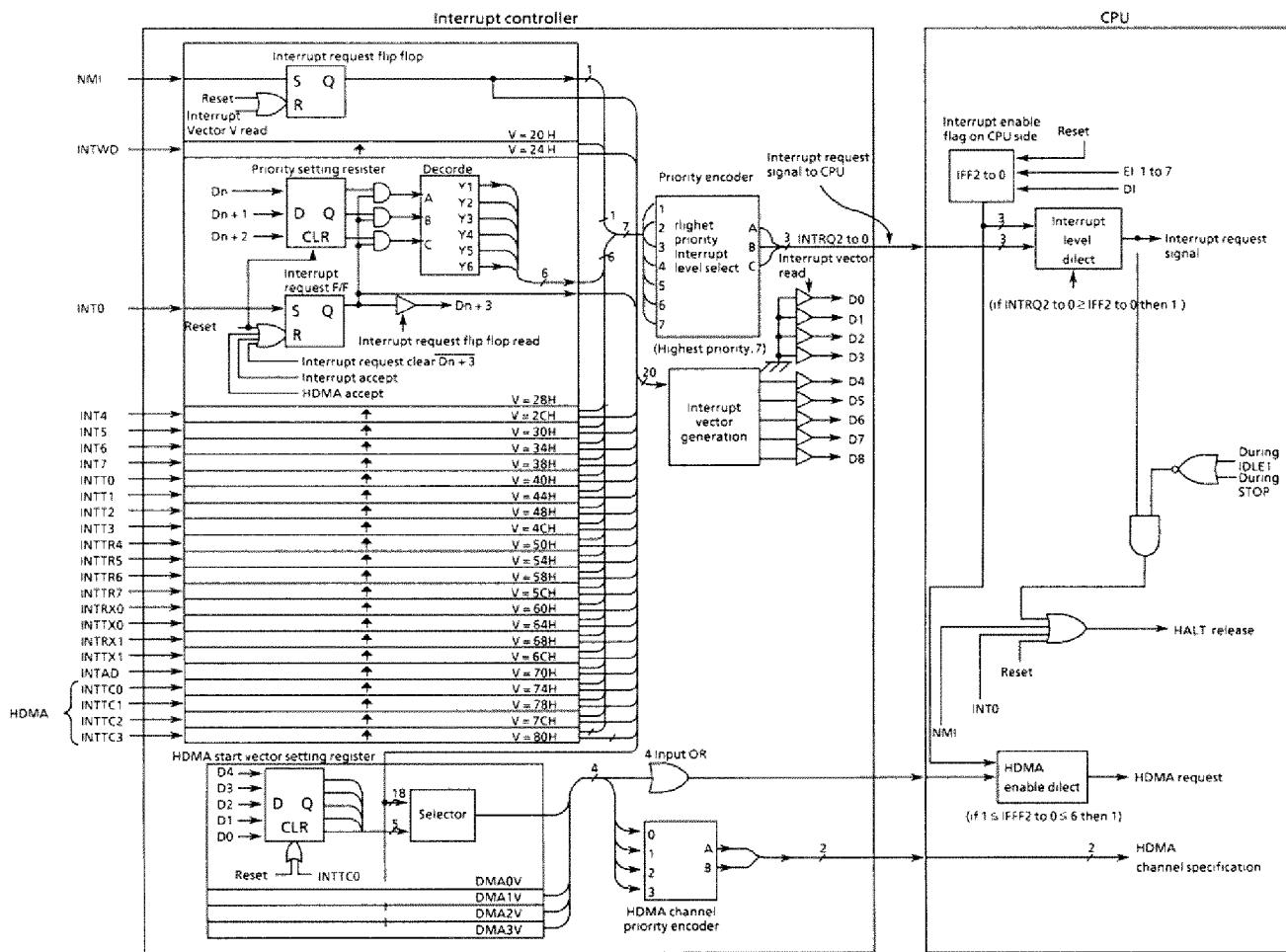


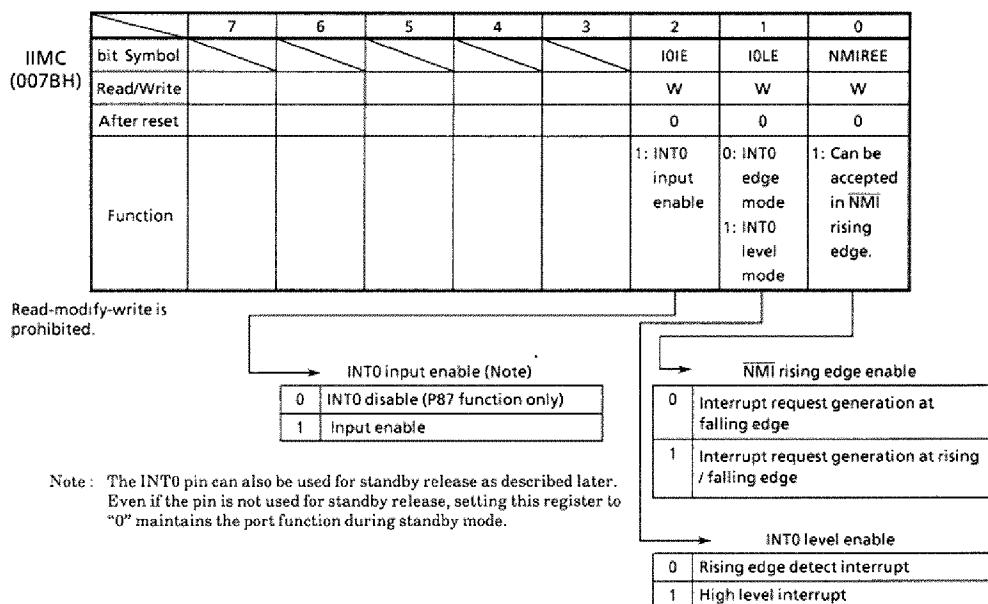
Figure 3.3.3 (1). Block Diagram of Interrupt Controller

## (1) Interrupt Priority Setting Register

Symbol	Address	7	6	5	4	3	2	1	0
		INTAD					INT0		
INTE0AD	0070H	IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INT5					INT4		
INTE45	0071H	I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INT7					INT6		
INTE67	0072H	I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTT1 (Timer 1)					INTT0 (Timer 0)		
INTET01	0073H	IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTT3 (Timer 3)					INTT2 (Timer 2)		
INTET23	0074H	IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTR5 (TREG5)					INTTR4 (TREG4)		
INTET45	0075H	IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTR7 (TREG7)					INTTR6 (TREG6)		
INTET67	0076H	IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTX0					INTRX0		
INTES0	0077H	ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTX1					INTRX1		
INTES1	0078H	ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTC1					INTTC0		
INTETC01	0079H	ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		INTTC3					INTTC2		
INTETC23	007AH	ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
		1	1	1	1				
		IxxM2			IxxM1	IxxM0	Function (Write)		
		0	0	0	1	0	Prohibits interrupt request		
		0	0	1	0	0	Sets interrupt request level to "1"		
		0	1	0	1	0	Sets interrupt request level to "2"		
		0	1	1	0	0	Sets interrupt request level to "3"		
		1	0	0	0	0	Sets interrupt request level to "4"		
		1	0	0	1	0	Sets interrupt request level to "5"		
		1	1	0	0	0	Sets interrupt request level to "6"		
		1	1	1	1	1	Prohibits interrupt request		
		IxxC			Function (Read)		Function (Write)		
		0	Indicates no interrupt request.		Clears interrupt request flag.		-----Don't care-----		
		1	Indicates interrupt request.						

←Interrupt source  
←Bit Symbol  
←Read / Write  
←After reset

## (2) External Interrupt Control



## Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
NMI	—	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	PB7	Rising edge	IIMC<IOLE> = 0, <IOIE> = 1
		Level	IIMC<IOLE> = 1, <IOIE> = 1
INT4	PB0	Rising edge	T4MOC<CAP12M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T4MOD<CAP12M1,0> = 1,0
INT5	PB1	Rising edge	_____
INT6	PB4	Rising edge	T5MOC<CAP34M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T5MOD<CAP34M1,0> = 1,0
INT7	PB5	Rising edge	_____

## (3) HDMA Start Vector

Register used to assign HDMA processing to an interrupt source. The interrupt source whose HDMA start vector matches the vector value set in this register is assigned as the HDMA start source.

When the HMDA transfer counter value reaches 0, the controller is notified of the HDMA transfer end interrupt corresponding to the channel, the HDMA start vector is cleared, and the HDMA start source of the channel is also cleared. To continue the HDMA processing, the HDMA start vector register must be set again within

the HDMA transfer end interrupt processing.

If the same vector is set in the HDMA start vector registers of the multiple channels, the interrupt generated in the channel with the smaller number has a higher priority.

Thus, if the same vector is set in the HDMA start vector registers of two channels, the interrupt generated in the channel with the smaller number is processed until the HDMA transfer end. If a HDMA start vector is not set, then the HDMA processing is started for the channel with the larger number is processed.

HDMA0 Start Vector (read-modify-write is not possible)								
DMA0V (007CH)	7	6	5	4	3	2	1	0
	bit Symbol			DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
	Read/Write						W	
	After reset			0	0	0	0	0

HDMA1 Start Vector (read-modify-write is not possible.)								
DMA1V (007DH)	7	6	5	4	3	2	1	0
	bit Symbol			DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
	Read/Write						W	
	After reset			0	0	0	0	0

HDMA2 Start Vector (read-modify-write is not possible.)								
DMA2V (007EH)	7	6	5	4	3	2	1	0
	bit Symbol			DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
	Read/Write						W	
	After reset			0	0	0	0	0

HDMA3 Start Vector (read-modify-write is not possible )								
DMA3V (007FH)	7	6	5	4	3	2	1	0
	bit Symbol			DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
	Read/Write						W	
	After reset			0	0	0	0	0

## (4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently of each other. Therefore, if the instruction used to clear an interrupt request flag of an interrupt is fetched before the interrupt is generated, it is possible that the CPU might execute the fetched instruction to clear the interrupt

request flag while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector "0028H" and start the interrupt processing from the address "FFFF28H".

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

INT0 Level mode	<p>IF INT0 is not an edge-based interrupt, the function of Interrupt Request Flip-flop is canceled. Therefore the interrupt request signal must be held until the interrupt request is acknowledged by the CPU. A change in the mode (edge to level) automatically clears the interrupt request flag.</p> <p>When the CPU has been put in the interrupt response sequence with INT0 level mode, it is necessary to leave INT0 at "1" until the interrupt response sequence is completed. Also, "1" must always be held until HALT is cleared when using the INT0 level mode to clear HALT. (Use care to prevent noise changing "1" back to "0".)</p> <p>When switching from the level mode to the edge mode, the interrupt request flag set in the level mode is not cleared; therefore, use the following sequence to clear the interrupt request flag.</p> <p>DI LD (007BH), 00H: switch from level to edge LD (0070H), 00H: clear interrupt request flag EI</p>
INTAD level mode	The Interrupt Request Flip-flop can be cleared only by resetting or reading the register that stores A/D conversion value, and cannot be cleared by an instruction.
INTRX level mode	The Interrupt Request Flip-flop is cleared only by resetting or reading the serial channel receiving buffer, and not by an instruction.

### 3.4 Standby Controller

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

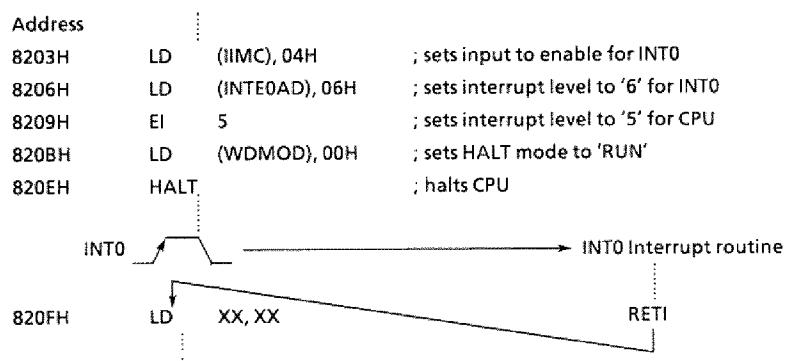
- (1) RUN : Only the CPU halts; power consumption remains unchanged.
- (2) IDLE : Only the built-in oscillator operates, while all other built-in circuits stop. The power consumption is reduced to 1/10 or less than that during NORMAL operation.

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

The HALT release depends on these three modes. For details, see "Table 3.4 (2)". (Note: The HALT state cannot be released by HDMA start.)

(Example releasing "RUN" mode)

INT0 interrupt releases HALT state when the RUN mode is on.



When the halt state is released by a reset, the status in

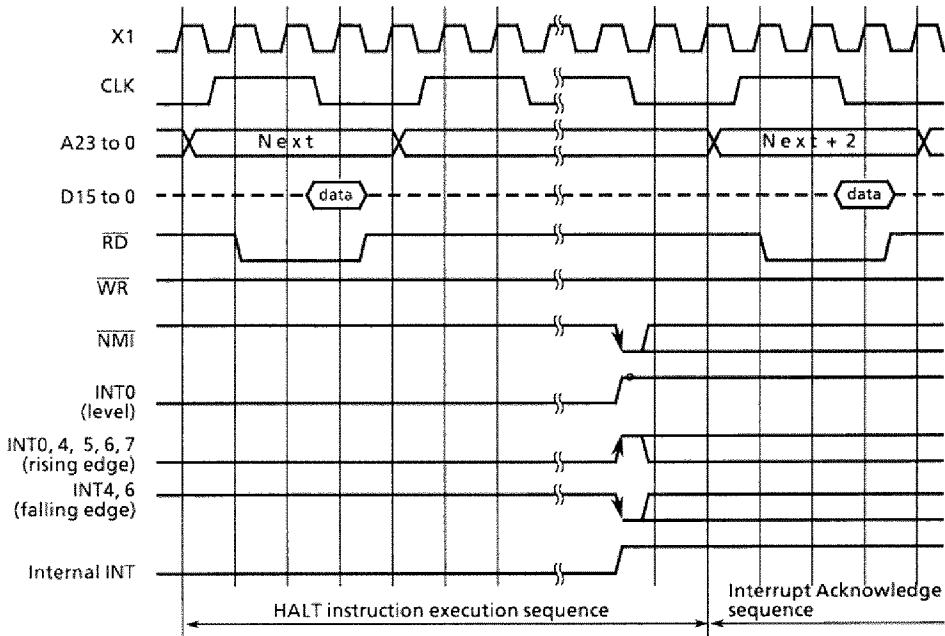
effect before entering the halt status is hold.

## (1) RUN Mode

Figure 3.4.1 shows the timing for releasing the HALT state by interrupts in the RUN mode.

In the RUN mode, the system clock in the MCU continues to operate even after a HALT instruction is exe-

cuted. Only the CPU stops executing the instruction. Until the HALT state is released, the CPU repeats dummy cycles. In the HALT state, an interrupt request is sampled with the rising edge of the "CLK" signal. The external interrupts (INT4, 5, 6, 7) release only RUN mode.



**Figure 3.4.1. Timing Chart for Releasing the HALT State by Interrupt in RUN Modes**

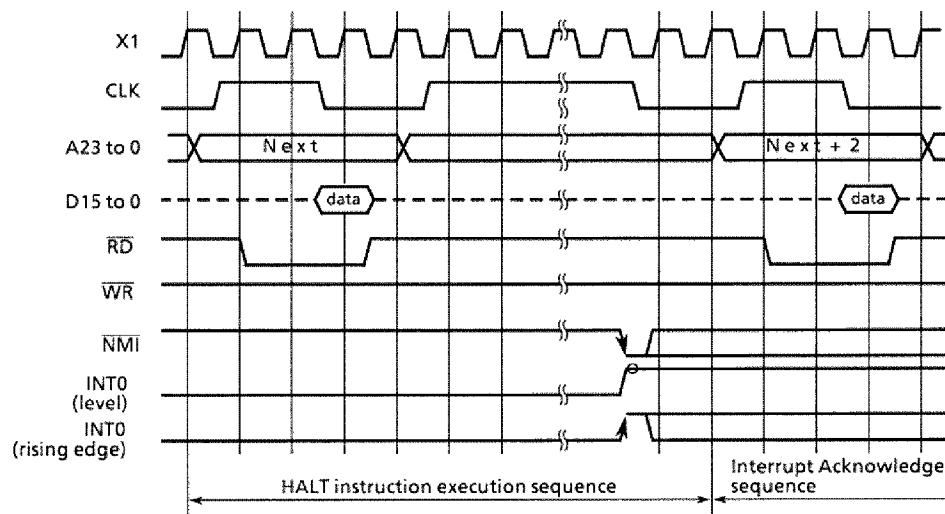
## (2) IDLE Mode

Figure 3.4.2 illustrates the timing for releasing the HALT state by interrupts in the IDLE mode.

In the IDLE mode, only their internal oscillator operates. The system clock in the MCU stops, and the CLK pin is fixed at the "1" level.

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

The interrupts except NMI and INT0 is disable during this mode.



**Figure 3.4.2. Timing Chart for Releasing the HALT State by Interrupts in RUN Mode**

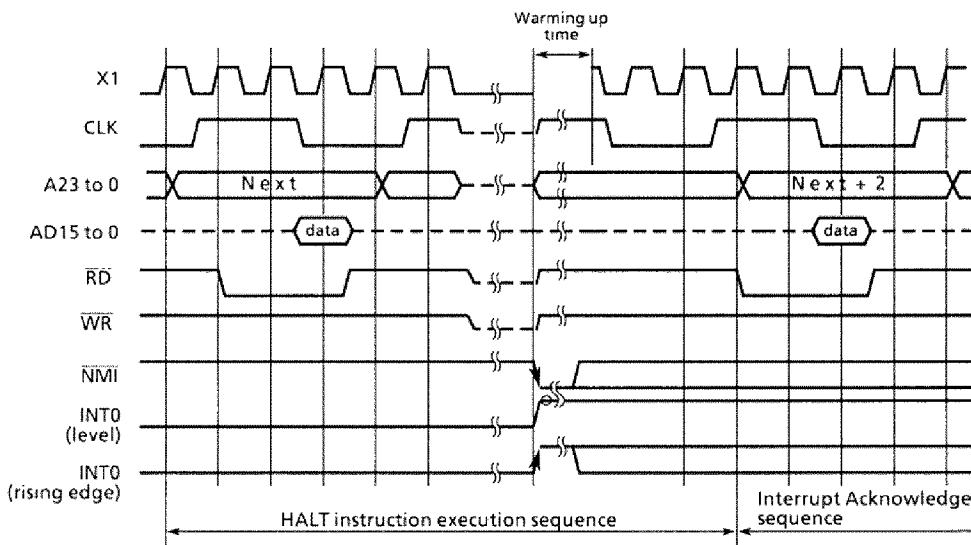
## (3) STOP Mode

Figure 3.4.3 is a timing chart for releasing the HALT state by interrupts in the STOP mode.

The STOP mode is selected to stop all internal circuits including the internal oscillator. In this mode, all pins except special ones are put in the high-impedance state, independent of the internal operation of the MCU. Note, however, that the pre-halt state (the status prior to execution of HALT instruction) of all output pins can be retained by setting the internal I/O register

WDMOD <DRVE> to "1". The content of this register is initialized to "0" by resetting.

When the CPU accepts an interrupt request, the internal oscillator is restarted immediately. However, to get the stabilized oscillation, the system clock starts its output after the time set by the warming up counter WDMOD <WARM>. A warming up time of either the clock oscillation time  $\times 2^{14}$  or  $2^{16}$  can be set by setting this bit to either "0" or "1". This bit is initialized to "0" by resetting.



**Figure 3.4.3. Timing Chart for Released by Interrupt in STOP Mode**

Only either the NMI, INT0, or RESET can release the STOP mode.

When the STOP is released by the except RESET, the system clock is started outputting after warming up time to get the stabilized oscillation.

When the STOP mode is released by RESET, it is necessary to keep the RESET signal at "0" long enough to

release to get the stabilized oscillation because the warming up counter is ignored.

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

**Table 3.4 (1) Pin States in STOP Mode**

Pin name	I/O	DRVE = 0	DRVE = 1
D0 to 7	I/O	HI-Z*	HI-Z*
P10 to P17 (D8 to D15)	Input mode (P10 to P17) Output mode (P10 to P17) I/O (D8 to D15)	HI-Z* HI-Z* HI-Z*	HI-Z* Output HI-Z*
P20 to P27 (A16 to A23)	Output	HI-Z	Output
A0 to A15	Output	HI-Z	Output
RD, WR	Output	HI-Z	"1"
P52 to P55 (HWR, BUSRQ, BUSAK, R/W)	Input mode Output mode	PU*	PU△ Output
P60 to P65 (CS, RAS, CAS, REFOUT)	Output	HI-Z	Output
P70 to P77 (PG00 to PG13)	Input mode Output mode	PU*	PU△ Output
P80 to P85 (TXD, RXD, SCLK, CTS)	Input mode Output mode	PU*	PU△ Output
P90 to P93 (AN0 to AN3)	Input (PORT) Input (AN0 to AN3)	invalid ◎	invalid ◎
PA0 (WAIT)	Input mode Output mode	PU*	PU△ Output
PA1 to PA3 (T10, T01, T03)	Input mode Output mode	PU*	PU△ Output
PB0 to PB6 (T14 to 7, T04 to 6, INT4 to 7)	Input mode Output mode	PU*	PU△ Output
PB7 (INT0)	Input mode Output mode	PU△	PU△ Output
NMI	Input	valid	valid
WDTOUT	Output	Output	Output
CLK	Output	HI-Z	"1"
RESET	Input	valid	valid
AM (8 / 16)	Input	◎	◎
EA	Input	◎	◎
X1	Input	invalid	invalid
X2	Output	"1"	"1"

Output : Output state before HALT state.

PU : Programmable pull-up pin.

\* : Input gate disable state. No through current even if the pin is set to high impedance.

△ : Fix the pin to avoid through current since the input gate operates when the pin is at high impedance.

◎ : need to be driven externally.

valid : Input is valid.

invalid : Input is invalid. No through current since input gate is disable.

Table 3.4 (2) I/O Operation and Cancel During Halt Mode

Halt mode		RUN	IDLE	STOP
WDMOD ( HALTM1,0 )		00	10	01
Operation block	CPU	Stopped		
	I/O port	See Table 3.4 (1)		
	8 bit Timer			
	8 bit PWM Timer			
	16 bit Timer	Operating		Stopped
	Pattern Generator			
	Serial Interface			
	A/D Converter			
	Watch Dog Timer			
DRAM Controller				
Interrupt Controller				

request level		interrupt request level			interrupt request level*2		
		≥ interrupt mask ( IFF2 to 0 )			< interrupt mask ( IFF2 to 0 )		
Halt mode		RUN	IDLE	STOP	RUN	IDLE	STOP
Halt release sources	NMI	◎	◎	◎*1	◎	◎	◎*1
	INTWD	◎	x	x	◎	x	x
	INT0	◎	◎	◎*1	○	○	○*1
	INT4-7	◎	x	x	x	x	x
	INTT0-3	◎	x	x	x	x	x
	INTTR4-7	◎	x	x	x	x	x
	INTRXDO, 1	◎	x	x	x	x	x
	INTTXDO, 1	◎	x	x	x	x	x
	INTAD	◎	x	x	x	x	x
RESET		◎	◎	◎	◎	◎	◎

◎ : Interrupt processing is processed after releasing HALT state. (Reset initializes LSI )

○ : Start executing an instruction that follows the HALT instruction after releasing HALT state.

x : Cannot be used for halt release.

\*1 : Release HALT state after the warming up time

\*2 : The DI instruction operates in the same way.

### 3.5 Functions of Ports

The TMP95C061 has a total of 56 bits when the AM8/16 pin is set to 1; a total of 48 bits when the AM8/16 pin is set to 0.

These ports are also used for internal CPU and I/O. Table 3.5 lists port pin functions.

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

**Table 3.5 Functions of Ports**

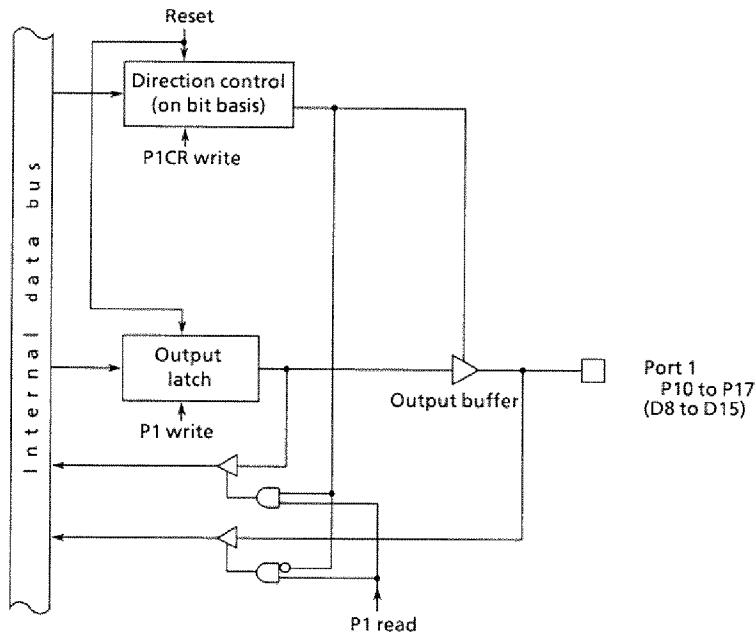
Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port1	P10 to P17	8	I/O	—	Bit	D8 to D15
Port2	P20 to P27	8	Output	—	(Fixed)	A16 to A23
Port5	P50	1	I/O	↑	Bit	HWR
	P53	1	I/O	↑	Bit	BUSRQ
	P54	1	I/O	↑	Bit	BUSAK
	P55	1	I/O	↑	Bit	R/W
Port6	P60	1	Output	—	(Fixed)	CS0
	P61	1	Output	—	(Fixed)	CS1
	P62	1	Output	—	(Fixed)	CS2
	P63	1	Output	—	(Fixed)	CS3/CAS
	P64	1	Output	—	(Fixed)	RAS
	P65	1	Output	—	(Fixed)	REFOUT
Port7	P70 to P77	8	I/O	↑	Bit	PG00 to PG03, PG10 to PG13
Port8	P80	1	I/O	↑	Bit	TXD0
	P81	1	I/O	↑	Bit	RXD0
	P82	1	I/O	↑	Bit	CTS0/SCLK0
	P83	1	I/O	↑	Bit	TCD1
	P84	1	I/O	↑	Bit	RXD1
	P85	1	I/O	↑	Bit	SCLK1
Port9	P90 to P93	4	Input	—	(Fixed)	AN0 to AN3
PortA	PA0	1	I/O	↑	Bit	WAIT
	PA1	1	I/O	↑	Bit	TI0
	PA2	1	I/O	↑	Bit	TO1
	PA3	1	I/O	↑	Bit	TO3
PortB	PB0	1	I/O	↑	Bit	TI4/INT4
	PB1	1	I/O	↑	Bit	TI5/INT5
	PB2	1	I/O	↑	Bit	TO4
	PB3	1	I/O	↑	Bit	TO5
	PB4	1	I/O	↑	Bit	TI6/INT6
	PB5	1	I/O	↑	Bit	TI7/INT7
	PB6	1	I/O	↑	Bit	TO6
	PB7	1	I/O	↑	Bit	INT0

### 3.5.1 Port 1 (P10 - 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. Resetting resets all bits of output latch P1 and control register P1CR 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 (D8 to 15).

Port 1 always functions as a data bus (D8 to 15) ( $AM8/16 = "0"$ ).



**Figure 3.5 (1). Port 1**

	7	6	5	4	3	2	1	0	
P1 (0001H)	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)	bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0 : IN 1 : OUT							
Read-modify-write is prohibited for registers P1CR.									
Port 1 function setting									
AM8/16		0		1					
P1CR <P1XC>		0		Address data bus (D15 to 8)		Input port			
1		1		Output port					

Note : <P1XC> is bitX in register P1CR.

**Figure 3.5 (2). Registers for Port 1**

### 3.5.2 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 P2FC. Resetting resets all bits of output latch P2 and function register P2FC. Resetting also sets P2 to input mode.

With the TMP95C061, which has no internal ROM, all bits of P2FC are set to "1" and operate A23 to A16 after reset input.

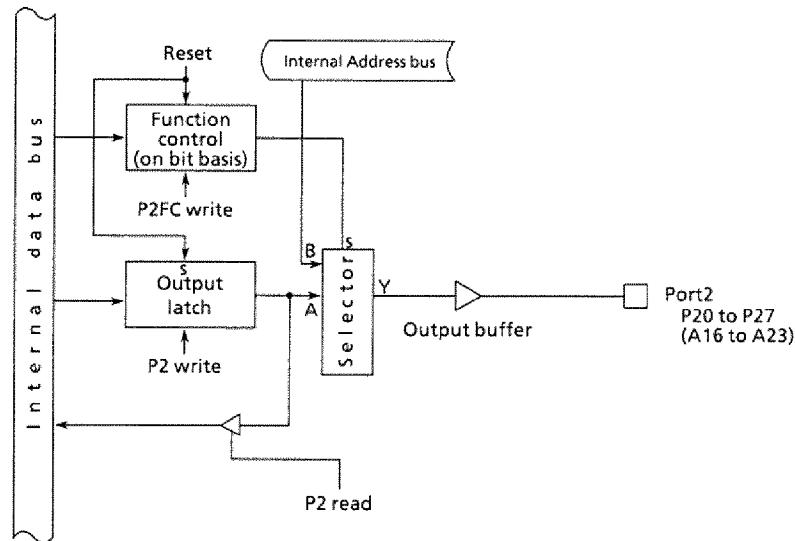


Figure 3.5 (3). Port 2

	7	6	5	4	3	2	1	0	
P2 (0006H)	bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W							
	After reset	(Output latch register is set to "1")							
Port 2 Function Register									
P2FC (0009H)	bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0 : Port, 1 : Address bus (A23 to A16)							

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

**Figure 3.5 (4). Registers for Port 2**

### 3.5.3 Port 5 (P52 to P55)

Port 5 is a 4-bit general-purpose I/O port. I/O can be set on bit basis using control register P5CR and the function register P5FC. Resetting does the following:

Resets all the bits of the output latch, the control register P5CR and the function register P5FC to "0" and sets each port input mode with pull-up resistors.

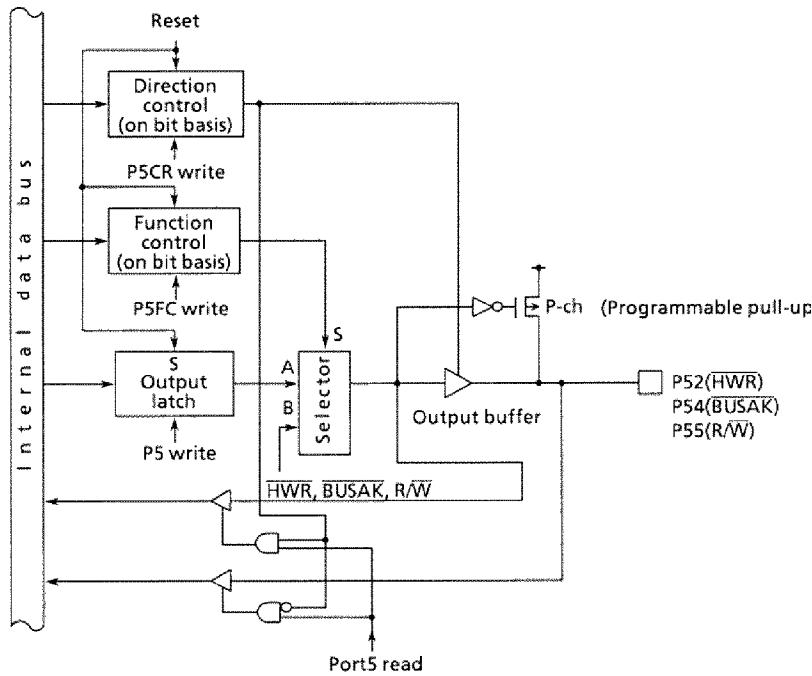
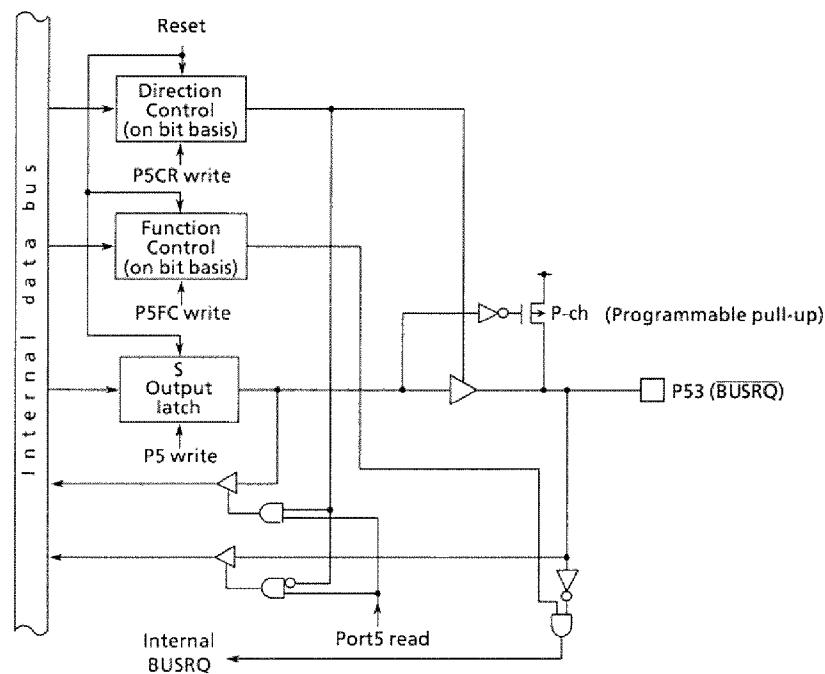


Figure 3.5 (5). Port5 (P50, P51, P52, P54, P55)



**Figure 3.5 (6). Port5 (P53)**

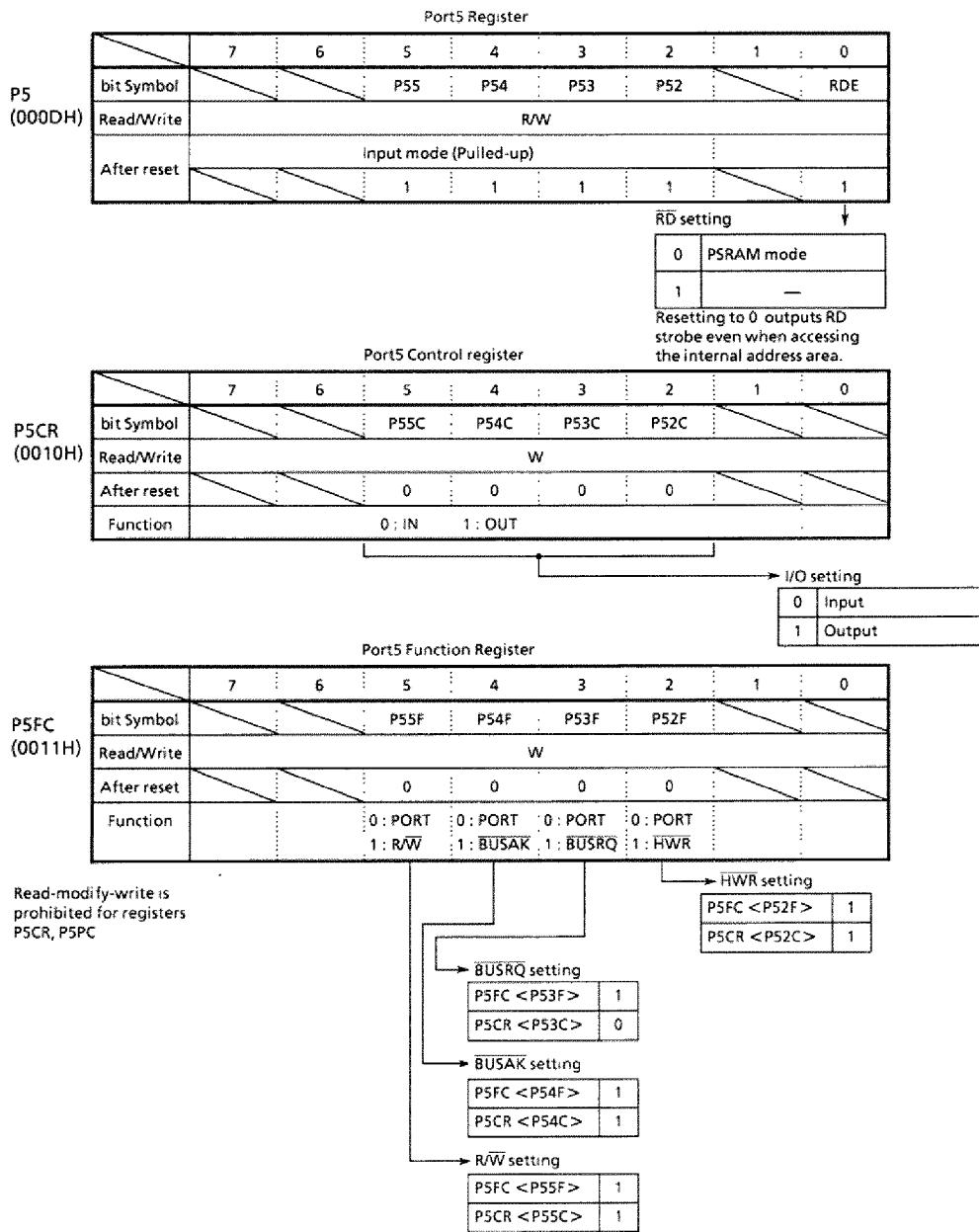


Figure 3.5 (7). Registers for Port5

### 3.5.4 Port6 (P60 to P65)

Port 6 is a 6-bit general-purpose output port. Resetting sets each output latch P62 = "0", P60, P61, P63 to P65 = "1".

Functions can be selected using P6FC and provided chip select and DRAM control functions (CS0 to 3, CAS, RAS and REFOUT). After resetting, each port operates as output port.

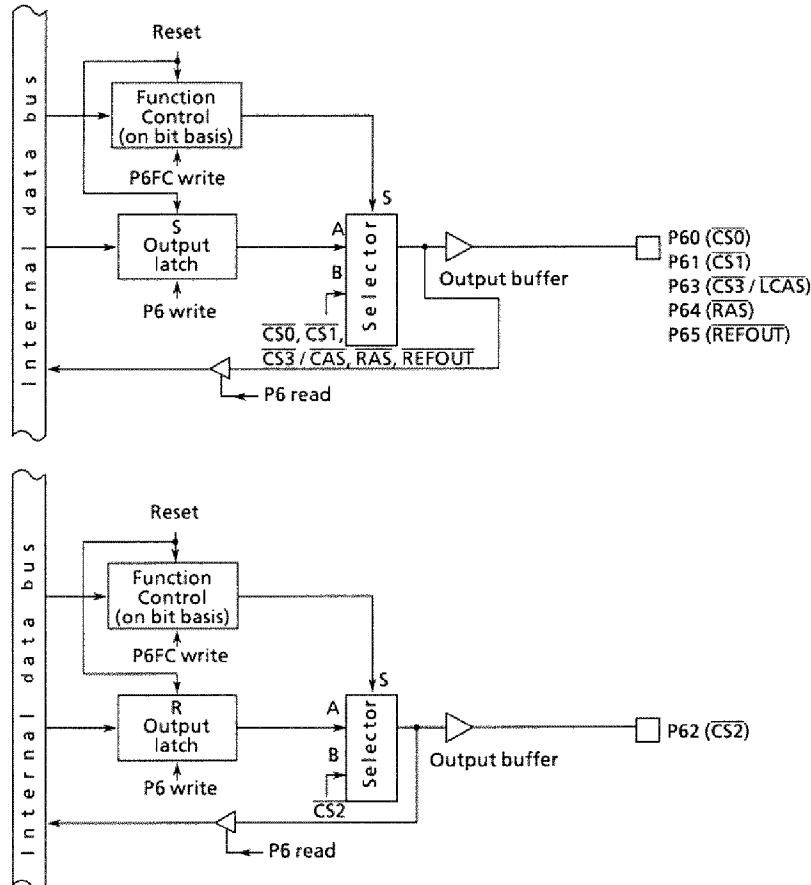


Figure 3.5 (8). Port6

Port 6 Register

	7	6	5	4	3	2	1	0	
P6 (0012H)				P65	P64	P63	P62	P61	P60
bit Symbol							R/W		
Read/Write									
After reset				1	1	1	0	1	1

Port 6 Function Register

	7	6	5	4	3	2	1	0
P6FC (0015H)			P65F	P64F	P63F	P62F	P61F	P60F
bit Symbol							W	
Read/Write								
After reset			0	0	0	0	0	0
Function			0 : PORT	1 : CS / CAS, RAS, REFOUT				

Read-modify-write is prohibited for registers P6FC

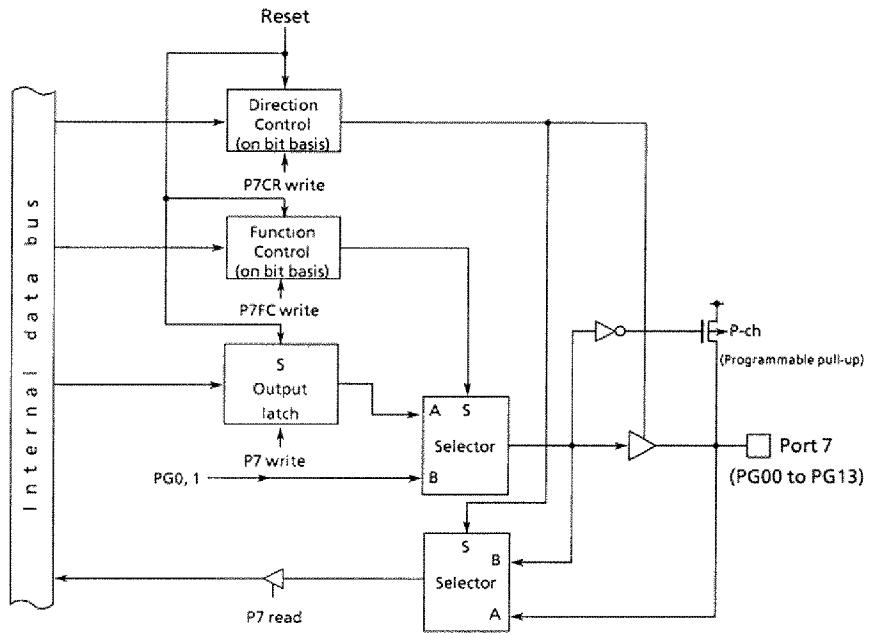
Note: The function of P63 (CS3 / CAS) is selected using B3CS register.

**Figure 3.5 (9). Register for Port 6**

### 3.5.5 Port7 (P70 to P77)

Port 7 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets Port 7 as an input port and connects a pull-up resistor. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, Port 7

also functions as a pattern-generator PG0/PG1 output. PG0 is assigned to P70 to P73; PG1, to P74 to P77. Writing in the corresponding bit of port 7 control register (P7CR) and function register (P7FC) enables PG output. Resetting resets the function register P7FC value to 0, and sets all bits to ports.



**Figure 3.5 (10). Port 7**

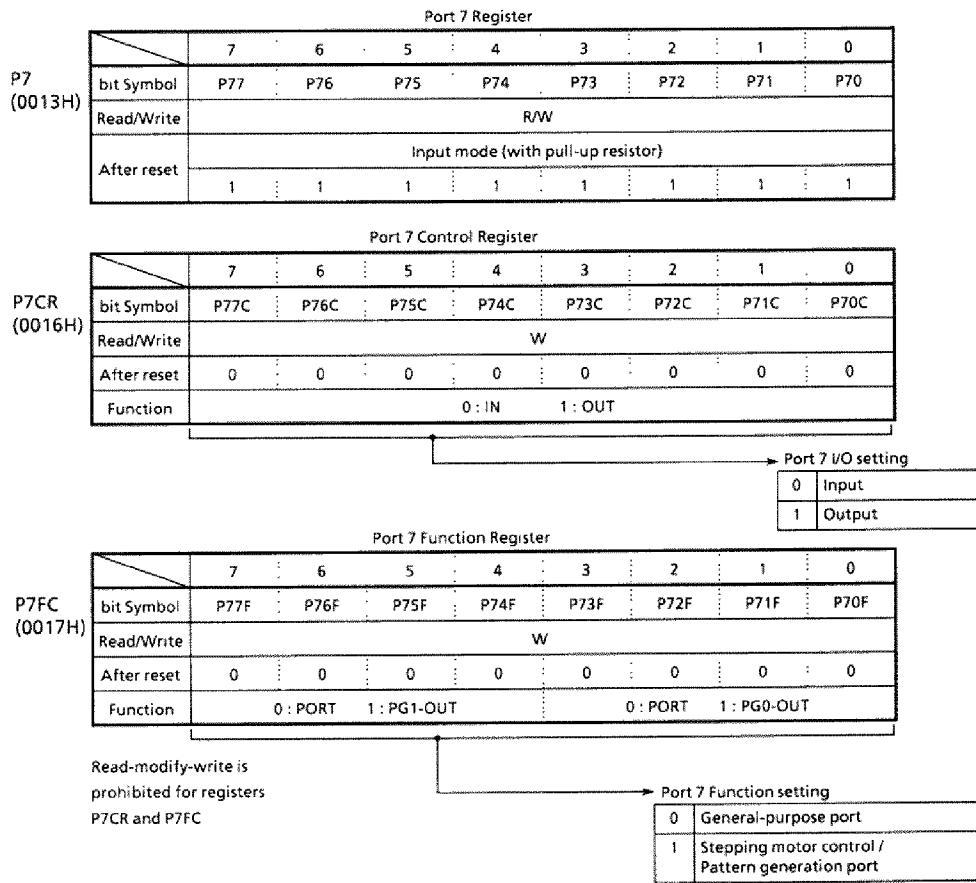


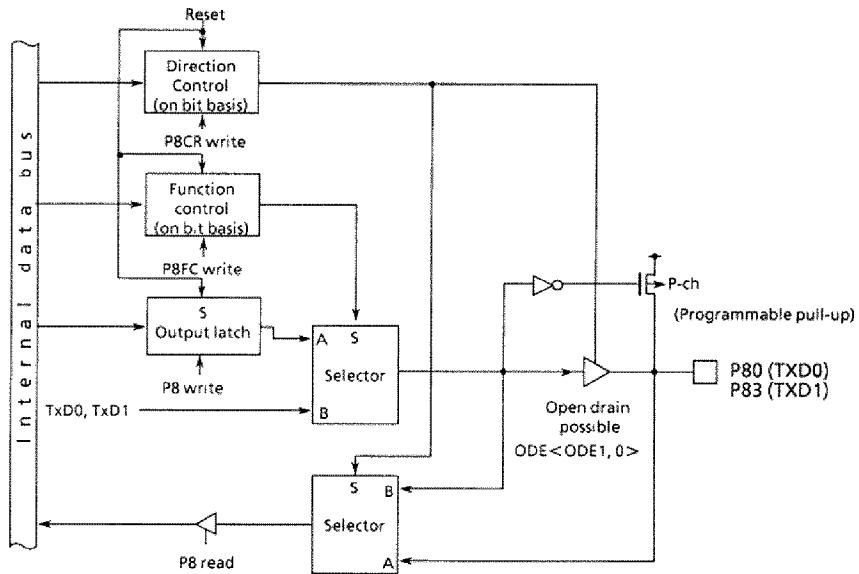
Figure 3.5 (11). Register for Port 7

### 3.5.6 Port 8 (P80 - P85)

Port 8 is a 6-bit general-purpose I/O port, also used as an analog input pin. I/O can be set on a bit basis. Resetting sets Port 8 as an input port and connects a pull-up resistor. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, Port 8 also functions as an I/O for serial channel 1, 0. Writing "1" in the corresponding bit of Port 8 function register enables those functions. Resetting resets the function register value to "0", and sets all bits to ports.

### (1) Port 80, 83 (TXD0/TXD1)

P80 and P83 also function as serial channel TXD output pins in addition to I/O ports. They have programmable open drain function.

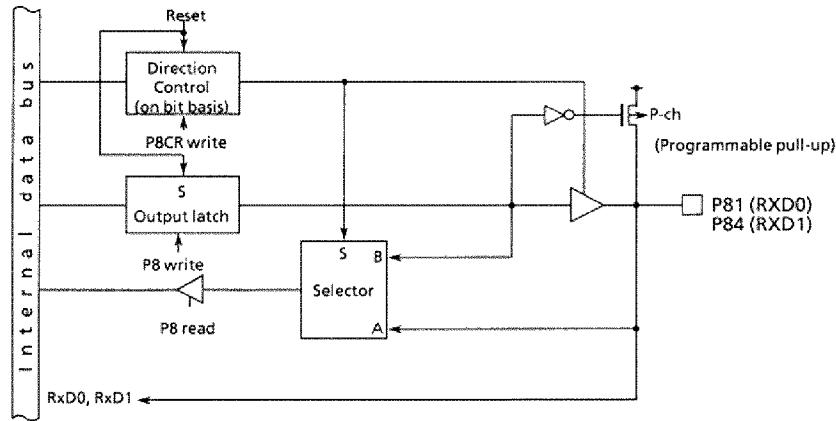


**Figure 3.5 (12). Port 80, 83**

- (2) Port 81, 84 (RXD0, 1)

input pins for serial channels.

P81 and P84 are I/O ports, and also used as RXD

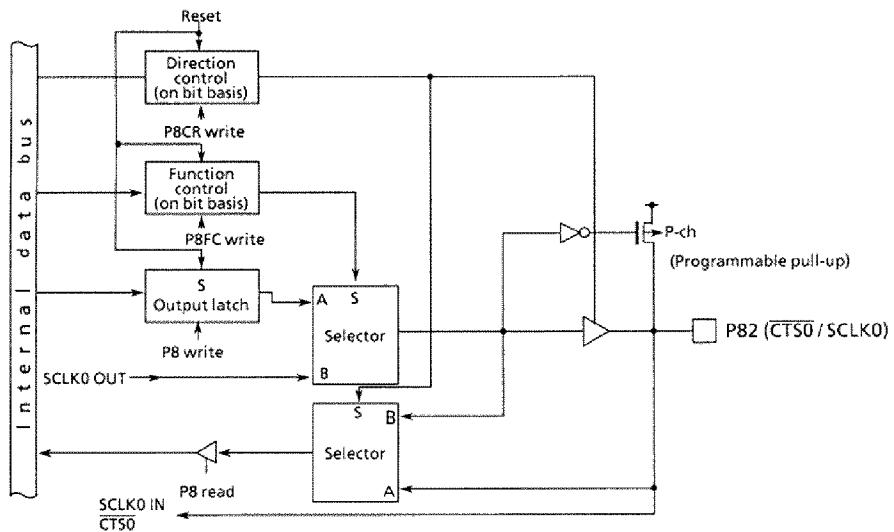


**Figure 3.5 (13). Port 81, 84**

- (3) Port 82 (CTS0/SCLK0)

as a SCLK0 I/O pin for serial channels.

P92 is an I/O port, and also used as a CTS input pin or

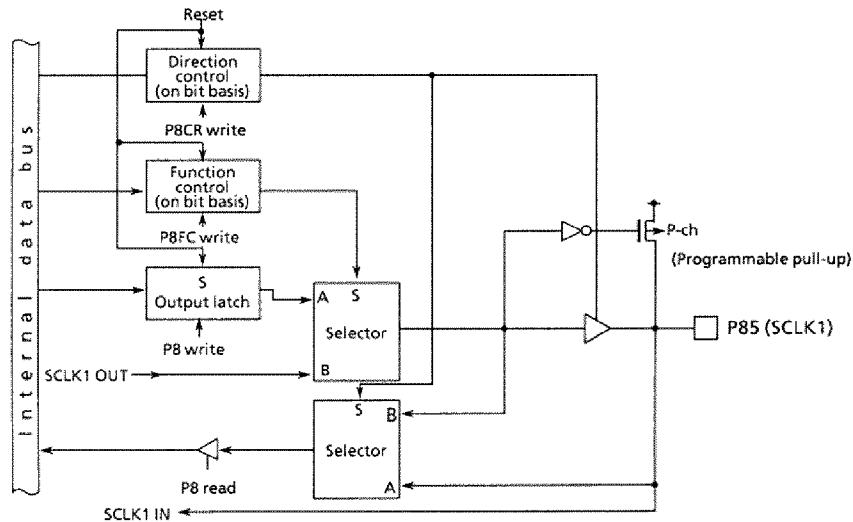


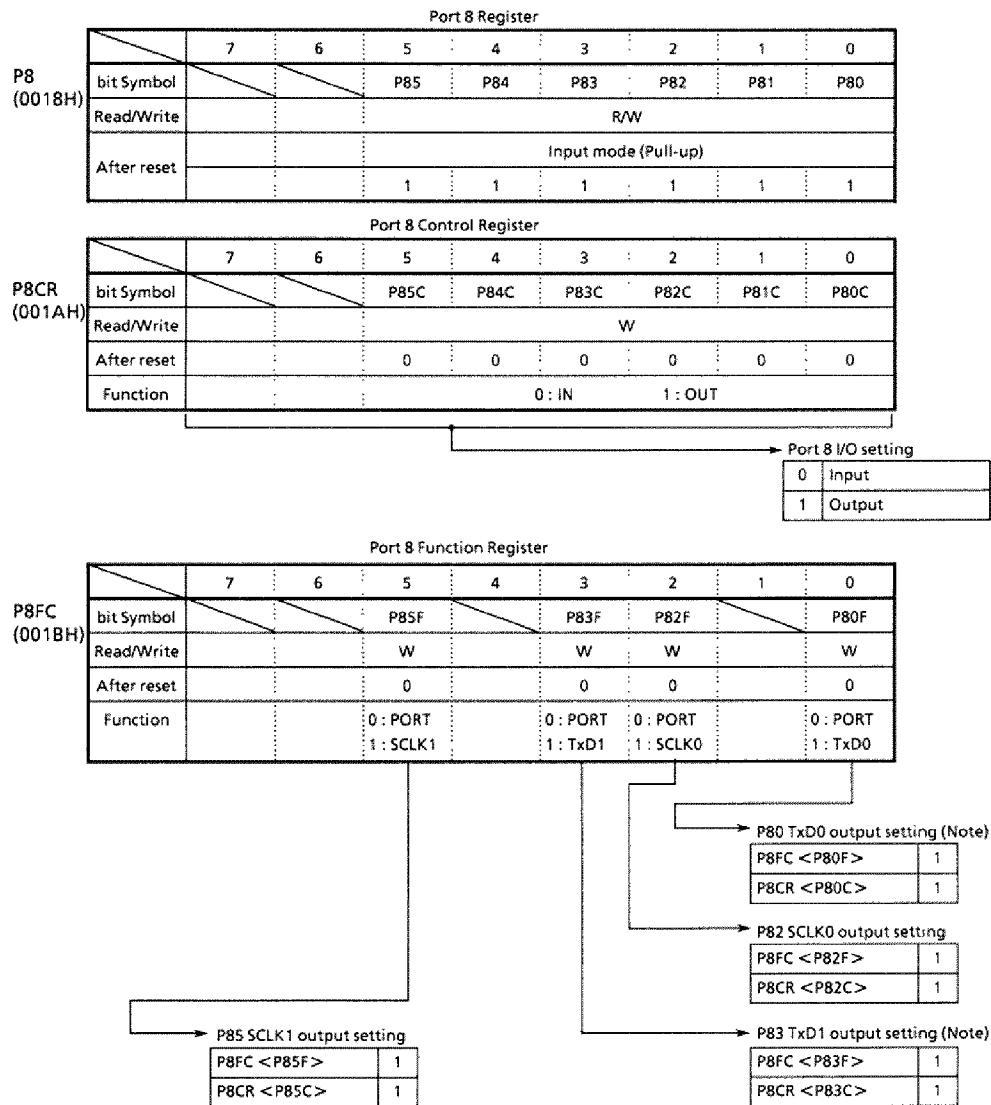
**Figure 3.5 (14). Port 82**

## (4) Port 85 (SCLK1)

SCLK1 I/O pin for serial channel 1.

P85 is a general-purpose I/O port. It is also used as a

**Figure 3.5 (15). Port 85**



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

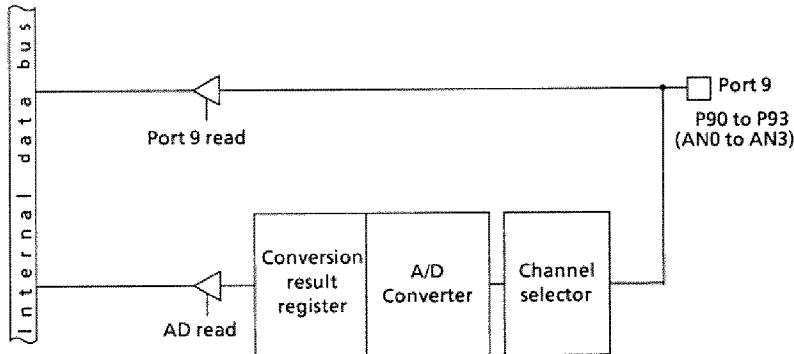
P81 / RXD0, P84 / RXD1 pins do not have a register changing PORT / FUNCTION.  
Therefore this is the same as P70 / TI0 pin.

**Figure 3.5 (16). Register for Port 8**

### 3.5.7 Port 9 (P90 to P93)

Port 9 is a 4-bit input I/O port, also used as analog input pins

for the internal A/D Converter.



**Figure 3.5 (17). Port 9**

Port 9 REgister								
P9 (0019H)	7	6	5	4	3	2	1	0
bit Symbol					P93	P92	P91	P90
Read/Write						R		
After reset							Input mode	

**Figure 3.5 (18). Register for Port 9**

### 3.5.8 Port A (PA0 to PA3)

Port A is a 4-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets Port 7 as an input port and connects a pull-up resistor. In addition to functioning as a general-purpose I/O port, Port A0 also functions as wait input pin  $\overline{WAIT}$ ;

Port A1 as an 8-bit timer input (TI0), Port A2 as a PWM0 output (TO1), and Port A3 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the Port A function register (PAFC) enables output of the timer. Resetting resets the function register PAFC value to 0, and sets all bits to ports.

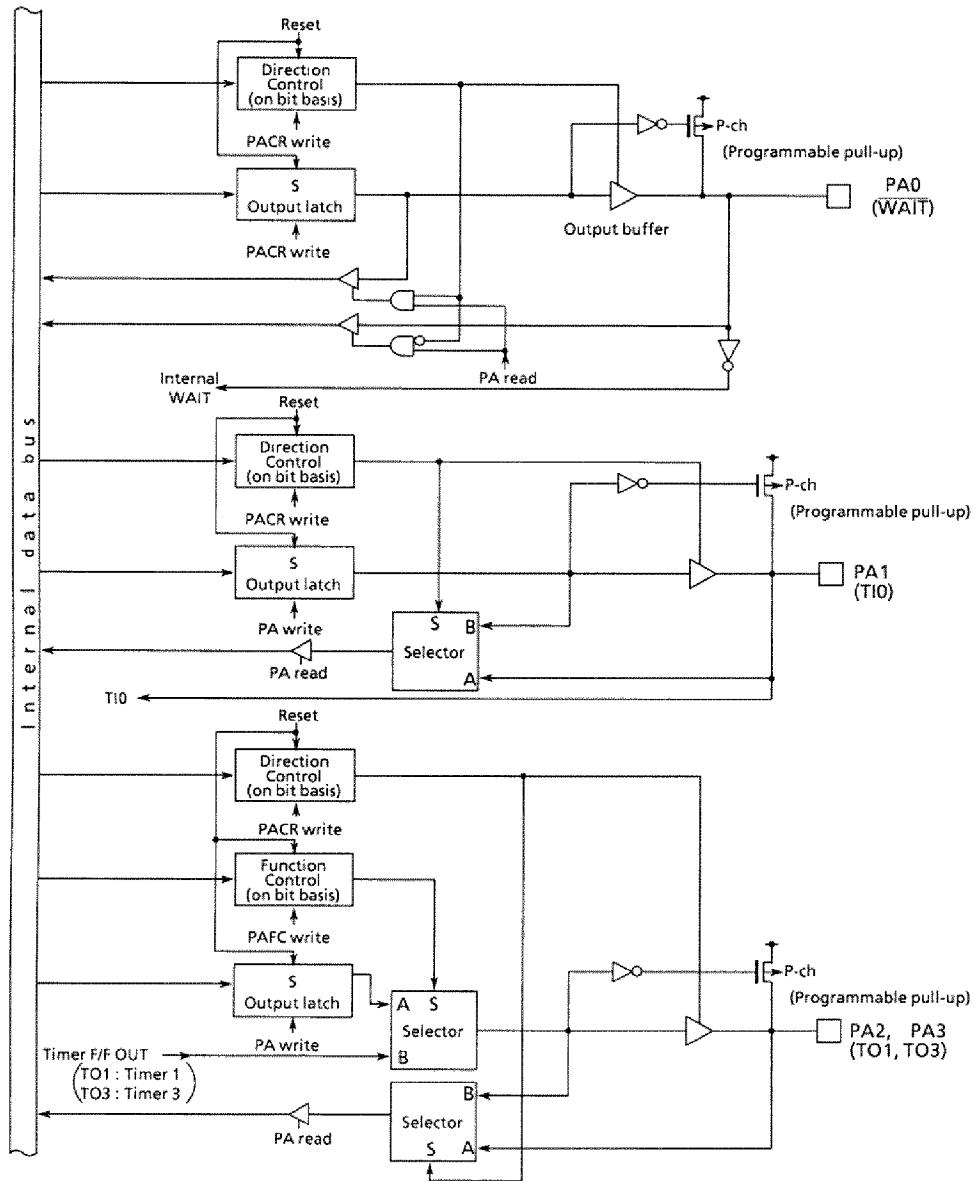
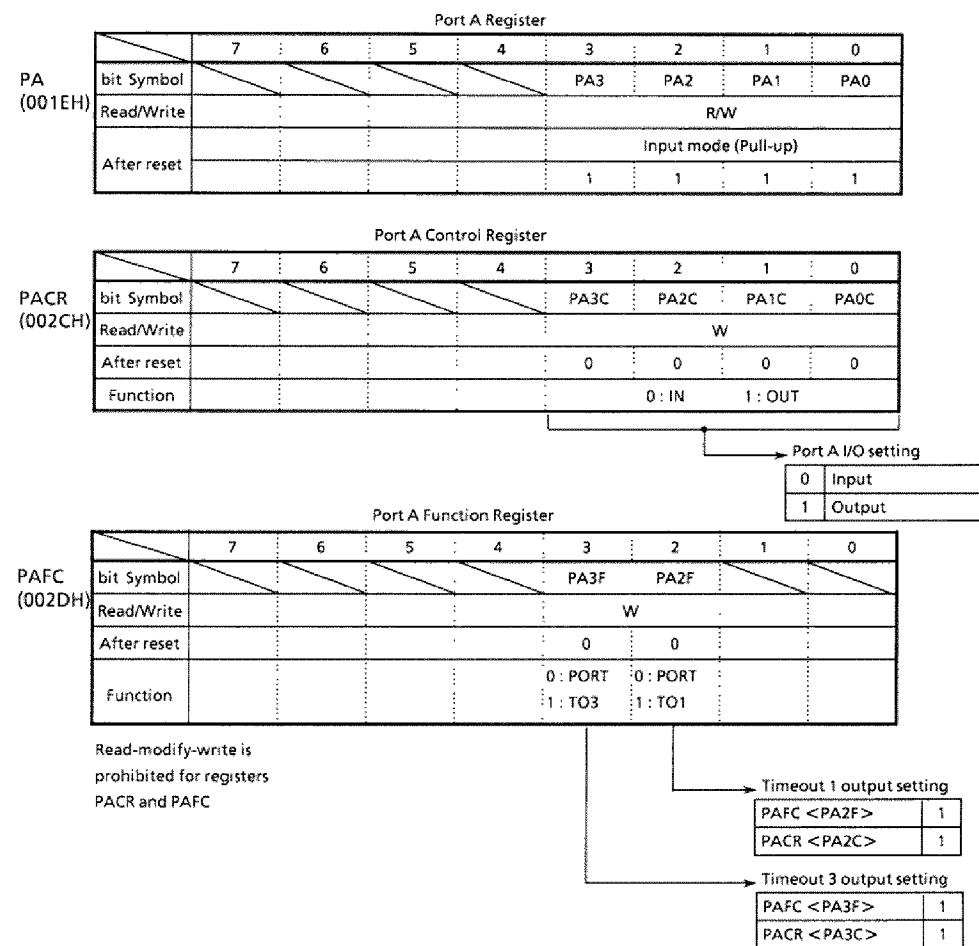


Figure 3.5 (19). Port A



Note: PA1/TI0 pin does not have a register changing Port / Function. For example, when it is used as an input port (PA1), the input signal for PA1 is inputted to 8 bit timer 0 as a timer input 0 (TI0).

**Figure 3.5 (20). Register for Port A**

### 3.5.9 Port B (PB0 to PB7)

Port B is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets Port B as an input port and connects a pull-up resistor. It also sets all bits of the output latch register PB to 1. In addition to functioning as a general-purpose I/O port, Port B also functions as an input for 16-bit timer 4 and 5

clocks, an output for 16-bit timer F/F 4, 5, and 6 output, and an input for INT0. Writing "1" in the corresponding bit of the Port B function register (PB FC) enables those functions. Resetting resets the function register PBFC value to "0", and sets all bits to ports.

- (1) PB0 ~ PB6

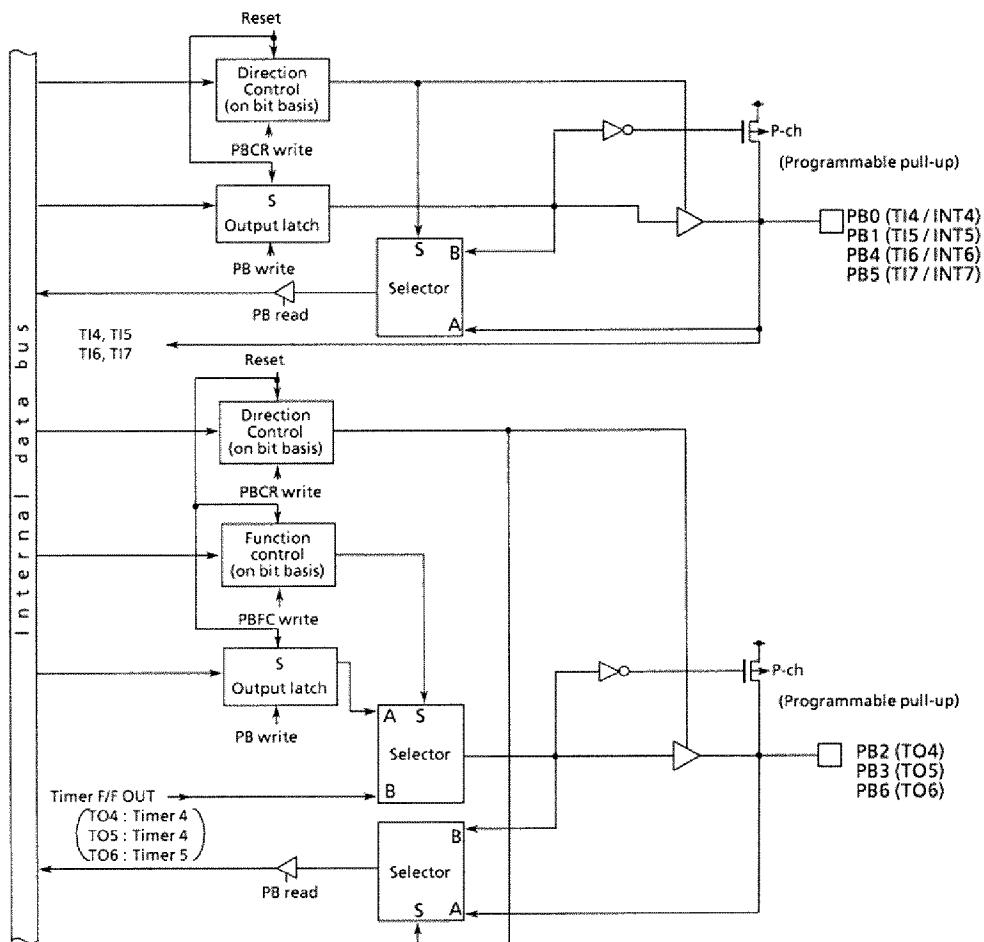
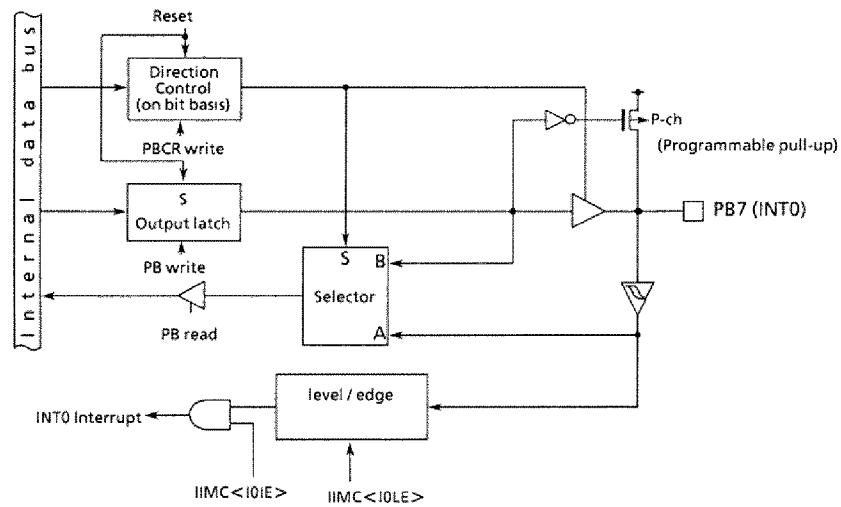


Figure 3.5 (21). Port B (PB0 - PB6)

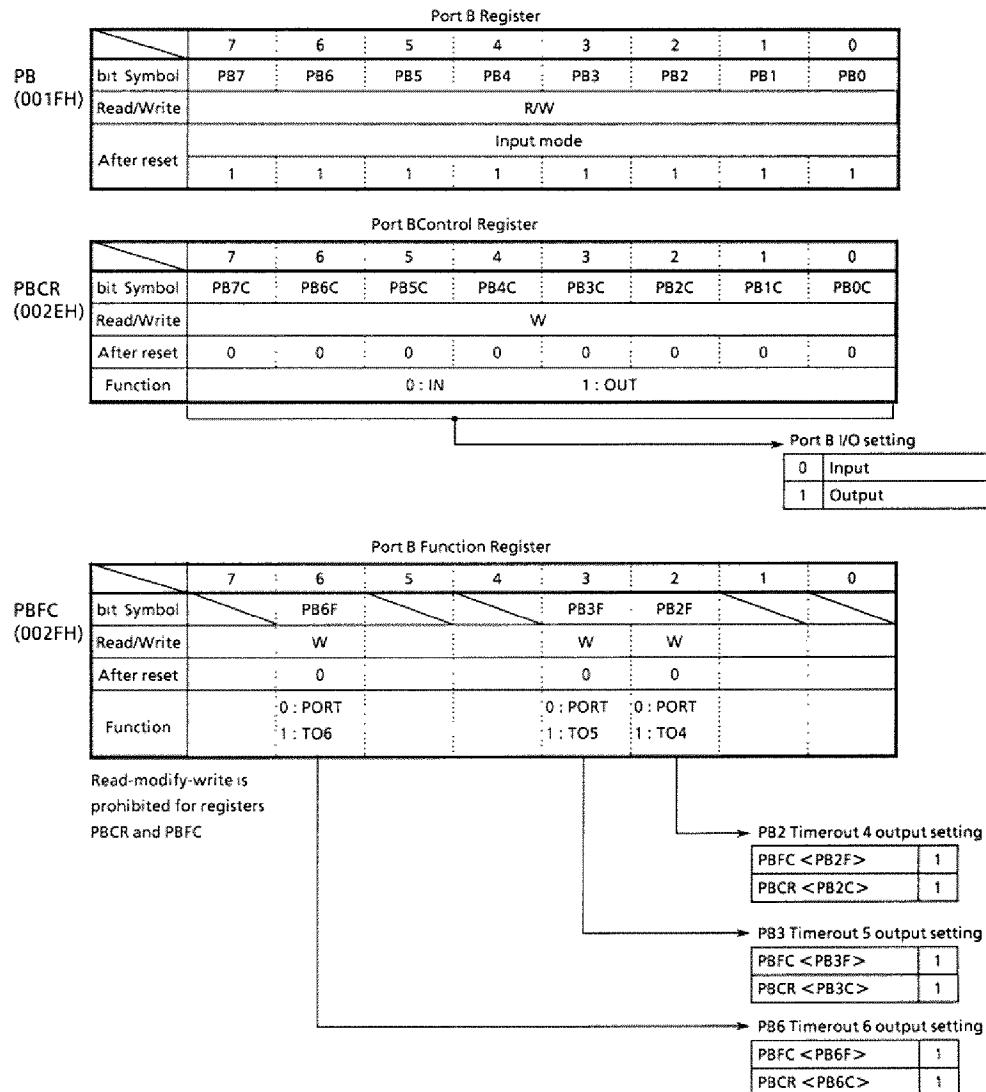
## (2) PB7 (INT0)

as an INT0 pin for external interrupt request input.

Port B7 is a general-purpose I/O port, and also used



**Figure 3.5 (22). Port B7**



Note) PB0 / T14, PB1 / T15, PB4 / T16 and PB5 / T17 pins do not have a register changing Port / Function.

Therefore this is the same as PA1 / T10 pin.

When P87 / INT0 pin is used as an INT0 pin, set PBCR <PB7C> to "0" and IIMC <IOIE> to "1".

**Figure 3.5 (23). Register for Port B**

### 3.6 Chip Select/Wait Control, AM8/16 pin

TMP96C061 has a built-in chip select/wait controller used to control chip select (CS0 to CS3 pins), wait (WAIT pin), and data bus size (8 or 16 bits) for any of the three block address areas.

Additionally, there is an AM8/16 pin which selects external data bus width for TMP95C061.

#### 3.6.1 Control Register

Table 3.6 (1) shows control registers

Each block address area is controlled by 1-byte CS/WAIT control registers. Start address register (MSAR0 to MSAR3) and address mask register (MAM0 to 3).

**Table 3.6 (1) Chip Select/Wait Control Register**

	7	6	5	4	3	2	1	0
bit Symbol				BOE	—	B0BUS	B0W1	B0W0
Read/Write				W	—	W	W	
After reset				0	—	0	0	0
Function				1: B0CS master bit	—	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	
bit Symbol				B1E	—	B1BUS	B1W1	B1W0
Read/Write				W	—	W	W	
After reset				0	—	0	0	0
Function				1: B1CS master bit	—	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	
bit Symbol				B2E	B2M	B2BUS	B2W1	B2W0
Read/Write				W	W	W	W	
After reset				1	0	0	0	0
Function				1: B2CS master bit	0: 16 M Area 1: Set MREG	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	
bit Symbol				B3E	B3CAS	B3BUS	B3W1	B3W0
Read/Write				W	W	W	W	
After reset				0	0	0	0	0
Function				1: B3CS master bit	0: CS3 output 1: CAS output	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	
bit Symbol				—	—	BEXBUS	BEXW1	BEXW0
Read/Write				—	—	W	W	
After reset				----	—	0	0	0
Function				----	—	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	

(1) Enable	Bit 4 (B0E, B1E, B2E, and B3E) of control register BXCS is a master bit used to specify enable (1)/disable (0) of the setting. Resetting sets B0E, B1E, and B3E to disable (0) and B2E to enable (1).	(3) Wait control	Control register bits 1 and 0 (B0W1, 0; B1W1, 0; B2W1, 0; B3W1, 0; BEXW1, 0) are used to specify the number of waits. Setting these bits to 00 inserts a 2-state wait regardless of the <u>WAIT</u> pin status. Setting them to 01 inserts a 1-state wait regardless of the <u>WAIT</u> status. Setting them to 10 inserts a 1-state wait and samples the <u>WAIT</u> 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 <u>WAIT</u> pin status. Resetting sets these bits to 00 (2-state wait mode).
(2) Data bus size select	Bit 2 (B0BUS, B1BUS, B2BUS, B3BUS, BEXBUS) 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. This bit is effective only in 16 bit bus mode ( $AM8/16 = 0$ ). In 8-bit bus mode ( $AM8/16 = 1$ ), this bit is negligible and all external memory areas are accessed in fixed 8 bit bus (See 3.1.2 External Data width selection pin ( $AM8/16$ )). 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.	Note: In case of competition of accessing and refreshing to DRAM, TMP95C061 automatically inserts refresh cycle in addition to settled wait cycle.	(4) CS/CAS Waveform Select
			Bit 3 of control register B3 is used to specify waveform mode output from the chip select pin ( <u>CS3/CAS</u> ). Setting this bit to 0 specifies <u>CS3</u> waveforms; setting it to 1 specifies <u>CAS</u> waveforms. Resetting clears bit 5 to 0.

Table 3.6 (2) Dynamic Bus Sizing

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

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

## (5) Extra CS Area Bus/Wait Control

BEXCS register is used to specify the data bus size and the number of wait in case of accessing address area which is not specified using CS0 to 3 register.s This register has no master enable bit, so always enable to unspecified area. Each bit has same meaning as BxCS.

## (6)

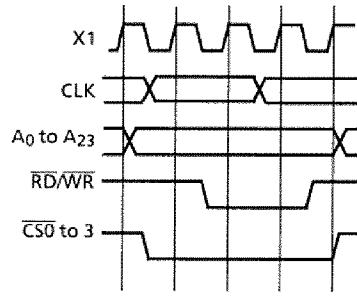
Setting B2CS <B2M> = 0 selects CS2 in the 16M-byte area (000080H to FFFFFFFH). Setting B2CS <B2M> = 1 selects CS2 according to the setting area for start address register MSAR2 and address mark register MAMR2, the same as for CS0 and SC1. A reset zero-clears this bit.

### 3.6.2 Address Area Specification

The address space is specified with the start address register (MSAR0 to 3). For each bus cycle, the chip select controller compares the address on the bus and value of this start address register. The value of the address mask register is used to ignore result of this address comparison. When there

is a match, the specified space is assumed to be accessed and a low strobe signal is output from the corresponding chip select pin ( $\overline{CS0}$  to  $\overline{CS3}$ ) if it is enabled ( $B0E$  to  $B3E = "1"$ ).

If the set address areas overlap or  $\overline{CS2}$  is enable for the 16M-byte area, the one with a smaller  $\overline{CS}$  number is selected.



**Figure 3.6 (1). Chip Select ( $\overline{CS0}$  to  $\overline{CS3}$ ) Operation Timing**

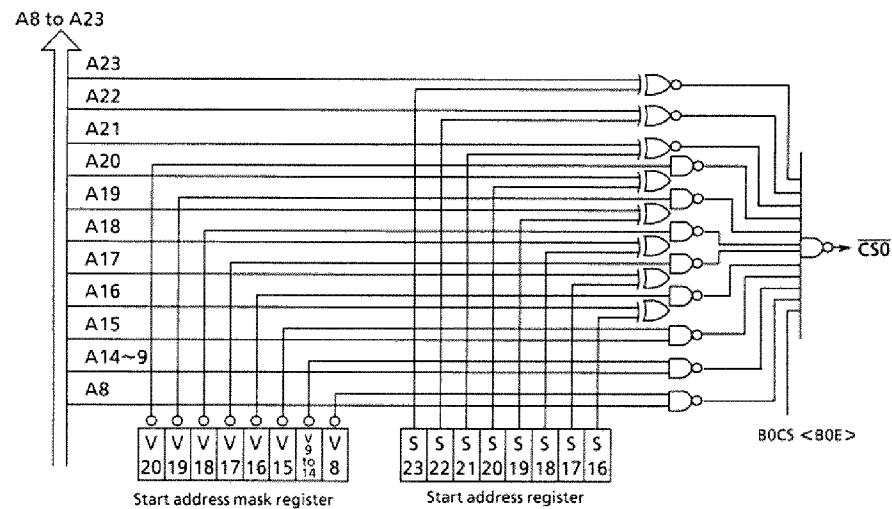


Figure 3.6 (2). CS0 Address Decode Block Diagram

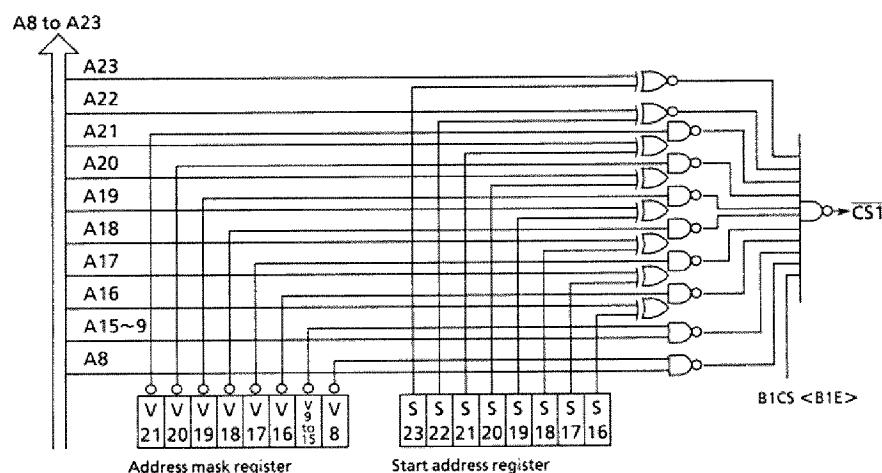
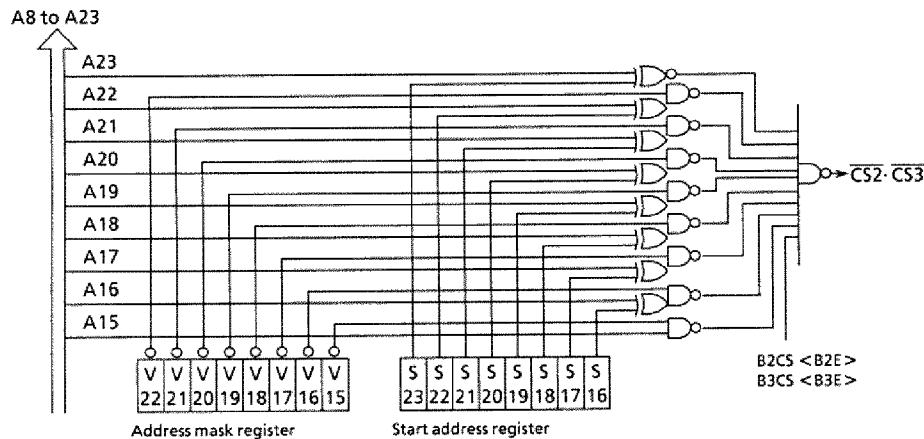


Figure 3.6 (3). CS1 Address Decode Block Diagram

**Figure 3.6 (4). CS1 Address Decode Block Diagram**

(1) Memory start address register

Memory address mask register

**Table 3.6 (3) Memory Start Address Register**

Memory address register ( $\overline{CS0}$ to $\overline{CS3}$ )										
	7	6	5	4	3	2	1	0		
MSAR0 (003CH)	MSAR1 (003EH)	bit Symbol	S23	S22	S21	S20	S19	S18	S17	S16
		Read/Write					R/W			
MSAR2 (005CH)	MSAR3 (005EH)	After reset	1	1	1	1	1	1	1	
		Function	Set start address A23 to A16							

Set start address for  $\overline{CS0}$  to  $\overline{CS3}$

**Table 3.6 (4) Memory Address Mask Register**

Memory address mask register (CS0)

	7	6	5	4	3	2	1	0
bit Symbol	V20	V19	V18	V17	V16	V15	V14 to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: Compare enable 1: Compare disable							

→ Control comparison of CS0 address  
A8 to A20

Memory address mask register (CS1)

	7	6	5	4	3	2	1	0
bit Symbol	V21	V20	V19	V18	V17	V16	V15 to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: Compare enable 1: Compare disable							

→ Control comparison of CS1 address  
A8 to A21

Memory address mask register (CS2, CS3)

	7	6	5	4	3	2	1	0
bit Symbol	V22	V21	V20	V19	V18	V17	V16	V15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: Compare enable 1: Compare disable							

→ Control comparison of CS2 to CS3  
address A15 to A22

MSAR0 3 < S23 > to < S16 > correspond to addresses A23 to A16 and S15, S14 to 9, and S8 corresponding to addresses A15, A14, to 9, and A8 are "0" by default. MAMR0 < V20 > to < V8 > enable/disable comparison of value set with MSAR0 and address and < V20 > to < V8 > correspond to < S20 > to < S16 >, S15, S14 to 9, and S8. In addition, V21, V22, and V23 corresponding to < S21 >, < S22 >, and < S23 > are "0" by default and comparison is always enabled.

Example of enabling/disabling comparison  
( $\overline{CS0}$  registers MSAR0 and MSAMR0)

When comparison is disabled by setting < V16 > = 1, the comparison of the value of < S16 > and address A16 is disabled and the value of < S16 > becomes invalid.

When comparison is enabled by setting < V16 > = 0, the comparison of the value of < S16 > and address A16 is enabled and  $\overline{CS0}$  is enabled only when they match.

$\overline{CS1}$ ,  $\overline{CS2}$ , and  $\overline{CS3}$  can be used in the same manner. Resetting sets the registers MSAR0, MSA1, MSAR2, MSAR3, MAMR0, MAMR1, MAMR2 and MAMR3 to "OFFH", and sets the control register bits B0E, B1E, to "0". So, chip select  $\overline{CS0}$ ,  $\overline{CS1}$ , and  $\overline{CS3}$  are disabled after resetting, while Bit B2E = 1, B2M = 0 and CS2 is enable for memory area 000080H to 0FFFFFH (16M byte).

## (2) How to the Start Address

The address decoder is output by specifying the start address for CS output and the space size.

The start address is set every 64K-byte because it is decoded by A16 to A23 as shown in the block diagram.

In other words, the DRAM start address is set to one of the 64K-byte intervals after "000000H".

However, note that the start address may be changed due to the value of the MAMR.

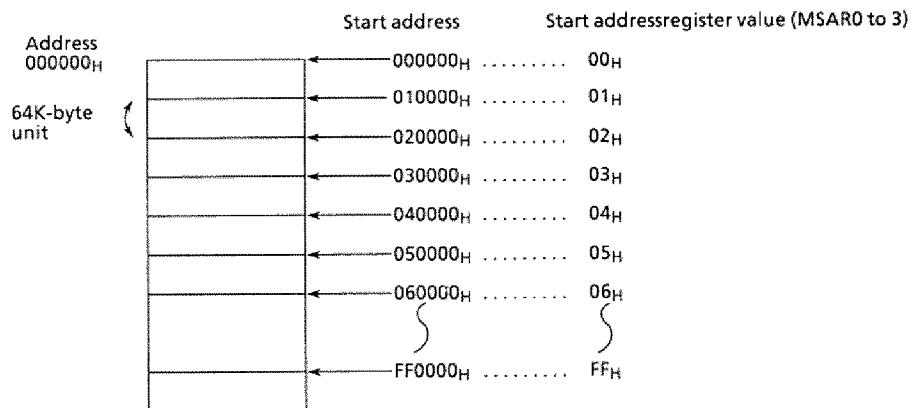


Figure 3.6 (5). Where to Set Start Address

## (3) How to Set the Address Space

The address space is specified by setting the memory start address mask register (MAMR0 to 3).

As shown in the address decoder block diagram (Fig-

ures 3.6 (2) to (4)),  $\overline{CS_0}$ ,  $\overline{CS_1}$ , or  $\overline{CS_2/CS_3}$  can specify the address area for which the chip select signal can be output depending on whether to compare the address A8 to A20, A8 to A21, or A15 to A22, respectively.

$\overline{CS}$	Size 256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
$CS_0$	<input type="radio"/>										
$CS_1$	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>							
$CS_2$			<input type="radio"/>								
$CS_3$			<input type="radio"/>								

**Figure 3.6 (6). Chip Select and Space Size**

## (4) Start Address/Address Space Setting Procedure

## (Setting Example)

- ① Set memory start address register (MSARx)  
(Set address)
- ② Set memory start address mask register (MAMRx)  
(Set area start area)
- ③ Set control register (BxCS)  
data bus width, number of waits, enable/disable of the area

When the setting the CS0 area to 64Kbyte (010000H to 01FFFFH), 16 bit data width and non-wait,

MSAR0 = 01H                    start address 010000H  
MAMR0 = 07H                    address area 64Kbyte  
B0CS = 13H                    16 bit data width, 0-wait

**3.7 Dynamic RAM (DRAM) Controller**

TMP95C061 consists of a control circuit to refresh DRAM, an access circuit to perform read/write.

- |   |   |
|---|---|
| 1) refresh mode<br>CAS before $\overline{RAS}$ refresh mode | 4) address mapping size<br>$\overline{CS3}$ area: 64K-8M byte   |
| 2) refresh interval<br>31-195 states (programmable)         | 5) memory access address length<br>8-11 bits  |
| 3) refresh cycle width<br>2-9 states (programmable)         | 6) wait controller<br>depends on the setting CS/WAIT controller   |
|   | 7) arbitration between refresh and memory access<br>refreshing is prior to memory access, automatically inserted wait cycle during memory access cycle. |

## Control Register

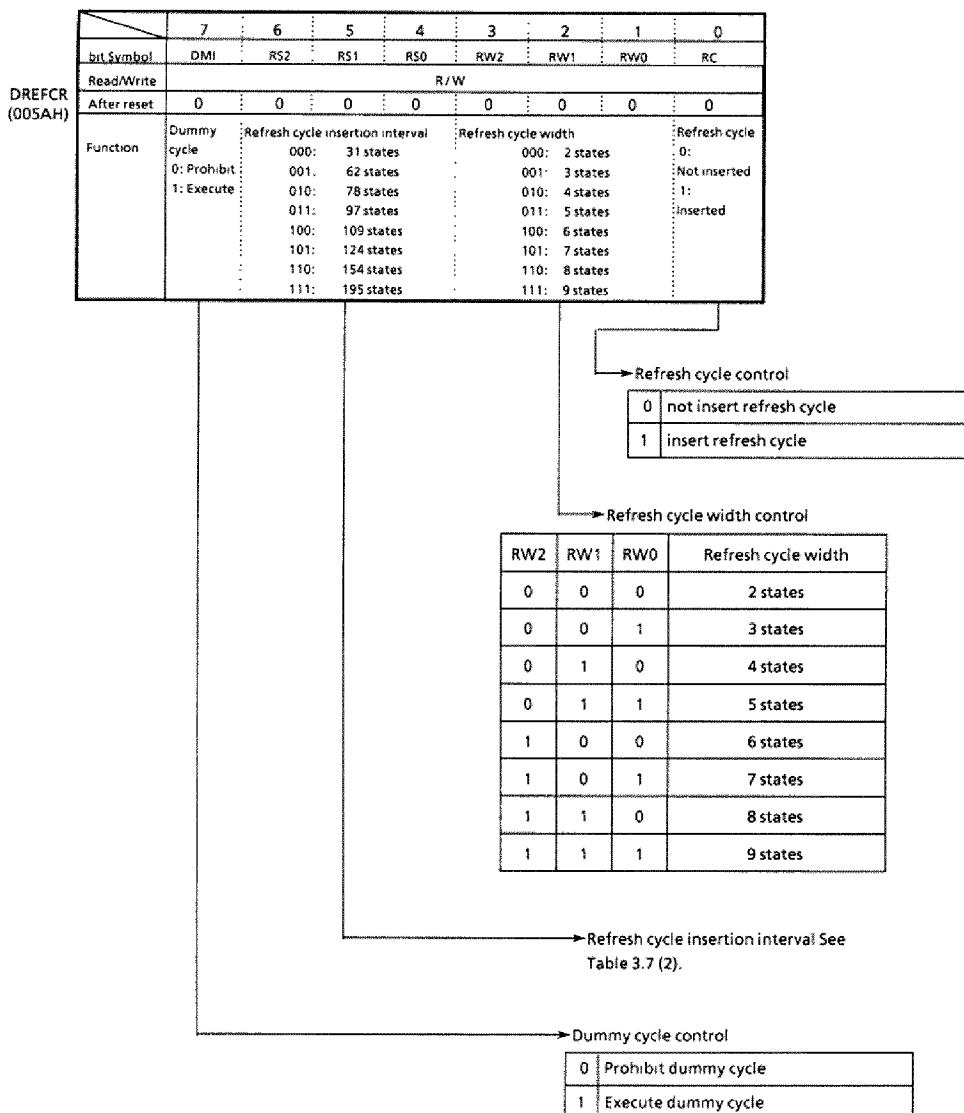
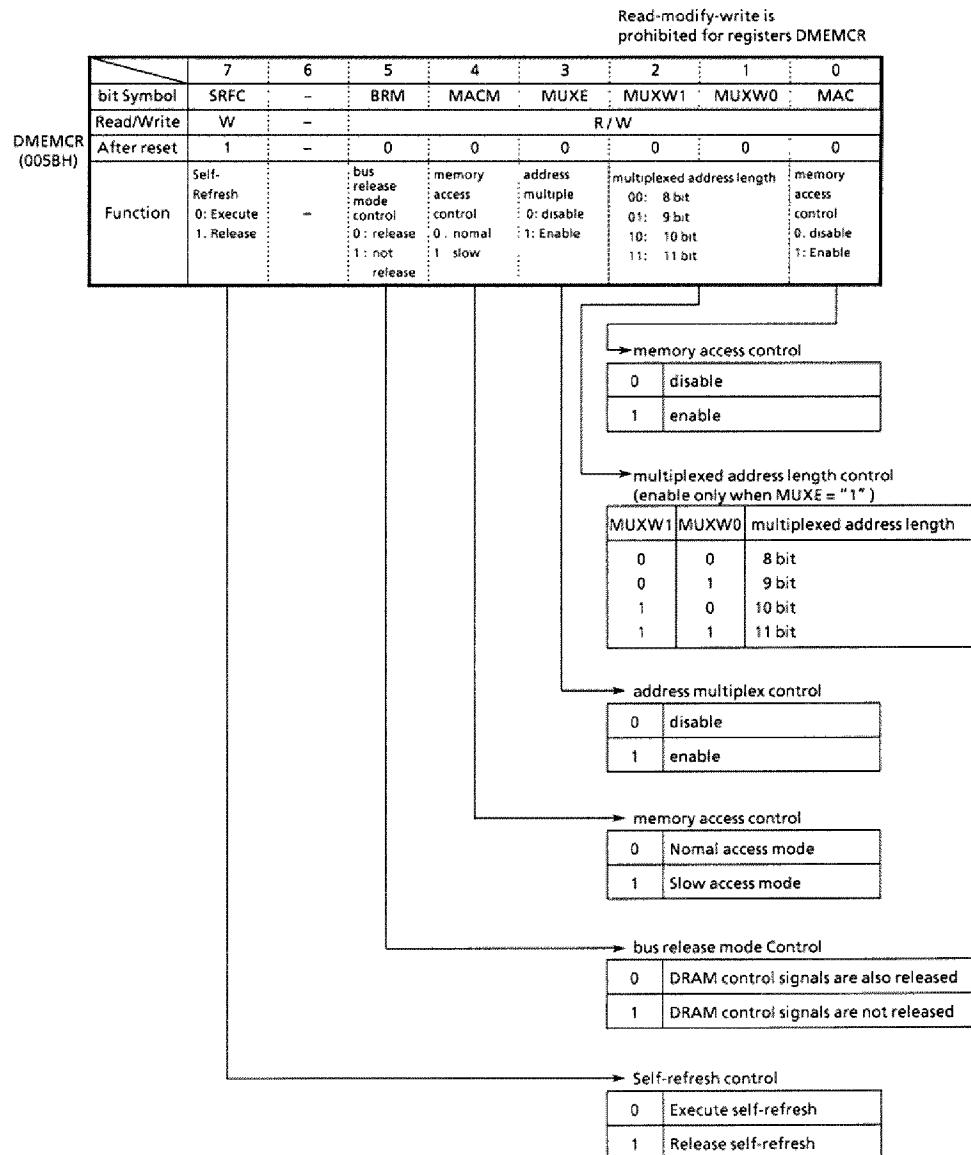


Figure 3.7 (1). Refresh Control Register



**Figure 3.7 (2). DRAM Memory Access Control Register**

## Operation Description

### (1) Memory Access Control

Access control is enable when DMEMCR <MAC> = 1. And then DRAM control signals (RAS, CAS, and REFOUT) are output during the time CPU access CS3 area. The cycle (bus width and number of wait) depend on the value of CS/WAIT controller.

To facilitate connection with low-speed DRAM, the DRAM controller can accelerate RAS rise at wait inser-

tion and delay RAS precharge time (RAS high width). This is called slow access mode. Set mode to slow access using DMEMCR <MACM>.

In the access cycle, Address multiplexer outputs row/column address through A0 to A11 pin. The enable/disable setting of address multiplexing and multiplexed address width are controlled by DMEMCR <MUXE> and <MUXW0, 1>. The relation between address width and bus width is below.

Figures 3.7 (3), (4) show the access timing.

**Table 3.7 Address Multiplex**

row address	column address							
	8 BIT		9 BIT		10 BIT		11 BIT	
	8	16	8	16	8	16	8	16
A0	A8	-	A9	-	A10	-	A11	-
A1	A9	A9	A10	A10	A11	A11	A12	A12
A2	A10	A10	A11	A11	A12	A12	A13	A13
A3	A11	A11	A12	A12	A13	A13	A14	A14
A4	A12	A12	A13	A13	A14	A14	A15	A15
A5	A13	A13	A14	A14	A15	A15	A16	A16
A6	A14	A14	A15	A15	A16	A16	A17	A17
A7	A15	A15	A16	A16	A17	A17	A18	A18
A8	-	A16	A17	A17	A18	A18	A19	A19
A9	-	-	-	A18	A19	A19	A20	A20
A10	-	-	-	-	-	A20	A21	A21
A11	-	-	-	-	-	-	-	A22

.....multiplexed  
 address width  
 .....bus width  
 (depend on the  
 value of  
 CS / WAIT  
 controller)

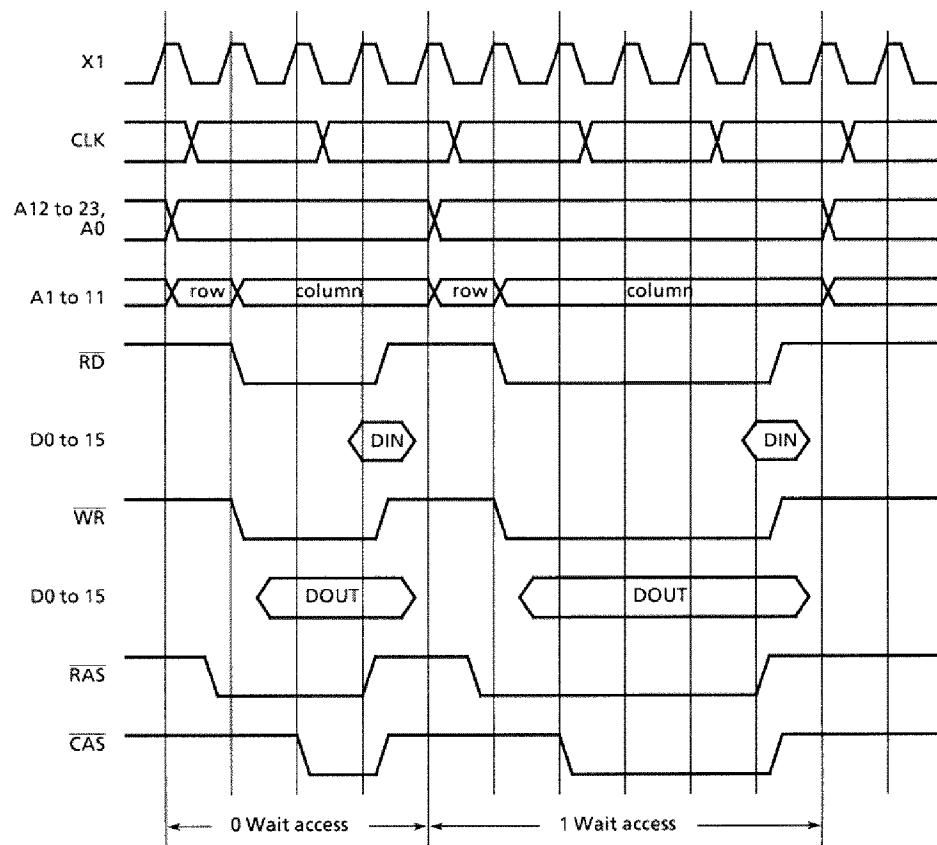


Figure 3.7 (3). DRAM Access Timing (Normal Access Mode)

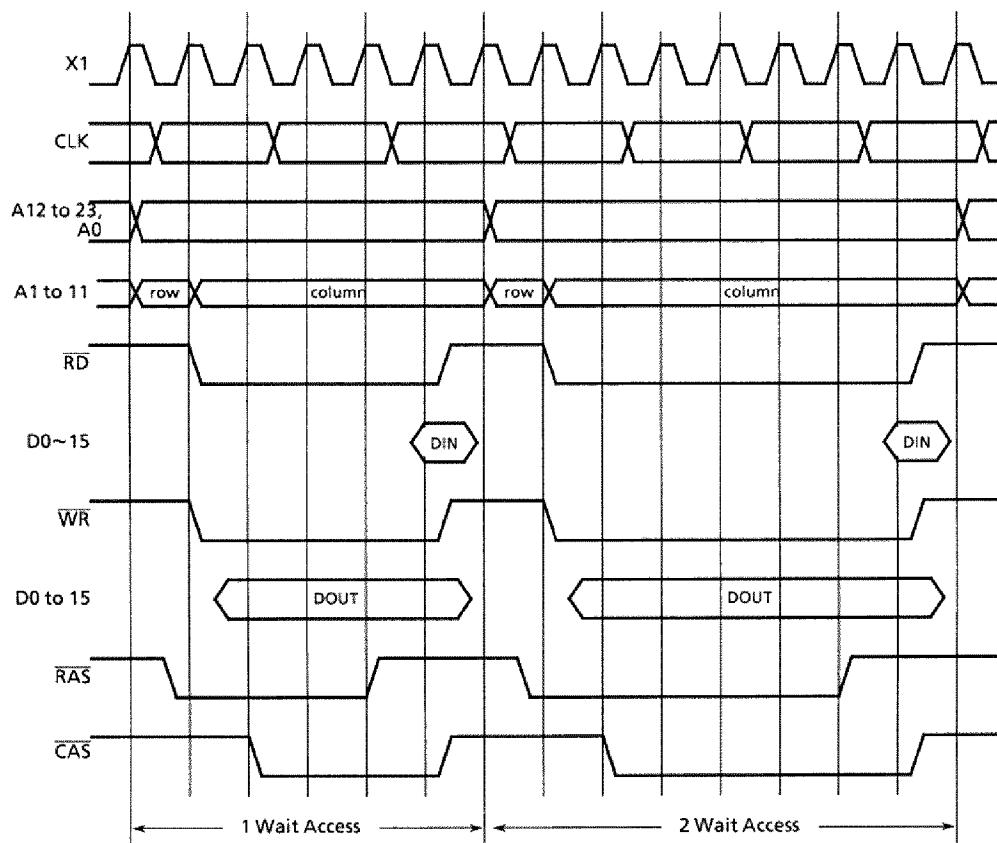


Figure 3.7 (4). DRAM Access Timing (Slow Access Mode)

## (2) Refresh Controller

The TMP95C061 can output RAS/CAS used to refresh the DRAM. At the same time the state signal REFOUT which indicates a refresh cycle is output. (Only for interval refresh mode.)

DRAM can be refreshed easily because RAS/CAS/REFOUT output frequency and pulse width are programmable.

The refresh controller has the following features.

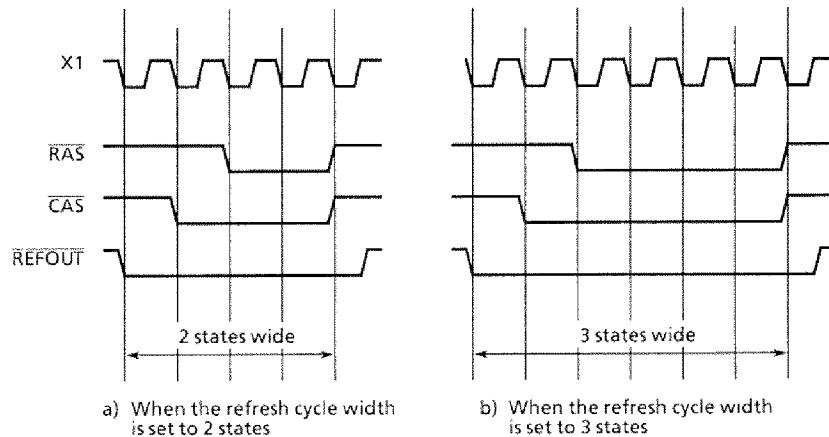
- Refresh mode: CAS before RAS interval refresh mode  
CAS before RAS self refresh mode
- Refresh interval: 31 to 195 states (programmable)
- Refresh cycle width: 2 to 9 states (programmable)
- Dummy cycle can be generated
- Refresh cycle is asynchronous with CPU operation cycle

i) CAS before RAS interval refresh mode

The refresh interval and refresh width for CAS before RAS interval refresh mode depends on the DRAM being used.

Therefore, TMP95C061 enables the refresh interval and refresh cycle width to be set with the refresh controller register value according to the system clock and DRAM that are being used.

Figure 3.7 (5) shows a timing example for CAS before RAS refresh cycle.



**Figure 3.7 (5). Refresh Cycle Timing Example**

How to set the register is described next.

Figure 3.7 (1) shows the bit structure of the refresh control register DREFCR.

① Refresh cycle insertion interval

The insertion interval is set with the three bits DREFCR <RS22 to 0> according to the system clock being used.

Example: When the system clock is 25MHz and the DRAM refresh cycle is to be 15.6 $\mu$ s, set these bits to "111".

**Table 3.13 (2) Refresh Cycle Insertion Interval**

Refresh Cycle			Insertion Interval (states)	Frequency (fosc)						
RS2	RS1	RS0		8 MHz	10 MHz	12.5 MHz	14 MHz	16 MHz	20 MHz	25 MHz
0	0	0	31	7.55	6.2	4.96	4.43	3.88	3.1	2.5
0	0	1	62	15.5	12.4	9.92	8.86	7.75	6.2	5.0
0	1	0	78	19.5	15.6	12.48	11.14	9.75	7.8	6.2
0	1	1	97	24.25	19.4	15.52	13.86	12.13	9.7	7.7
1	0	0	109	27.25	21.8	17.44	15.57	13.63	10.9	8.7
1	0	1	124	31.0	24.8	19.84	17.72	15.5	12.4	9.9
1	1	0	154	38.5	30.8	24.7	22.0	19.3	15.4	12.3
1	1	1	195	97.5	48	39.0	31.2	27.3	24.4	15.6

(Unit :  $\mu$ s)

② The three bits DREFCR <RW2 to 0> can be used to change the refresh cycle width (RAS, CAS Low output width). (2 to 9 states)

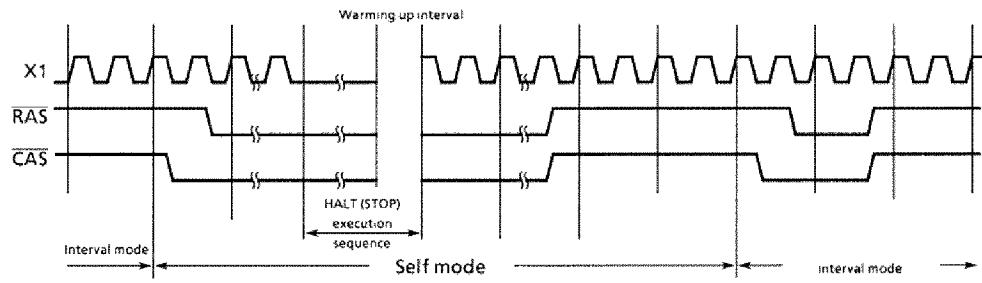
③ Refresh cycle control

The refresh cycle can be disabled/enabled with the bit DREFCR <RC>.

ii)  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  self refresh mode

This mode is used when DRAM controller or is halted with HALT (IDLE, STOP) instruction while refreshing with CAS before RAS interval refresh mode (hereafter referred to as interval mode).

However,  $\overline{\text{REFOUT}}$  is not output. ("1" is output.) Figure 3.7 (6) shows the self refresh mode timing diagram.



**Figure 3.7 (6). Self Refresh Cycle Timing**

This mode is executed as follows. First, the settings are made from normal interval mode. Then B3CS <SRFC> is set to "0" before a HALT instruction to perform one normal refresh. Then the CAS pin and RAS pin are kept at low level and the self refresh mode is entered. Cancelling HALT and supplying a clock to the DRAM controller automatically sets DMEMCR <SRFC> to 1 and cancels self refresh mode. After cancellation, refresh is performed once normally and processing returns to interval mode. (Note that when HALT is cancelled by a reset, the I/O registers are initialized, therefore, refresh is not normally performed.)

After DMEMCR <SRFC> to "0", make sure that the

## (4) Priority

The DRAM refresh cycle may overlap with the DRAM read/write cycle because it is not synchronized with the CPU operating cycle. In this case, the DRAM controller gives priority to the cycle that starts operation first. If the priority is given to the refresh cycle, a wait is automatically inserted in the memory access cycle.

## (5) Bus Release Mode

The TMP95C061 has a bus release function. Setting dedicated DRAM control pins (RAS, CAS, REFOUT) enables selection of release mode (by setting the pins to high impedance like other pins) or non-release (remain driving) mode in which refresh cycle output is supported. For the states of other pins at bus release, see 3.14 (2), Pin states at bus release.

## (i) Mode used by DRAM control dedicated pin to release bus (DMECR &lt;BRM&gt; = 0)

When the bus release request (BUSRQ) pin is set to active (low level), the TMP95C061 acknowledges the bus release request. After the current bus cycle (including DRAM access cycle) ends, the TMP95C061 sets the DRAM control dedicated pin (RAS, CAS, REFOUT) to high, sets the output buffer

to off, and sets the pin to high impedance.

The refresh cycle is asynchronous with the access cycle. When a refresh request is generated and the refresh cycle is at wait because of a conflict with the access cycle until the bus release, the bus release timing is delayed until the refresh cycle is completed. The refresh counter keeps counting during bus release. The refresh request generated during bus release is held for one cycle. The refresh cycle is performed immediately after the TMP95C061 regains bus mastership.

The bus release request or refresh counter is asynchronous with the bus cycle. To use this mode, the external bus master must generate a refresh cycle during bus release.

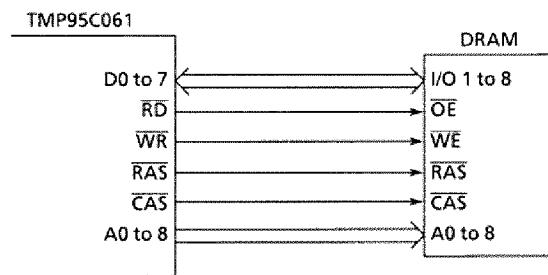
## (ii) Mode not used by DRAM control dedicated pin to release bus (DMECR &lt;BRM&gt; = 1)

Valid even if the DRAM is not accessed by the external bus master during bus release. If this mode is set, the DRAM dedicated pin does not release the bus even if a bus release request is generated but keeps supporting a refresh cycle only. Note that all the other pins release the bus. Unlike (i), bus release timing is not influenced by a refresh request.

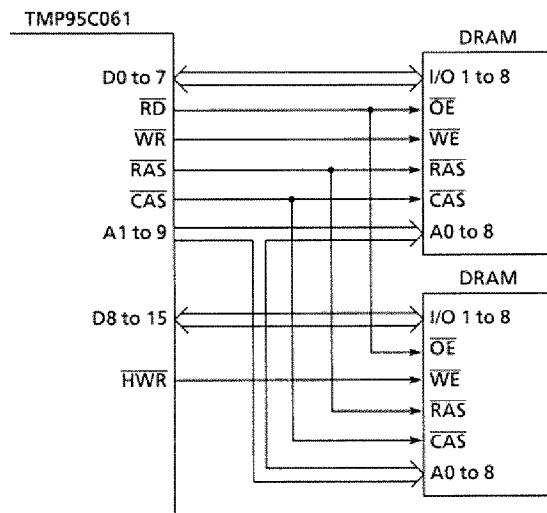
A reset DMECR <BRM> to 0 and the DRAM control dedicated pin to bus release mode.

## (6) Connection Example

## (1) 8 bit bus configuration



## (2) 16 bit bus configuration



### 3.8 8-bit Timers

TMP95C061 contains four 8-bit timers (timers 0, 1, 2 and 3), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timers. The following four operating modes are provided for the 8-bit timers:

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

Figure 3.8 (1) shows the block diagram of the 8-bit timer (timer 0 and timer 1).

Timers 2/3 have the same circuit configuration as timer 0 and timer 1. The difference between Timer 0 and Timer 2 is that Timer 0 has an external clock input pin (TIO), while Timer 2 has none.

Each interval timer consists of an 8-bit comparator, and 8-bit timer register. Besides, timer flip-flops (TFF1, TFF3) are provided for each pair of timer 0/1 and timer 2/3.

Among the input clock sources for the interval timers, the internal clocks of  $\phi T_1$ ,  $\phi T_4$ ,  $\phi T_{16}$ , and  $\phi T_{256}$  are obtained from the 9-bit prescaler shown in Figure 3.8 (2).

The operation modes and timer flip-flops of the 8-bit timer are controlled by five control registers T01MOD, T23MOD, TFFCR, TRUN, and TRDC.

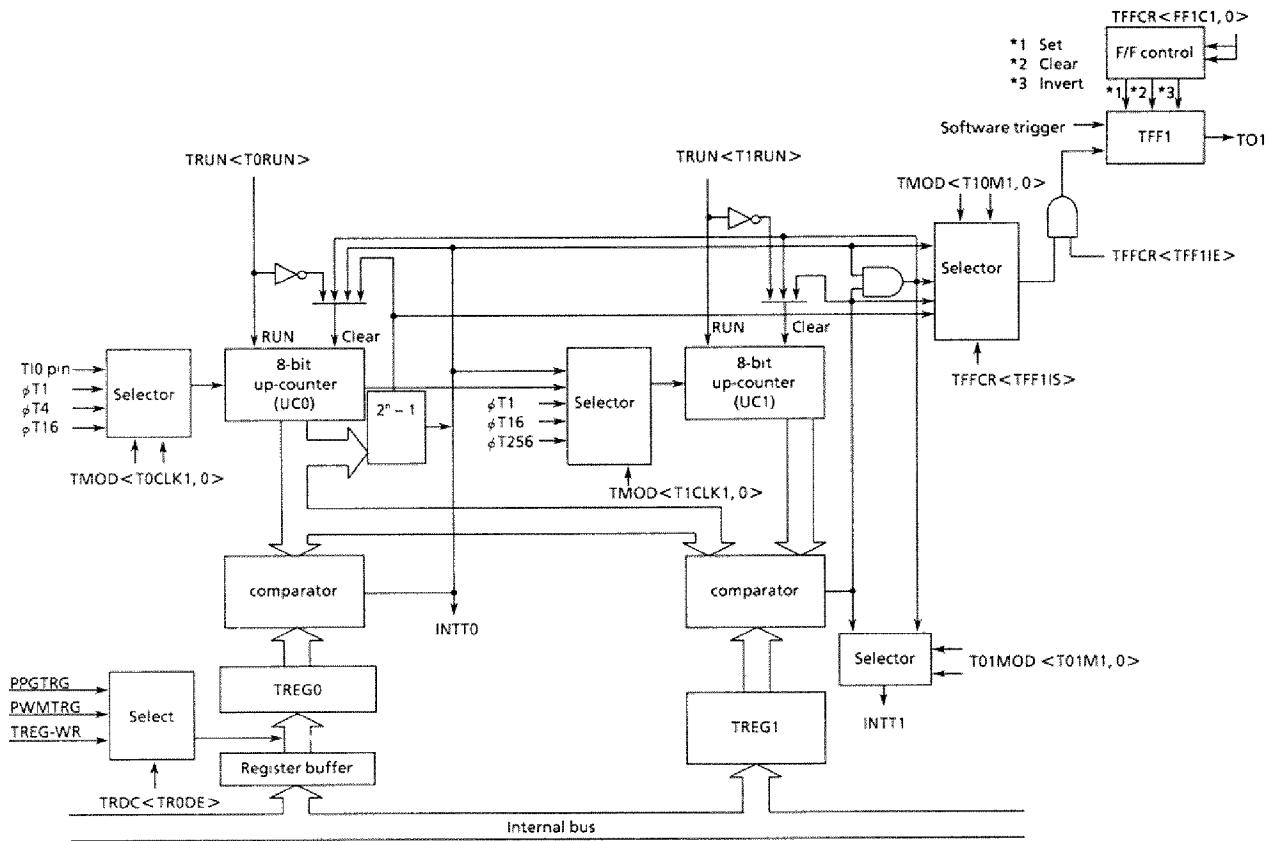


Figure 3.8 (1). Block Diagram of 8-bit Timers (Timers 0 and 1)

### ① Prescaler

These are 9 bit prescaler and prescaler clock selection register to generates input clock for 8-bit Timer 0/1, Timer 4/5 and Serial Interface 0/1.

The 8-bit Timer 0, uses 4 types of clock:  $\phi T1$ ,  $\phi T4$ ,

$\phi T16$  and  $\phi T256$  among the prescaler output.

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

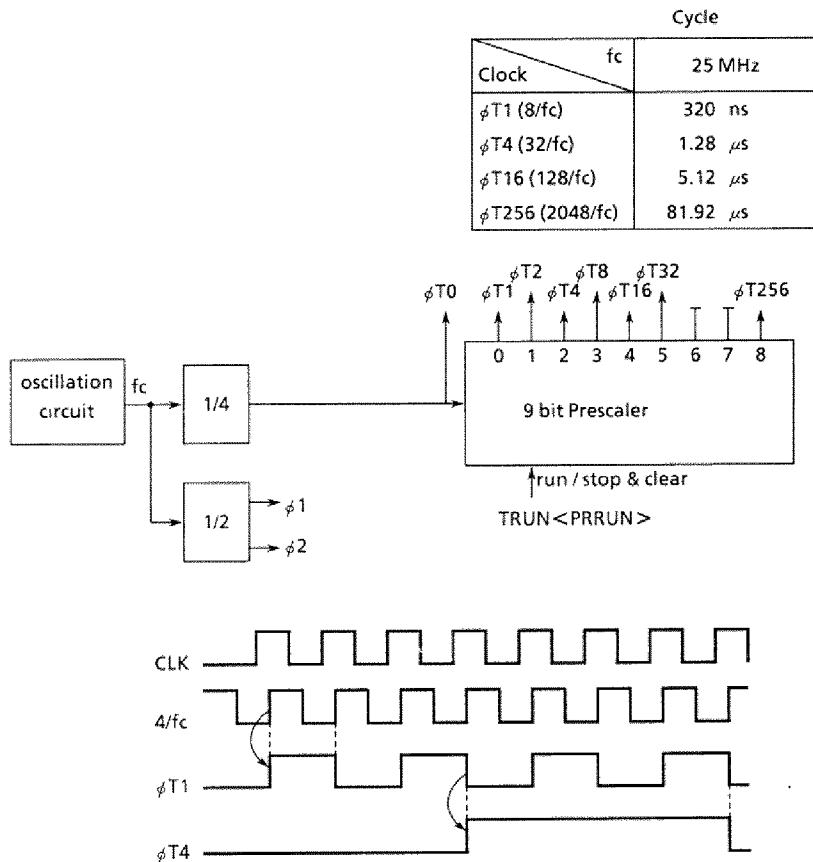


Figure 3.8 (2). Prescaler

## ② Up-counter

There is an 8 bit binary counter which counts up by the input clock pulse specified by the Timer 0/1 mode register T01MOD and Timer 2/3 mode register T23MOD.

The input clocks of timer 0/2 are selected from the three internal clocks  $\phi T_1$ ,  $\phi T_4$ , and  $\phi T_{16}$  and the external clock input (TIO: timer 0 only) using the mode register T01MOD and T23MOD.

The input clocks of timer 1/3 differ depending on the operation mode. When the timers are set to 16 bit timer mode, the overflows output of timer 1/3 are used as the input clock. When the timers are not set to the 16 bit mode, the input clock is selected from the internal clocks  $\phi T_1$ ,  $\phi T_{16}$ , and  $\phi T_{256}$ , and the output comparator (match detection).

- Example:
- When T01MOD <T10M1,0> = 01  
the overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer).
  - When T01MOD7, 6 = 00 and T01MOD3, 2 = 01,  $\phi T_1$  becomes the input of timer 1 (8 bit timer mode).

Operation mode is also set by T01MOD register and T23 MOD register. When reset, it is initialized to T01MOD <T01M1, 0> = 00, T23MOD <T23M1, 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

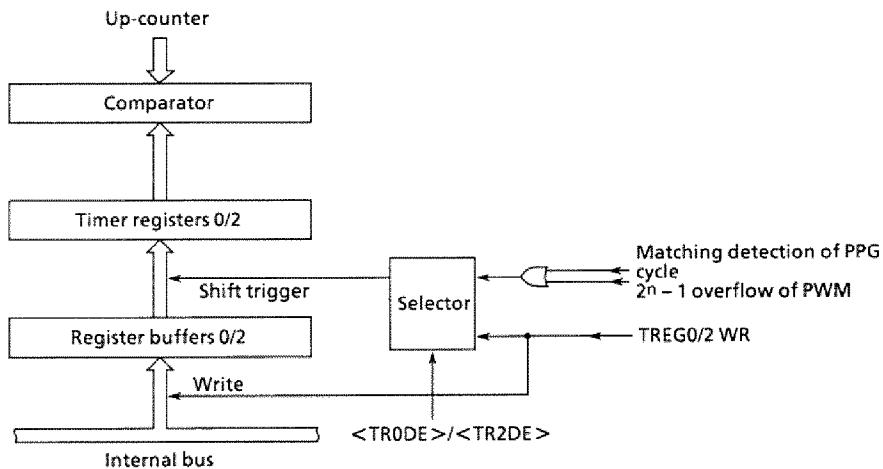
This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, TREG2, TREG3, matches the value of up-counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up-counter overflows.

Timer register TREG0/TREG2 is of double buffer structure, each of which makes a pair with register buffer.

The timer register double buffer register TRDC <TR0DE, TR2DE> bit controls whether the double buffer structure in the TREG0/TREG2 should be enabled or disabled. It is disabled when <TR0DE>/<TR2DE> = 0, and enabled when they are set to 1.

In the condition of double buffer state, the data is transformed 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.

When reset, it will be initialized to <TR0DE>/<TR2DE> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <TR0DE>/<TR2DE> to 1, and write the following data in the register buffer.



**Figure 3.8 (3). Configuration of Timer Register 0/2**

Note: Timer register and the register buffer are allocated to the same memory address. When  $\langle \text{TR0DE} \rangle / \langle \text{TR2DE} \rangle = 0$ , the same value is written in the register buffer as well as the timer register, while when  $\langle \text{TR0DE} \rangle / \langle \text{TR2DE} \rangle = 1$  only the register buffer is written.

The memory address of each timer register is as follows.

TREG0: 000022H

TREG1: 000023H

TREG2: 000026H

TREG3: 000027H

All the registers are write-only and cannot be read.

The initial value is indeterminate; when using the 8-bit timer, always write data to the timer register.

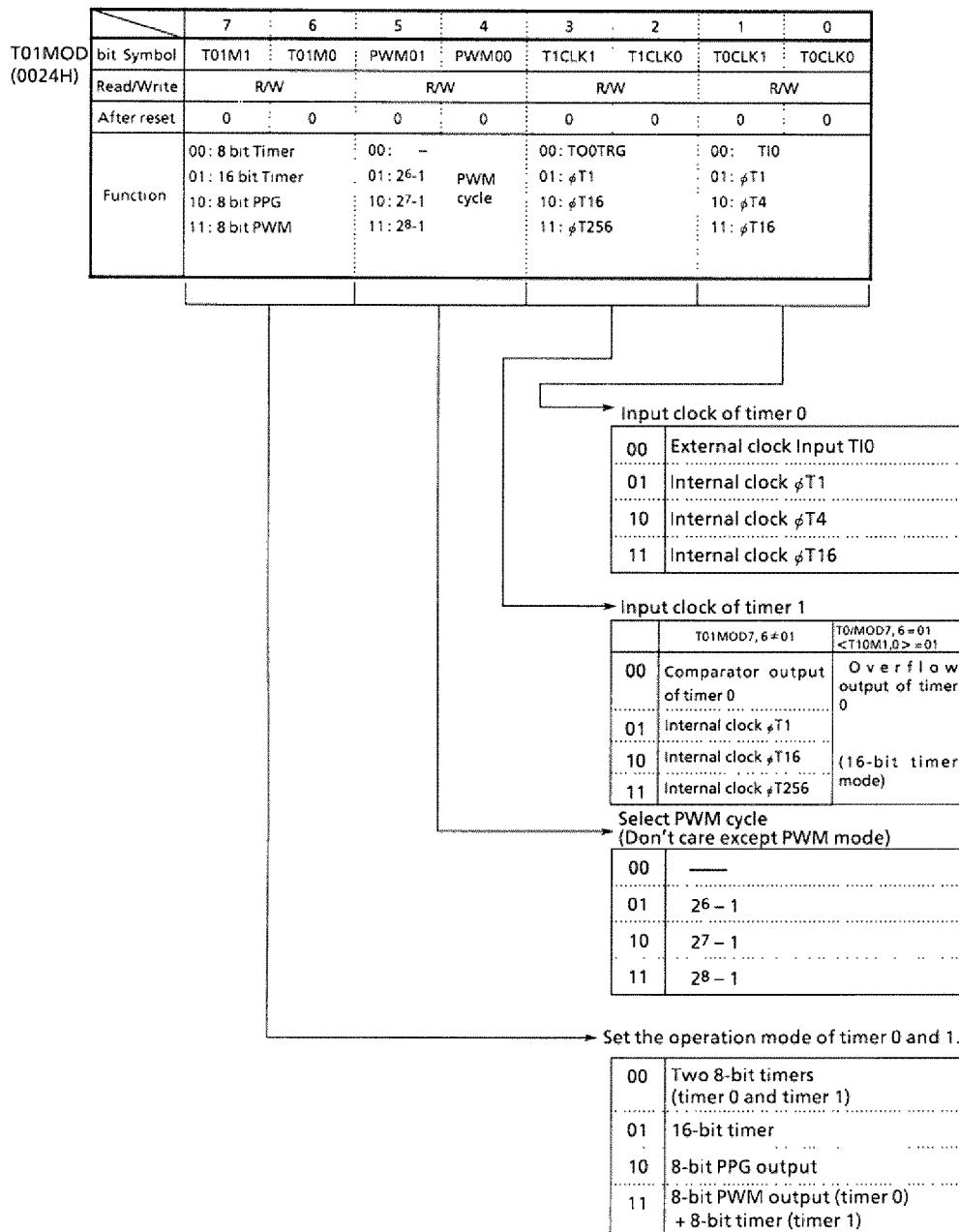


Figure 3.8 (4). Timer 0/1 Mode Control Register (T01MOD)

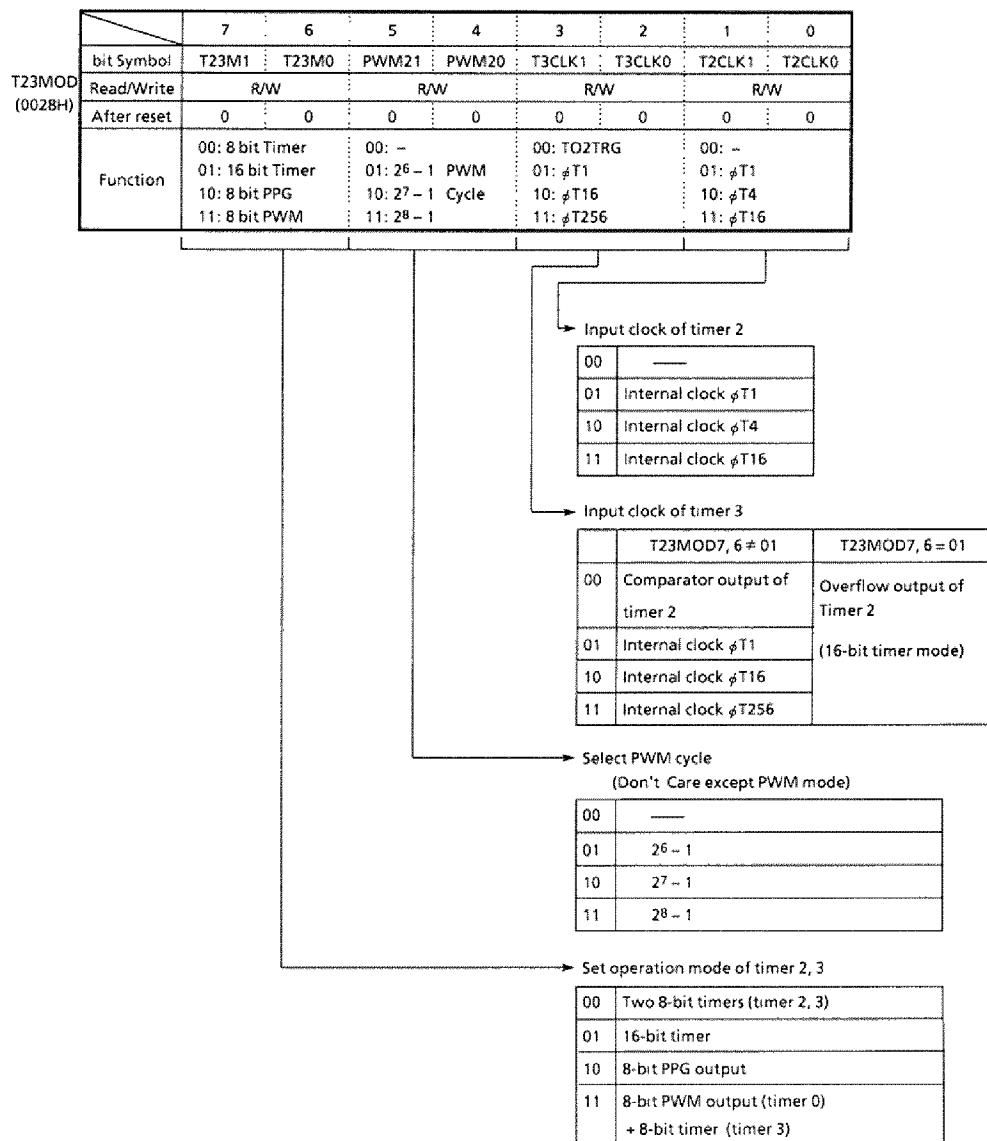
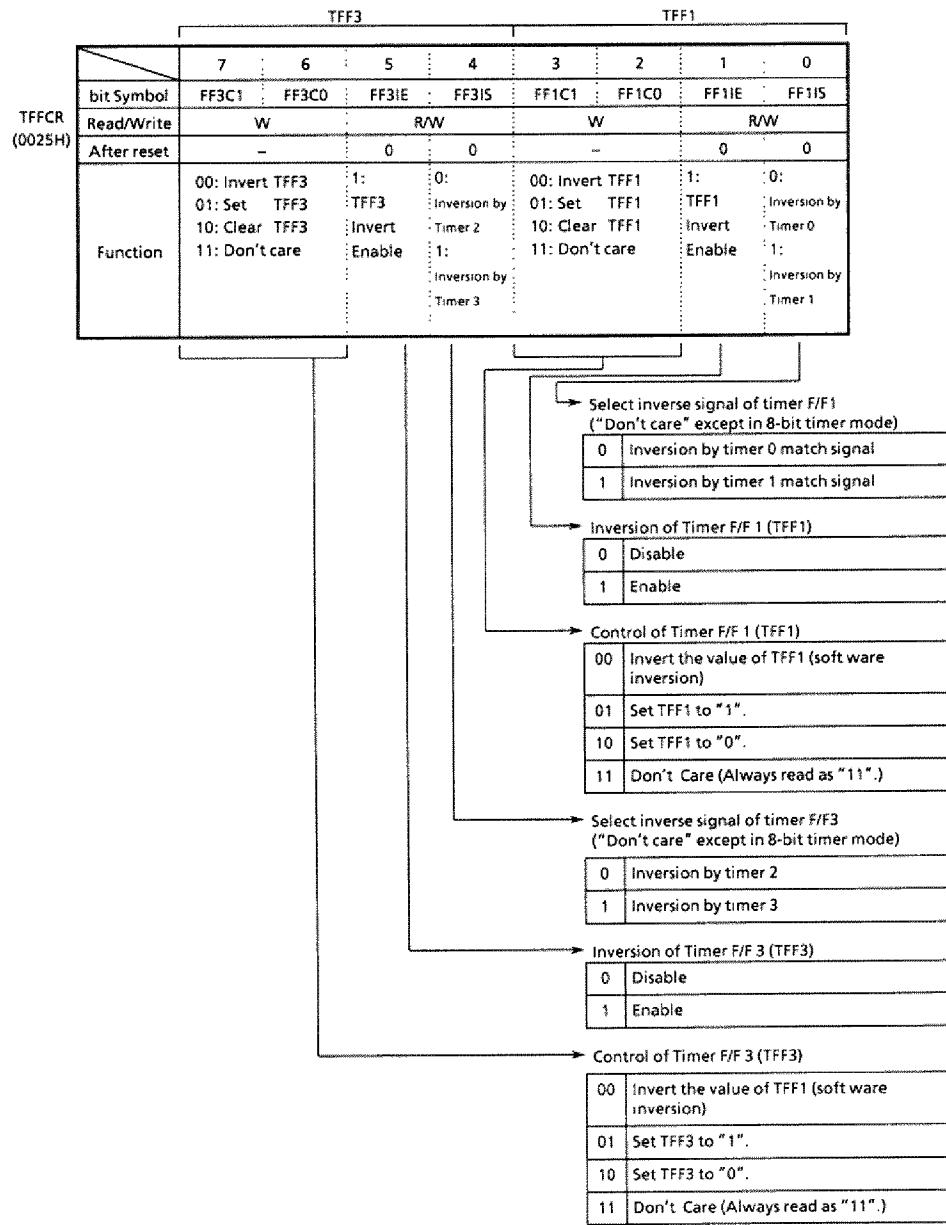
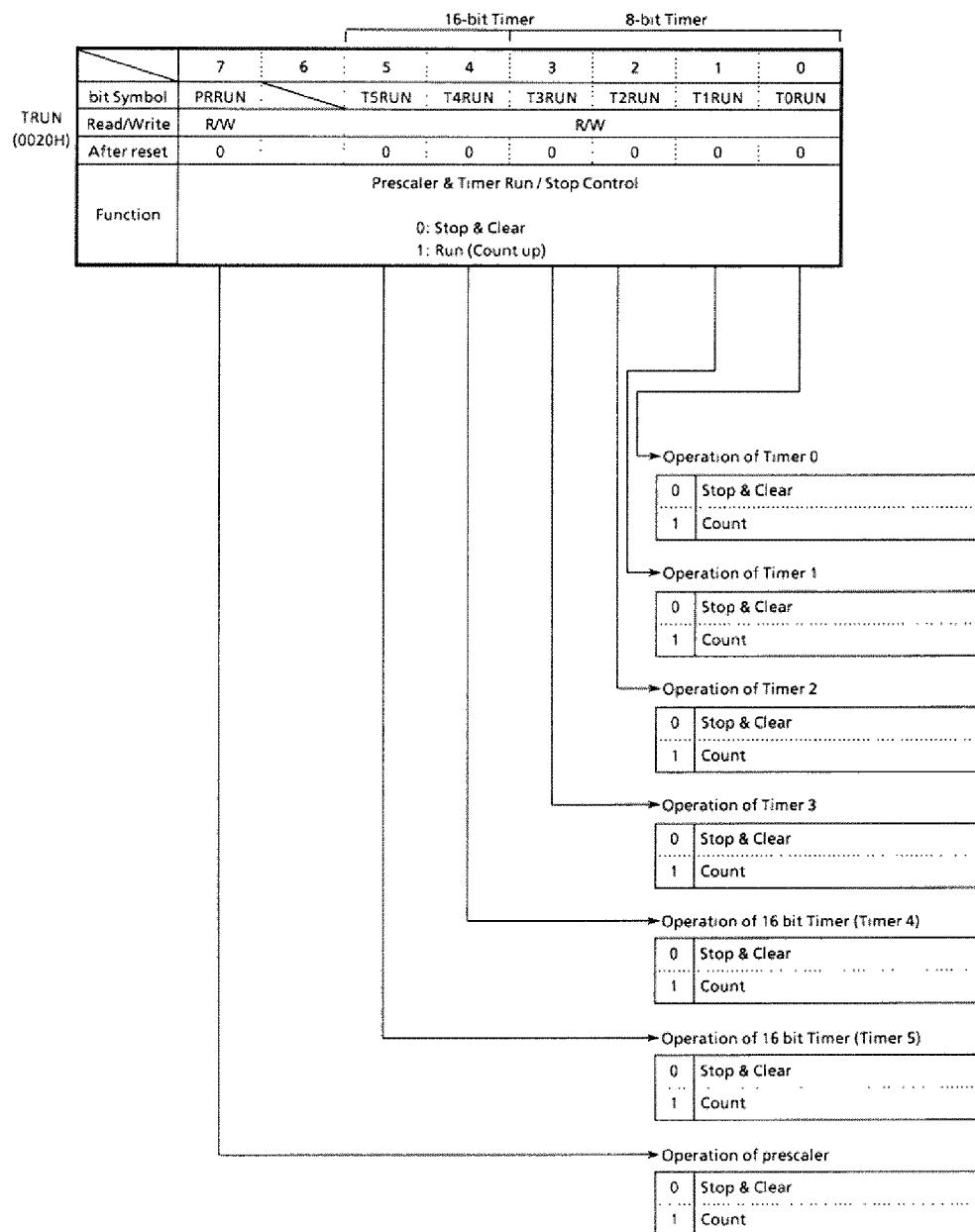


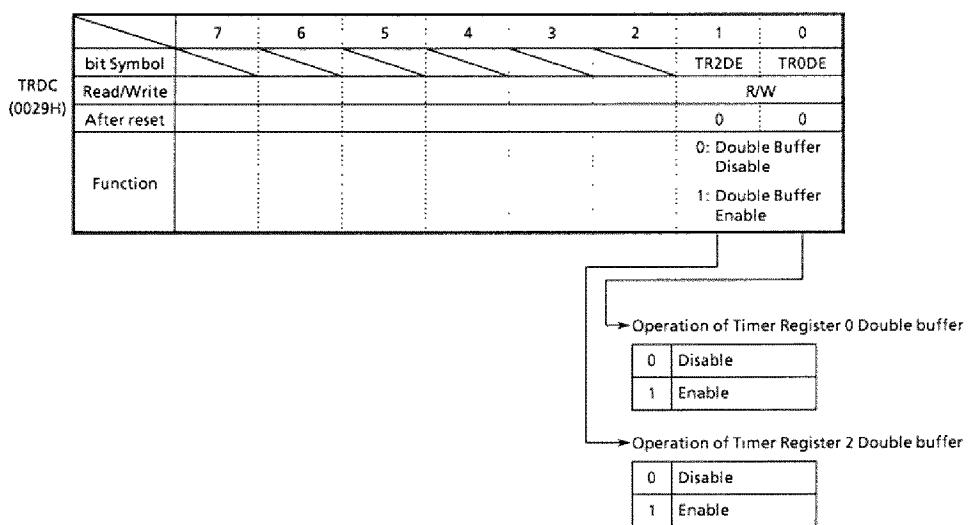
Figure 3.8 (5). Timer 2/3 Mode Register (T23MOD)



**Figure 3.8 (6). 8-Bit Timer Flip-flop Control Register (TFFCR)**



**Figure 3.8 (7). Timer Operation Control Register (TRUN)**



**Figure 3.8 (8). Timer Register Double Buffer Control Register (TRDC)**

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 to 3) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

⑤ Timer flip-flop (timer F/F)

The status of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer and the value can be output to the timer output pins TO1 (also used as PA2) and TO3 (also used as PA3). The timer F/F are provided for a pair of timer 0/1 and timer 2/3. The outputs of timer F/F are TFF1 and TFF3, and output signals through the TO1 and TO3.

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

(1) 8-bit Timer Mode

Four interval timers, 0, 1, 2, and 3, can be used independently as an 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 synchronization to T01MOD and TREG1, respectively. Then, enable interrupt INTT1 and start the counting of timer 1.

Example: To generate timer 1 interrupt every  $32\mu s$  at  $f_c = 25MHz$ , set each register in the following manner.

	MSB	LSB	
	7 6 5 4 3 2 1 0		
TRUN	← - X - - - - 0 -		Stop timer 1, and clear it to "0".
T01MOD	← 0 0 X X 0 1 - -		Set the 8-bit timer mode, and select $\phi T_1 (0.32 \mu s @ f_c = 25 MHz)$ as the input clock.
TREG1	← 0 1 1 0 0 1 0 0		Set the timer register $32 \mu s / \phi T_1 = 100 = 64H$
INTET01	← 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 Table 3.8 (1) for selecting the input clock.

**Table 3.8 (1) Setting the Interrupt Period and Input Clock for 8 Bit Timer**

Input clock	Interrupt period (at $f_c = 25MHz$ )	Resolution
$\phi T_1 (8/f_c)$	$32\mu s$ to $81.92\mu s$	$32\mu s$
$\phi T_4 (32/f_c)$	$12.8\mu s$ to $327.7\mu s$	$1.28\mu s$
$\phi T_{16} (128/f_c)$	$5.12\mu s$ to $1.311ms$	$5.12\mu s$
$\phi T_{256} (2048/f_c)$	$81.92\mu s$ to $20.97ms$	$81.92\mu s$

## ② Generating a 50% duty square wave pulse

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

Example: To output a  $1.92\mu s$  square wave pulse from TO1 pin at  $f_c = 25\text{MHz}$ , set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

<pre> 7 6 5 4 3 2 1 0 TRUN ← - X - - - - 0 - T01MOD← 0 0 X X 0 1 - - TREG1 ← 0 0 0 0 0 0 1 1 TFFCR ← - - - - 1 0 1 1 PACR ← X X X X - 1 - - PAFC ← X X X X - 1 X X TRUN ← 1 X - - - - 1 - </pre>	Stop timer 1, and clear it to "0". Set the 8-bit timer mode, and select $\phi T1$ as the input clock. Set the timer register at $1.92\mu s \div \phi T1 \div 2 = 3$ . Clear TFF1 to "0", and set to invert by the match detect signal from timer 1. Select PA2 as TO1 pin. Start timer 1 counting.
--	---

Note: X ; don't care   - ; no change

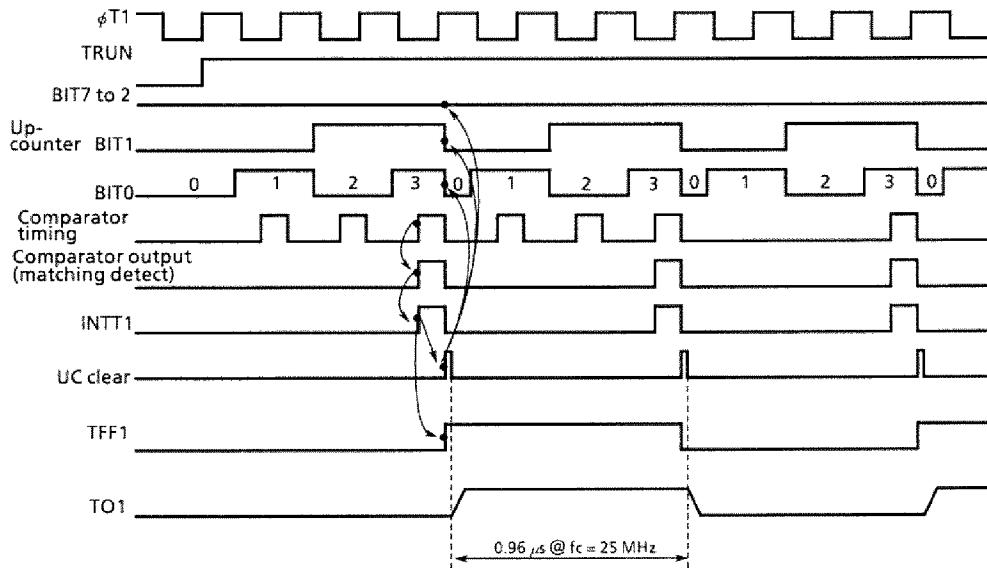
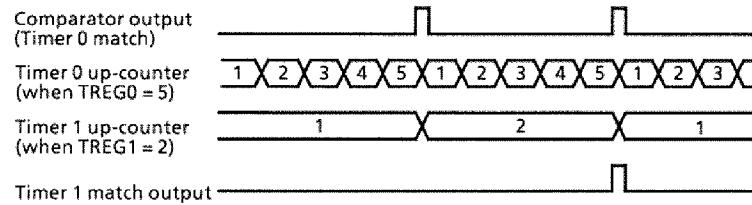


Figure 3.8 (9). Square Wave (50% Duty) Output Timing Chart

- 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 3.8 (10). Timer 1 Count Up by Timer 0**

- ④ Output inversion with software

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

Writing 00 into TFFCR <FF1C1, 0> inverts the value of TFF1. Writing 00 into TFFCR <FF3C1, 0> inverts the value of TFF3.

- ⑤ Initial setting of timer flip-flop (TFF)

The value of TFF can be initialized to "0" or "1", independent of timer operation.

For example, write "10" in TFFCR <FF1C1, 0> to clear TFF1 to "0", while write "01" in TFFCR <FF1C1, 0> to set TFF1 to "1".

Note: The value of timer F/F and timer register cannot be read.

## (2) 16-bit timer mode

A 16-bit interval timer is configurated by using the pair of timer 0/1 and timer 2/3.

Timer 2/3 operate as Timer 0/1, so described have about Timer 0/1.

To make a 16-bit interval timer by cascade connection timer 0 and timer 1, set timer 0/timer 1 mode register T01MOD <T01M1, 0> to "0, 1".

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value of clock control register TCLK.

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, which is restarted by writing data into TREG1).

**Table 3.8 (2) Interrupt Period and Input Clock in 16 Bit Timer Mode**

Input clock	Interrupt period (at fc = 25MHz)	Resolution
$\phi T1$ (8/fc)	32μs to 20.971ms	32μs
$\phi T4$ (32/fc)	12.8μs to 83.885ms	1.28μs
$\phi T16$ (128/fc)	5.12μs to 335.539ms	5.12μs

Setting example: To generate an interrupt INTT1 every 0.32 seconds at fc = 25MHz, set the following values for timer registers TREG0 and TREG1.

When counting with input clock of  $\phi T16$  (5.12μs @ 25MHz)

$$0.32s \div 5.12\mu s = 62500 = F424H$$

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, and then the INTT0 is not decremented.

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.

	Timer0			Timer1		
	INT T0	TO1	Compared Value	INT T1	TO1	Compared Value
16 bit Timer mode <i>(Input overflow of Timer 0 to Timer 1)</i>	not generate the interrupt	output disable	TREG0 <i>(Continued to count up after match)</i>	generate the interrupt	output enable	TREG1+2 <sup>8</sup> + TREG0
8 bit Timer mode <i>(input match of Timer 0 to Timer 1)</i>	generate the interrupt	output enable <i>(Timer 0 or Timer 1)</i>	TREG0 <i>(Cleared after match)</i>	generate the interrupt	output enable <i>(Timer 0 or Timer 1)</i>	TREG1+ TREG0

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

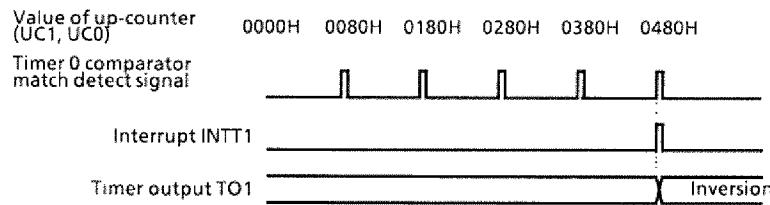


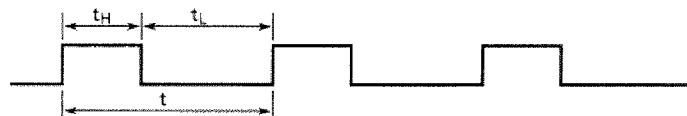
Figure 3.8 (11). Timer Output by 16-Bit Timer Mode

### (3) 8-bit PPG (Programmable Pulse Generation) Mode

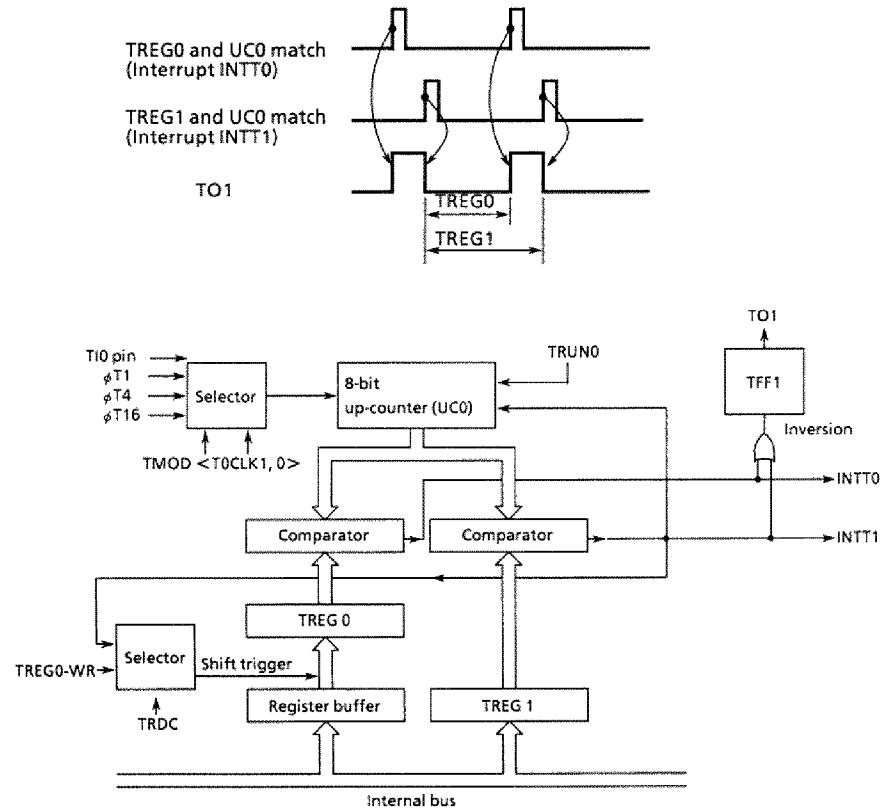
Square wave pulse can be generated at any frequency and duty by timer 0 and timer 2. The output pulse may

be either low-active or high-active. In this mode, timer 1 and timer 3 cannot be used.

Timer 0 outputs pulse through TO1 pin (also used as PA2). Timer 2 outputs pulse to TO3 pin (also used as PA3).



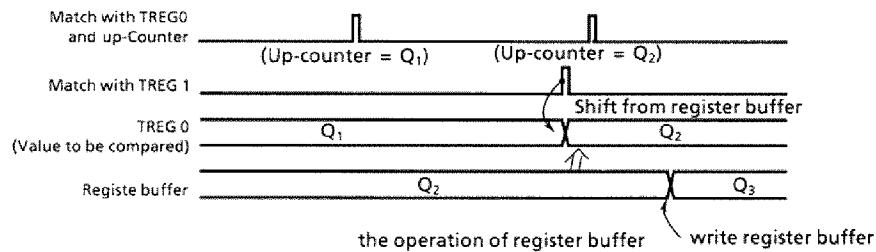
As an example, Timer 0 will be explained below. Timer 2 provides the same functions.



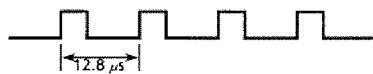
**Figure 3.8 (12). Block Diagram of 8-bit PPG Output Mode**

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

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



Example: Generating 1/4 duty 78.125kHz pulse (at  $f_c = 25\text{MHz}$ )



- Calculate the value to be set for timer register To obtain the frequency 78.125kHz, the pulse cycle  $t$  should be:  $t = 1/78.125\text{kHz} = 12.8\mu\text{s}$

$$\begin{aligned} \text{Given } \varnothing T1 &= 0.32\mu\text{s} \text{ (at 25MHz)}, \\ 12.8\mu\text{s} &\div 0.32\mu\text{s} = 40 \end{aligned}$$

Consequently, to set the timer register 1 (TREG1) to  $TREG1 = 40 = 28H$  and then duty to 1/4,  $t \times 1/4 = 12.8\mu\text{s} \times 1/4 = 3.2\mu\text{s}$

$$3.2\mu\text{s} \div 0.32\mu\text{s} = 10$$

Therefore, set timer register 0 (TREG0) to  $TREG0 = 10 = 0AH$ .

MSB	LSB	
$\leftarrow 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0$		
$\leftarrow TRUN$	$\leftarrow 0\ X\ -\ -\ -\ 0\ 0$	Stop timer 0 and timer 1 and clear it to "0".
$\leftarrow T01MOD$	$\leftarrow 1\ 0\ X\ X\ 0\ 1\ 0\ 1$	Set the 8-bit PPG mode, and select $\varnothing T1$ as input clock.
$\leftarrow TFFCR$	$\leftarrow -\ -\ -\ 0\ 1\ 1\ x$	Sets TFF 1 and enable the inversion and double buffer enable.
	↓	Writting "10" provides negative logic pulse.
$\leftarrow TREG0$	$\leftarrow 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0$	Write "0AH".
$\leftarrow TREG1$	$\leftarrow 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0$	Write "28H".
$\leftarrow PACR$	$\leftarrow X\ X\ X\ X\ ~\ 1\ ~\ ~$	Set PA2 as the T01 pin.
$\leftarrow PAFC$	$\leftarrow X\ X\ X\ X\ -\ 1\ X\ X$	
$\leftarrow TRUN$	$\leftarrow 1\ X\ -\ -\ -\ 1\ 1$	Start timer 0 and Timer 1 counting.

Note: X ; don't care    - ; no change

## (4) 8-bit PWM (Pulse Width Modulation) Mode

This mode is valid only for timer 0/2. In this mode, 2-8 bit resolution of PWM pulse can be output. PWM pulse is output through TO1 pin when using timer 0. When using timer 2, the pulse is through TO3 pin. Timer 1 and timer 3 are valid for 8-bit timers.

As an example, the PWM mode operation of Timer 0 will be explained below. Timer 2 provides the same operation as Timer 0.

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 T01MOD <PWM01, 0>) counter overflow occurs. Up-counter UC1 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 overflow value of  $2^n - 1$  counter)

(Set value of timer register) ≠ 0

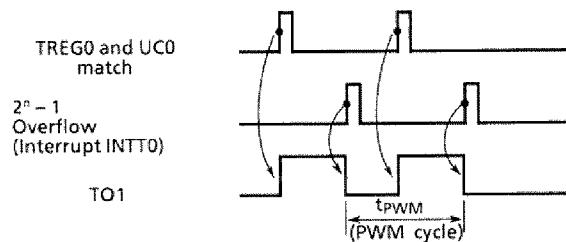


Figure 3.8 (13) shows the block diagram of this mode.

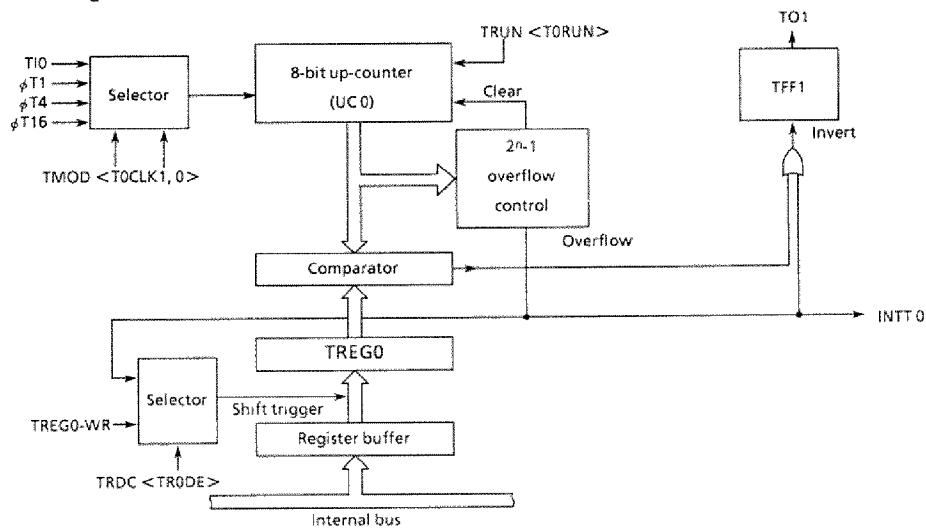
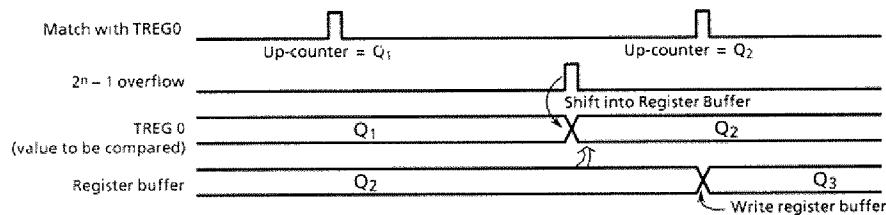


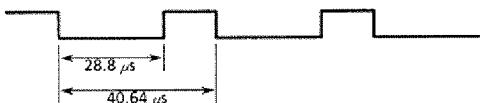
Figure 3.8 (13). Block Diagram of 8-Bit PWM Mode

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



Example: To output the following PWM waves to TO1 pin at  $f_c = 25\text{MHz}$ .



To realize  $40.64\mu\text{s}$  of PWM cycle by  $\phi T1 = 0.32\mu\text{s}$  (at  $f_c = 25\text{MHz}$ ),

$$40.64\mu\text{s} \div 0.32\mu\text{s} = 127 = 2^n - 1$$

Consequently,  $n$  should be set to 7.

As the period of low level is  $28.8\mu\text{s}$ , for  $T1 = 0.32\mu\text{s}$ , set the following value for TREG0.

$$28.8\mu\text{s} \div 0.32\mu\text{s} = 90 = 5AH$$

	MSB	LSB	
	7 6 5 4 3 2 1 0		
TRUN	← X X - - - - 0		Stop timer 0, and clear it to "0".
T01MOD	← 1 1 1 0 - - 0 1		Set 8-bit PWM mode (cycle: $2^7 - 1$ ) and select $\phi T1$ as the input clock.
TFFCR	← - - - - 1 0 1 X		Clears TFF1, enable the inversion and double buffer.
TREG0	← 0 1 0 1 1 0 1 0		Writes "5AH".
PACR	← X X X X - 1 - -		Set PA2 as the TO1 pin.
PAFC	← X X X X - 1 X X		
TRUN	← 1 X - - - - 1		Start timer 0 counting.
Note : X; don't care    - ; no change			

Table 3.8 (3) PWM Cycle and Selection of  $2^n - 1$  Counter

	PWM cycle (@ $f_c = 25\text{MHz}$ )		
	$\phi T1$	$\phi T16$	$\phi T256$
$2^6 - 1$	$20.2\mu\text{s}$	$80.6\mu\text{s}$ (12.4kHz)	$322.6\mu\text{s}$ (3.1kHz)
$2^7 - 1$	$40.6\mu\text{s}$	$162.6\mu\text{s}$ (6.2kHz)	$650.2\mu\text{s}$ (1.5kHz)
$2^8 - 1$	$81.6\mu\text{s}$	$326.4\mu\text{s}$ (3.1kHz)	$1.31\text{ms}$ (0.8kHz)

- (5) Table 3.8 (4) shows the list of 8-bit timer modes.

**Table 3.8 (4) Selection of 8 Bit Timer Mode and Control Register**

Timer mode (8-bit timer x 2 channels)	T01M (T23M)	PWM0 (PWM2)	Upper input T1CLK (T3CLK)	Lower input T0CLK (T2CLK)	Invert select FF1IS (FF31S)
16-bit timer (16-bit) x 1 ch	01	–	–	(External clock $\phi T1, 4, 16$ )	–
8-bit timer (Input of upper timer is output of power one)	00	–	00	(External clock $\phi T1, 4, 16$ )	0: Lower timer 1: Upper timer
8-bit timer x 2 ch	00	–	( $\phi T1, 16, 256$ )	(External clock $\phi T1, 4, 16$ )	0: Lower timer 1: Upper time
8-bit PPG x 1 ch	10	–	–	(External clock $\phi T1, 4, 16$ )	–
8-bit PWM x 1 ch (Lower) 8-bit timer x 1 ch (Upper)	11	PWM cycle	( $\phi T1, 16, 256$ )	(External clock $\phi T1, 4, 16$ )	–

### 3.9 16-bit Timer

The TMP95C061 contains two (timer 4 and timer 5) multifunctional 16-bit timer/event counter with the following operating modes:

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

Timer/event counter consists of 16-bit up-counter, two 16-bit timer registers, two 16-bit capture registers (one of them applies double-buffer), two comparators, capture input controller, and timer flip-flop and the control circuit.

Timer/event counter is controlled by 4 control registers: T4MOD/T5MOD, T4FFCR/T5FFCR, TRUN, and T45CR.

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

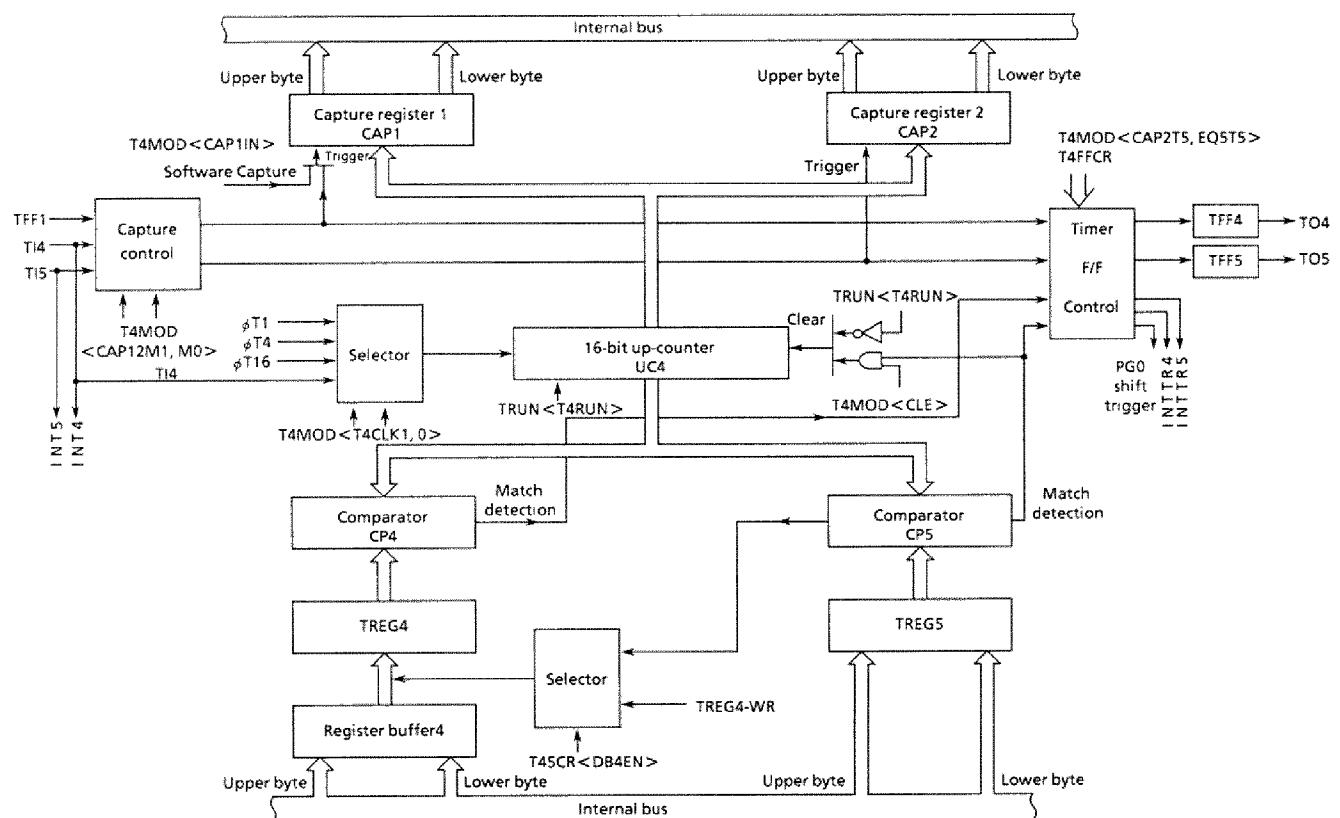


Figure 3.9 (1). Block Diagram of 16-Bit Timer (Timer 4)

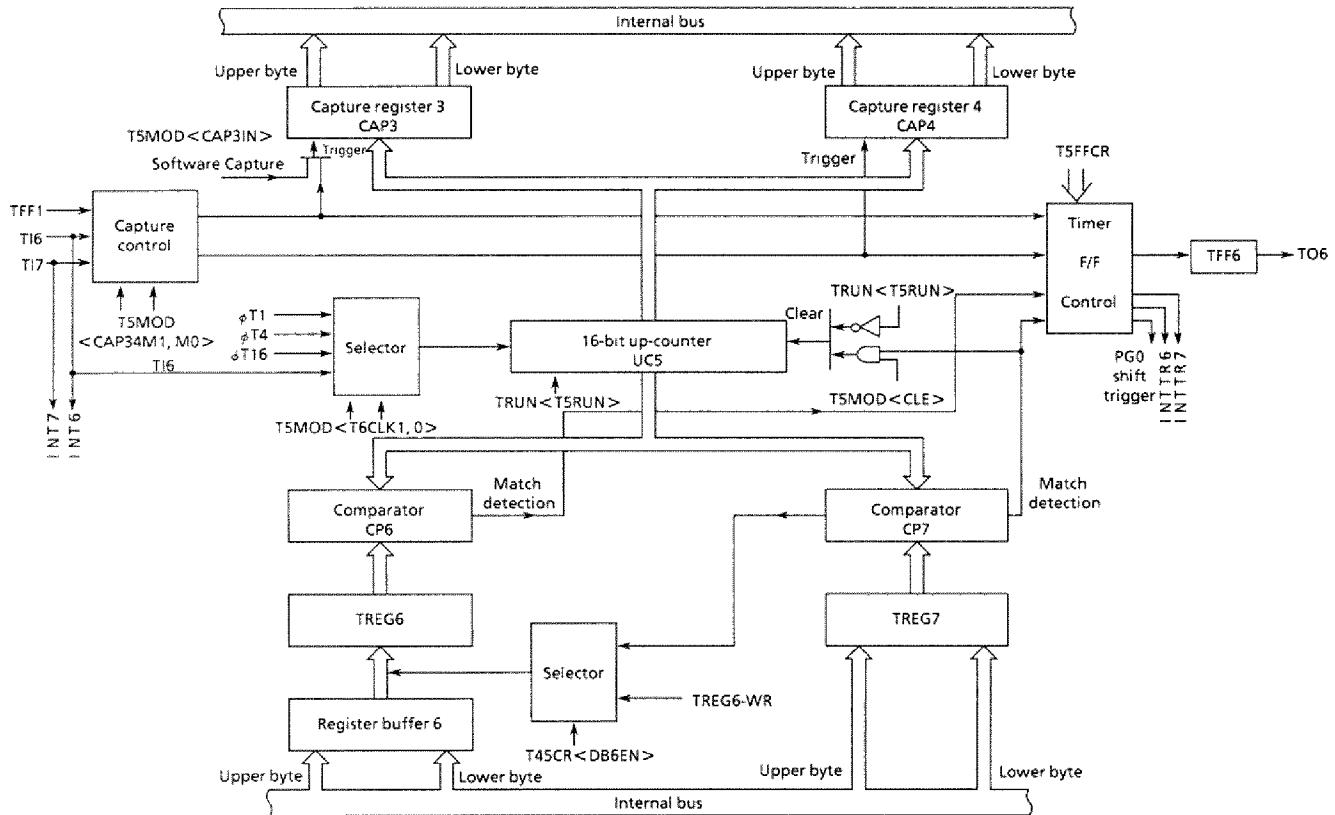


Figure 3.9 (2). Block Diagram of 16-bit Timer (Timer 5)

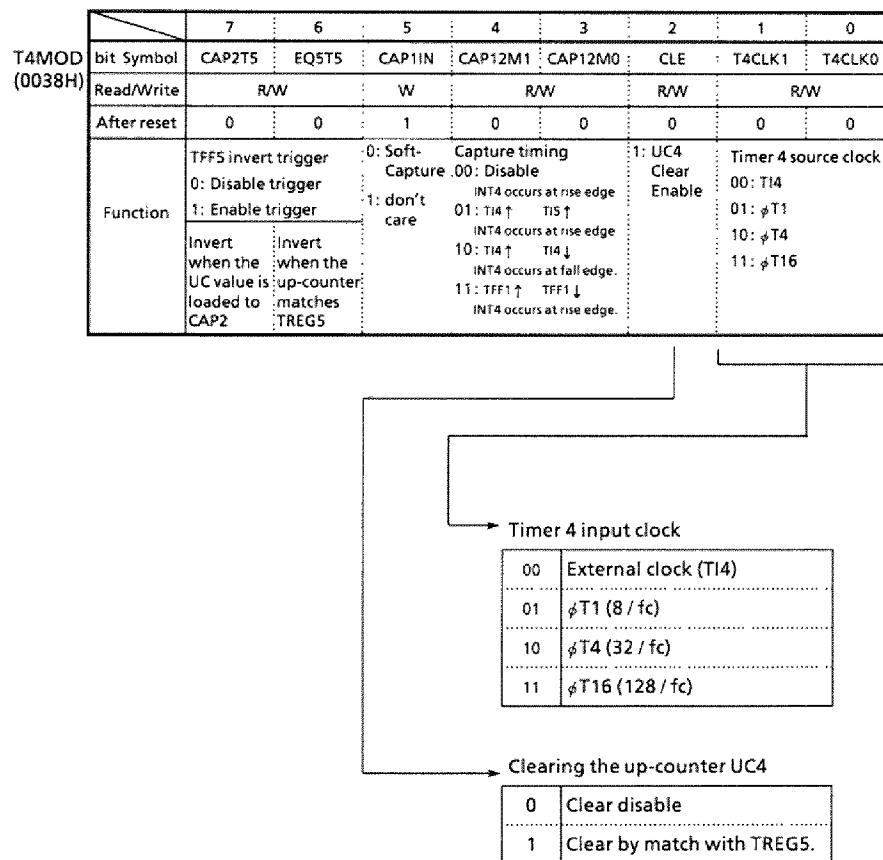
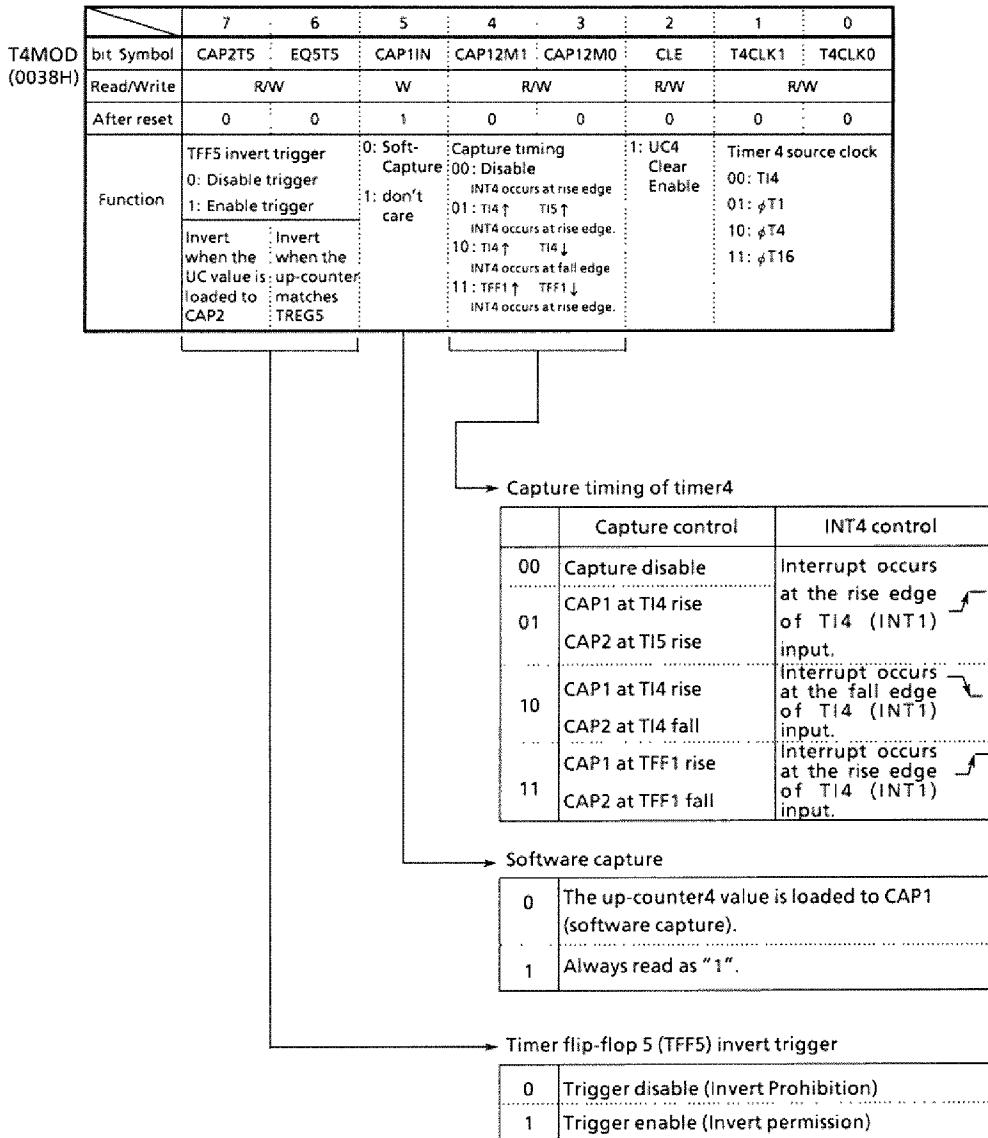


Figure 3.9 (3). 16-Bit Timer Mode Controller Register (T4MOD) (1/2)



CAP2TS : Invert when the up-counter value is loaded to CAP2  
 EQ5TS : Invert when the up-counter matches TREG5

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

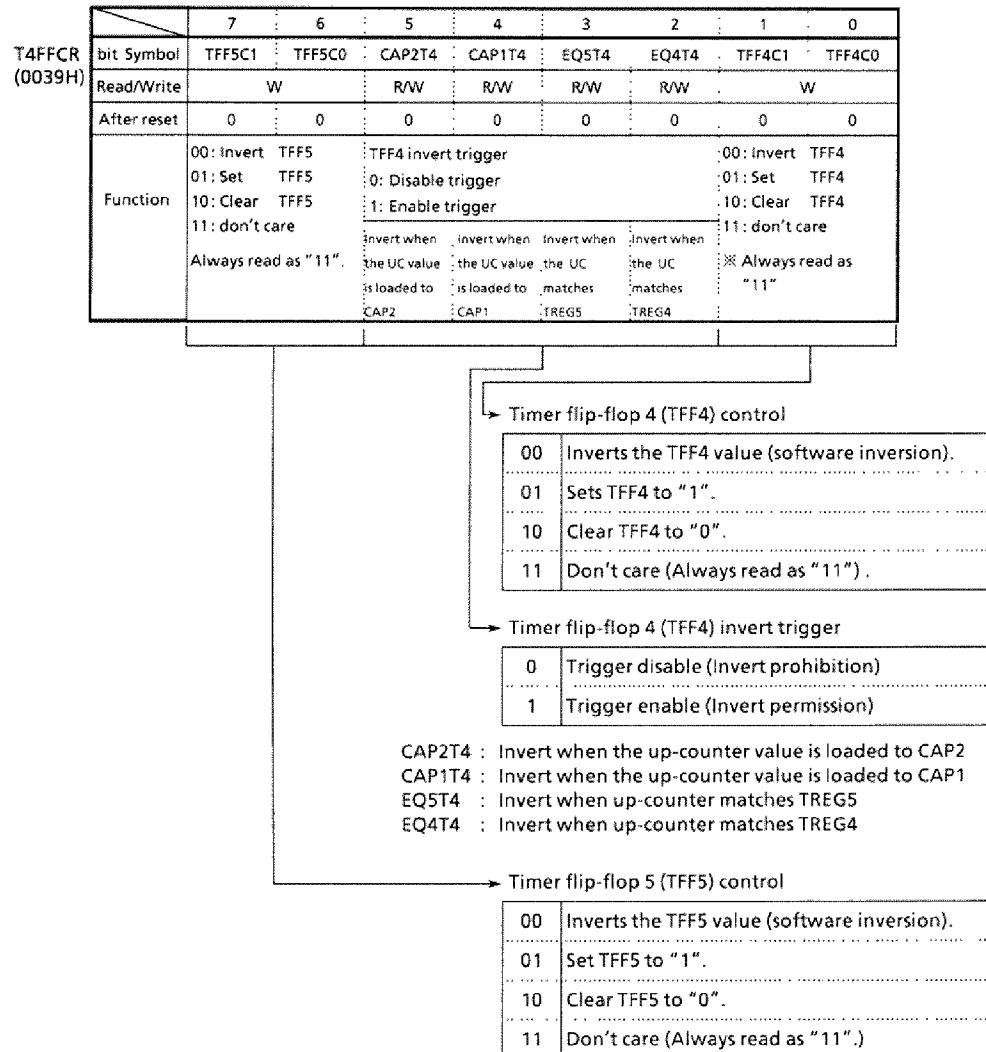


Figure 3.9 (5). 16-Bit Timer 4 F/F Control (T4FFCR)

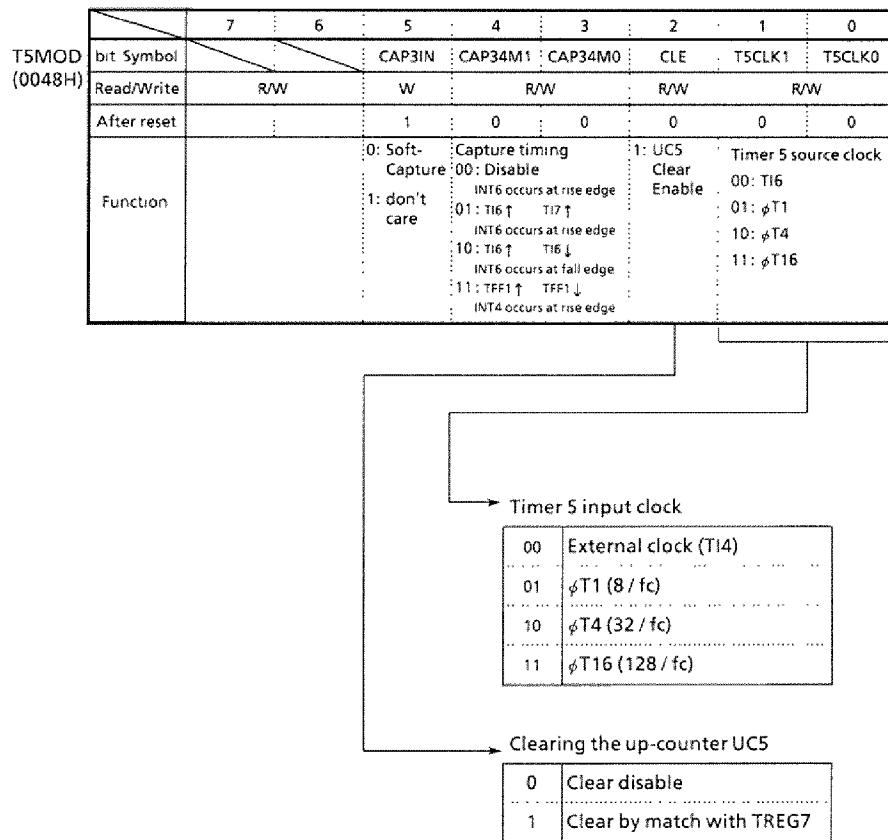


Figure 3.9 (6). 16-Bit Timer Mode Control Register (T5MOD) (1/2)

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

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

T45CR (003AH)	7	6	5	4	3	2	1	0
	bit Symbol	—			PG1T	PG0T	DB6EN	DB4EN
	Read/Write	R/W				R/W		
	After reset	0			0	0	0	0
	Write "0" ※ Always read as "0"				PG1 shift Trigger 0:8 bit	PG0 shift Trigger 0:8 bit	0: Disable 1: Enable	
					3 (Timer 2, 3) 1: 16 bit	1 (Timer 0, 1) 1: 16 bit	Double buffer of TREG6	Double buffer of TREG4
					Timer Trigger (Timer 5)	Timer Trigger (Timer 4)		

→ Double buffer control

0	Disable
1	Enable

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

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

TRUN (0020H)	7	6	5	4	3	2	1	0	
	bit Symbol	PRRUN		TSRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
	Read/Write	R/W				R/W			
	After reset	0		0	0	0	0	0	
	Function				Prescaler & Timer Run / Stop CONTROL				
					0: Stop & Clear				
					1: Run (Count up)				

→ Operation of 16-bit timer (Timer 4)

0	Stop & clear
1	Count

→ Operation of 16-bit timer (Timer 5)

0	Stop & clear
1	Count

→ Operation of prescaler

0	Stop & clear
1	Count

Figure 3.9 (10). Timer Operation Control Register (TRUN)

### ① Up-counter (UC16)

UC16 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD <T4CLK1, 0> register or T5MOD <T5CLK1, 0> register. As the input clock, one of the internal clocks  $\phi T_1$ ,  $\phi T_4$ , and  $\phi T_{16}$  from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as PB0/INT4 pin) and TI67 pin (also used as PB4/INT6 pin) can be selected. When reset, it will be initialized to <T4CLK1, 0>/<T5CLK1, 0> = 00 to select TI4, TI6 input mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN>, <T5RUN>.

When clearing is enabled, up-counter UC4/UC5 will be cleared to zero each time it coincides or matches the

timer register TREG5, TREG7. The “clear enable/disable” is set by T4MOD <CLE> and T5MOD <CLE>. If clearing is disabled, the counter operates as a free-running counter.

### ② Timer Registers

These two 16-bit registers are used to set the value of counter. When the value of up-counter UC4/UC5 matches the set value of this timer register, the comparator match detect signal will be active.

Setting data for timer register (TREG4, TREG5/TREG6 and TREG7) is executed using 2 byte data load instruction or by using 1 byte data load instruction twice for lower 8 bits and upper 1 bits in order.

TREG 4

Upper 8 bits	Lower 8 bits
000031H	000030H

TREG 6

Upper 8 bits	Lower 8 bits
000041H	000040H

TREG 5

Upper 8 bits	Lower 8 bits
000033H	000032H

TREG 7

Upper 8 bits	Lower 8 bits
000043H	000042H

The timer register TREG4/TREG6 make double buffer structure, which are paired with register buffer. The timer control register T45CR <DB4EN, DB6EN> controls whether the double buffer structure should be enabled or disabled. : disabled when <DB4EN, DB6EN> = 0, while enabled when <DB4EN, DB6EN> = 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 UC5) and timer register TREG5 and TREG7.

When reset, it will be initialized to <DB4EN, DB6EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN, DB6EN> = 1, and then write the following data in the register buffer.

TREG4, TREG6 and register buffer are allocated to the same memory addresses 000030H/000031H and 000040H/000041H. When <DB4EN, DB6EN> = 0, the same value will be written in both the timer register and the register buffer. When <DB4EN, DB6EN> = 1, the value is written into only the register buffer.

Since the timer register is indeterminate after a reset, always write data to higher and lower bits.

### ③ Capture Register (CAP1 and CAP2)

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 load instruction or two 1-byte data load instruction, from the lower 8 bits followed by the upper 8 bits.

CAP 1

Upper 8 bits	Lower 8 bits
000035H	000034H

CAP 3

Upper 8 bits	Lower 8 bits
000045H	000044H

CAP 2

Upper 8 bits	Lower 8 bits
000037H	000036H

CAP 4

Upper 8 bits	Lower 8 bits
000047H	000046H

#### ④ Capture Input Control Circuit

This circuit controls the timing to latch the value of up-counter UC4/UC5 into (CAP1, CAP2/CAP3, CAP4). The latch timing of capture register is controlled by register T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0>.

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 00

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

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 01

Data is loaded to CAP1/CAP3 at the rise edge of TI4 pin (also used as PB0/INT7) input, while data is loaded to CAP2/CAP4 at the TI5 pin (also used as P81/INT5) and TI7 pin (also used as PB5/INT7) input. (Time difference measurement)

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 10

Data is loaded to CAP1/CAP3 at the rise edge of the TI4 pin/TI6 pin input, while data is loaded to CAP2/CAP4 at the fall edge. Only in this setting, interrupt INT4/INT6 occurs at fall edge. (Pulse width measurement)

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 11

Data is loaded to CAP1/CAP3 at the rise edge of timer flip-flop TFF1, while to CAP2/CAP4 at the fall edge.

Besides, the value of up-counter can be loaded to capture registers by software. Whenever "0" is written in T4MOD <CAPIN>, T5MOD <CAP3IN>, the current value of up-counter will be loaded to capture register CAP1/CAP3. It is necessary to keep the prescaler in RUN mode (TRUN <PRUN> to be "1").

#### ⑤ Comparator

These are 16-bit comparators which compare the up-

counter UC4/UC5 value with the set value of (TREG4, TREG5, TREG5/TREG6, TREG7) to detect the match. When a match is detected, the comparators generate an interrupt (INTTR4, INTTR5/INTTR6, INTTR7), respectively. The up-counter UC4/UC5 is cleared only when UC4/UC5 matches TREG5/TREG7. (The clearing of up-counter UC4/UC5 can be disabled by setting T4MOD <CLE>/T5MOD <CLE> = 0).

#### ⑥ Timer Flip-flop (TFF4/TFF6)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of the inversion can be set for each element by T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4> /T5FFCR <CAP 4T6, CAP3T6, EQ7T6, EQ6T6>. TFF5/TFF6 will be inverted when "00" is written in T4FFCR < TFF4C1, 0>/T5FFCR < TFF6C1, 0>. Also, it is set to "1" when "10" is written, and cleared to "0" when "10" is written. The value of TFF4 can be output to the timer output pin TO4 (commonly used as PB2)/TO6 (also used as PB6).

#### ⑦ Timer Flip-flop (TFF5)

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

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

#### (1) 16-bit Timer Mode

Timer 4 and Timer 5 can be operated independently. Both can be operated all the same, so, Timer 4 is shown here for the purposes of illustration only.

Generating interrupts at fixed intervals, 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.
INTET45	← 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 disable the capture function. (* = 01, 10, 11)
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 timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4 pin/TI6 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.

The counter counts at the rise edge of TI4 pin/TI6 pin input.

TI4 pin/TI6 pin can also be used as PB0/INT4 and PB4/INT6.

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

<pre> 7 6 5 4 3 2 1 0 TRUN ← - X - 0 - - - -  PBCR ← - - - - - - - 0 INTET45 ← 1 1 0 0 1 0 0 0 T4FFCR ← 1 1 0 0 0 0 1 1 T4MOD ← 0 0 1 0 0 1 0 0 TREG5 ← * * * * * * * * * * * * * * * * TRUN ← 1 X - 1 - - - - </pre>	Stop timer 4. Set P80 to input mode Enable INTTR5 and sets interrupt level 4, while disables INTTR4. Disable trigger. Select TI4 as the input clock. Set the number of counts (16 bits). 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

Timer 4 and Timer 5 can be operated all the same, Timer 4 is used for the purposes of explanation.

The PPG mode is obtained by inversion of the timer

flip-flop TFF4 that is to be enabled by match of the up-counter UC4 with the timer register TREG 4 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)

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

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

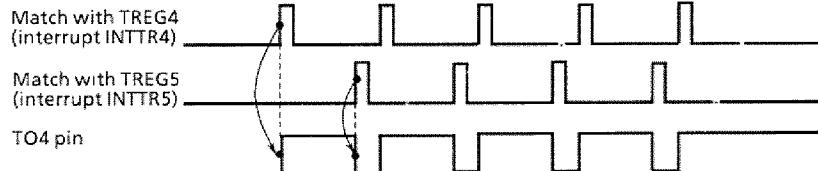
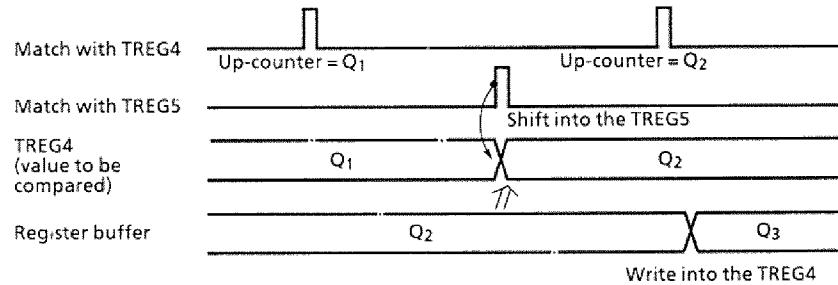


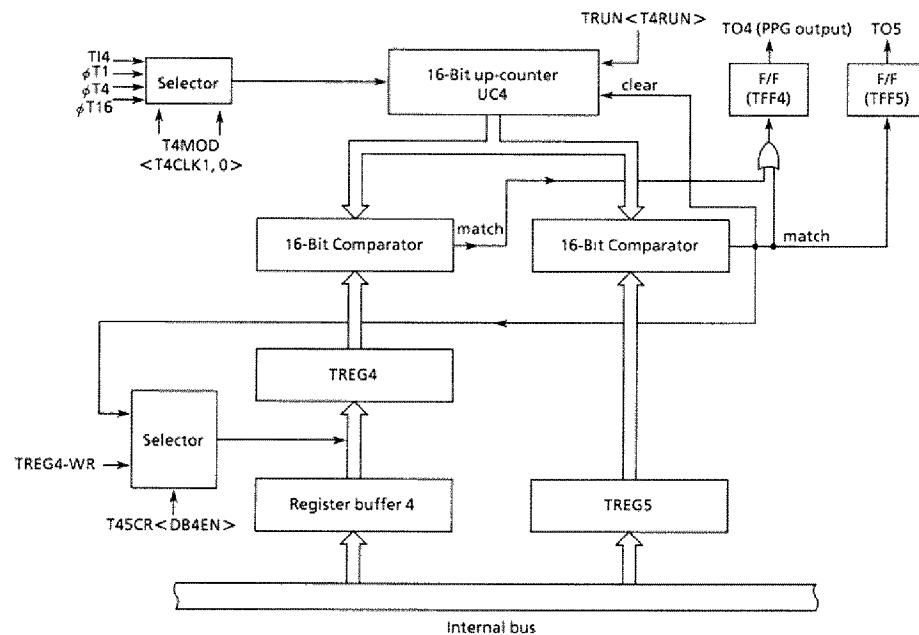
Figure 3.9 (11). Programmable Pulse Generation (PPG) Output Waveforms

When the double buffer of TREG4 is enabled in this mode, the value of register buffer 4 will be shifted in

TREG4 at match with TREG5. This feature makes easy the handling of low duty waves.



**Figure 3.9 (12). Operation of Register Buffer**



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

## (4) Application examples of capture function

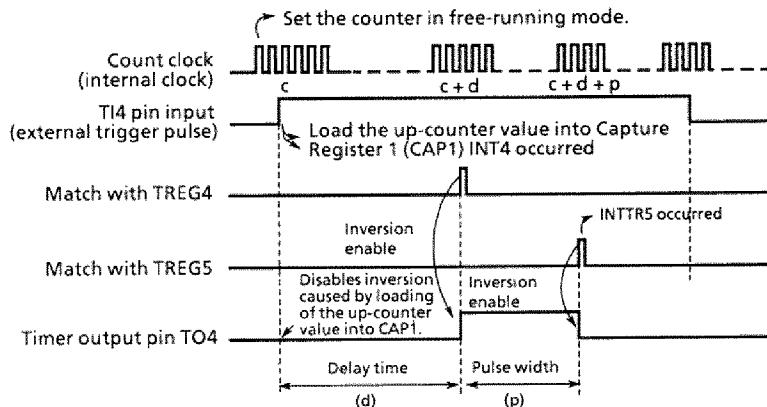
Timer 4 and Timer 5 can be operated all the same, Timer 4 is used for the purposes of explanation  
 The loading of up-counter (UC4) values into the capture registers CAP1 and CAP2, the timer flip-flop TFF4 inversion due to the match detection by comparators CP4 and CP5, and the output of the TFF4 status 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 the capture register CAP1 at the rising edge of TI4 pin. Then set to T4MOD <CAP12M1, 0> = 01.

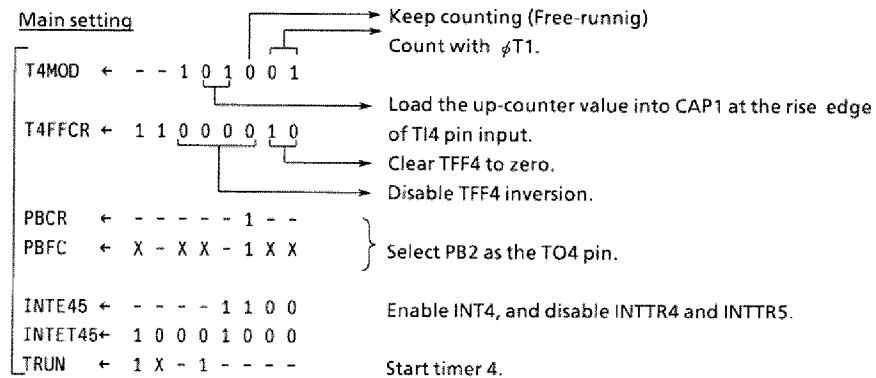
When the interrupt INT4 is generated at the rising edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 ( $= c + d$ ), and set the above set value ( $c + d$ ) pulse a one shot pulse width (p) the TREG5 ( $= c + d + p$ ). When the interrupt INT4 occurs the T4FFCR <EQ5T4, EQ4T4> register should be set that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or 5. When interrupt INTTR5 occurs, this inversion will be disabled.



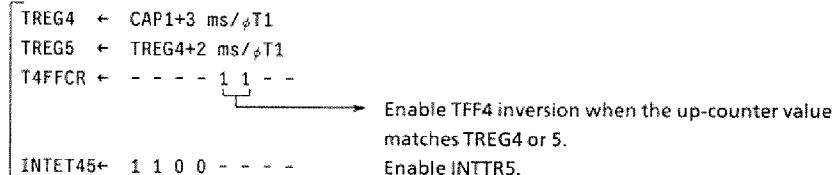
**Figure 3.9 (14). One-Shot Output (with Delay)**

Setting example: To output 2ms one-shot pulse with a

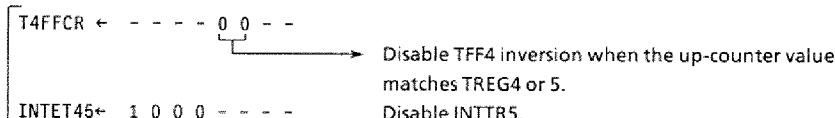
3ms delay to the external trigger pulse to T14 pin.



### Setting of INT4



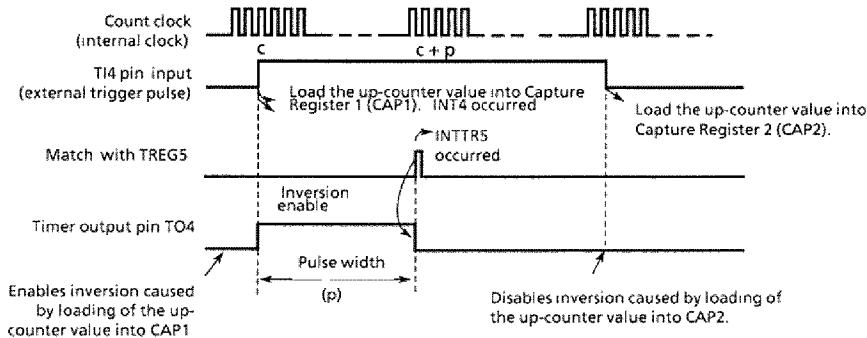
## Setting of INTS



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

When delay time is unnecessary, invert timer flip-flop TFF4 when the up-counter value is loaded into capture register 1 (CAP1), and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt

INT4 occurs. The TFF4 inversion should be enabled before the up-counter (UC4) value matches TREG5, and disabled when generating the interrupt INTTR5.



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

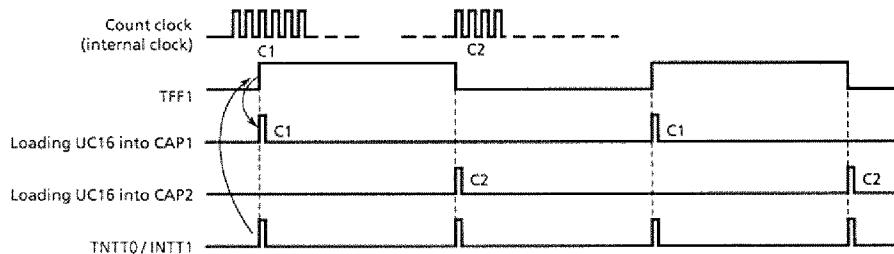
## ② Frequency measurement

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

The TI4 pin input should be selected for the input clock

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

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



**Figure 3.9 (16). Frequency Measurement**

For example, if the value for the level "1" width of TFF1 of the 8-bit timer is set to 0.5 sec. and the difference

between CAP1 and CAP2 is 100, the frequency will be  $100/0.5 [s] = 200 [Hz]$ .

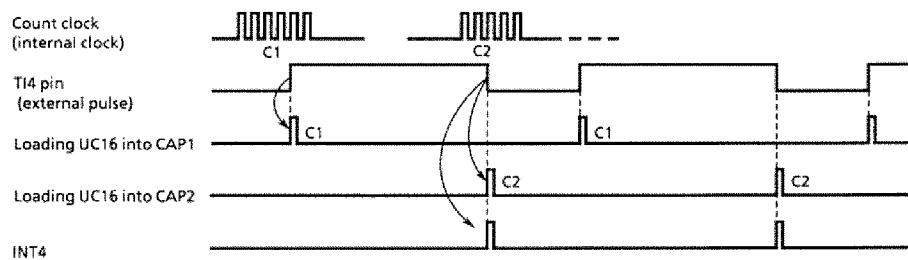
### ③ Pulse width measurement

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 8.0 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 (17) Pulse Width Measurement**

Note: Only in this pulse width measuring mode (T4MOD <CAP12M1, 0> = 10), external interrupt INT1 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 INT1 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 INT1 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT2.

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

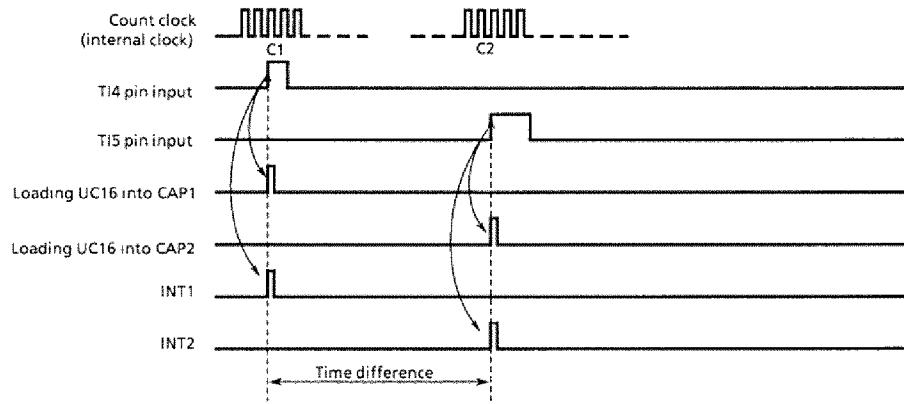


Figure 3.9 (18). Time Difference Measurement

## (5) Different Phased Pulses Output Mode

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

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

This mode can be used only in Timer 4.

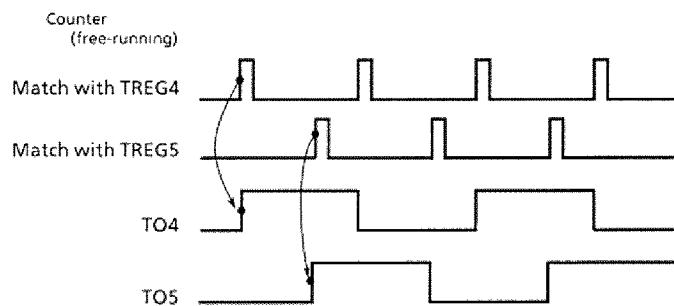


Figure 3.9 (19). Phase Output

Cycles (counter overflow time) of the above output waves are listed on Table 3.9 (2). The following table shows cycles (counter overflow) of the above output wave.

	20MHz	25MHz
$\theta T1$	26.214ms	20.97ms
$\theta T4$	104.856ms	83.88ms
$\theta T16$	419.424ms	335.54ms

### 3.10 Stepping Motor Control/Pattern Generation Port

TMP95C061 contains two channels (PG0 and PG1) of 4-bit hardware stepping motor control/pattern generation (herein after called PG) which actuate in synchronization with the (8-bit/16-bit) timers. The PG (PG0 and PG1) are shared in 8-bit I/O ports P7.

Channel 0 (PG0) is synchronous with 8-bit timer 2 or timer 3, 16-bit timer 5, to update the output.

The PG ports are controlled by control registers (PG01CR) and can select either stepping motor control mode or pattern generation mode. Each bit of the P7 can be used as

the PG port.

PG0 and PG1 can be used independently.

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

Different Points between PG0 and PG1

	<b>PG0</b>	<b>PG1</b>
Trigger Signal	from 8-bit timer 0, 1 or 16-bit timer 4	from 8-bit timer 2, 3 or 16-bit timer 5

Different Points between PG0 and PG1

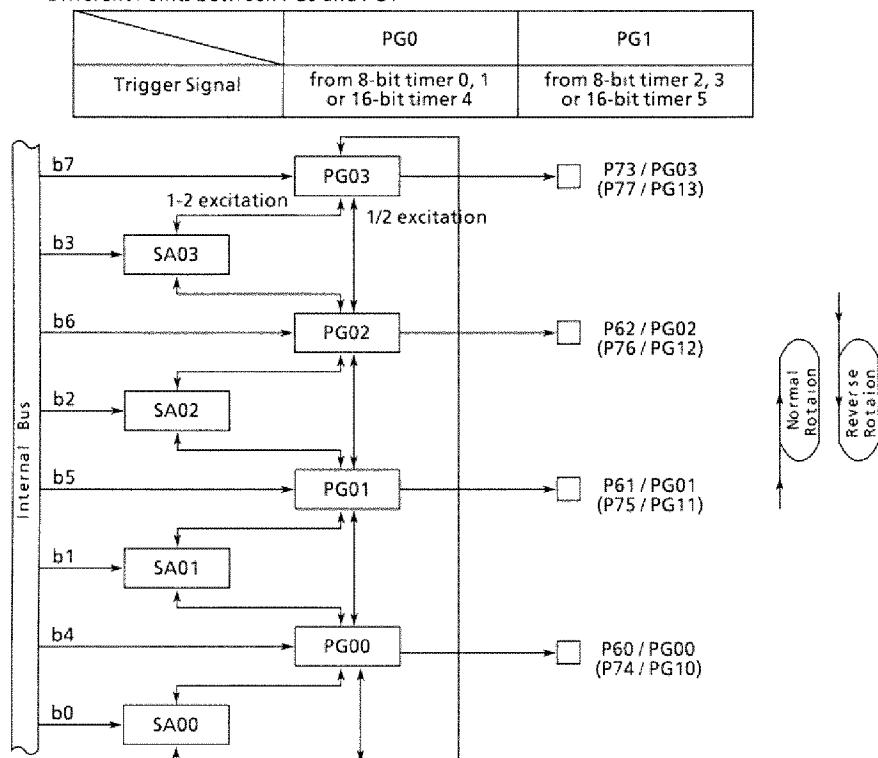


Figure 3.10 (1). PG Block Diagram

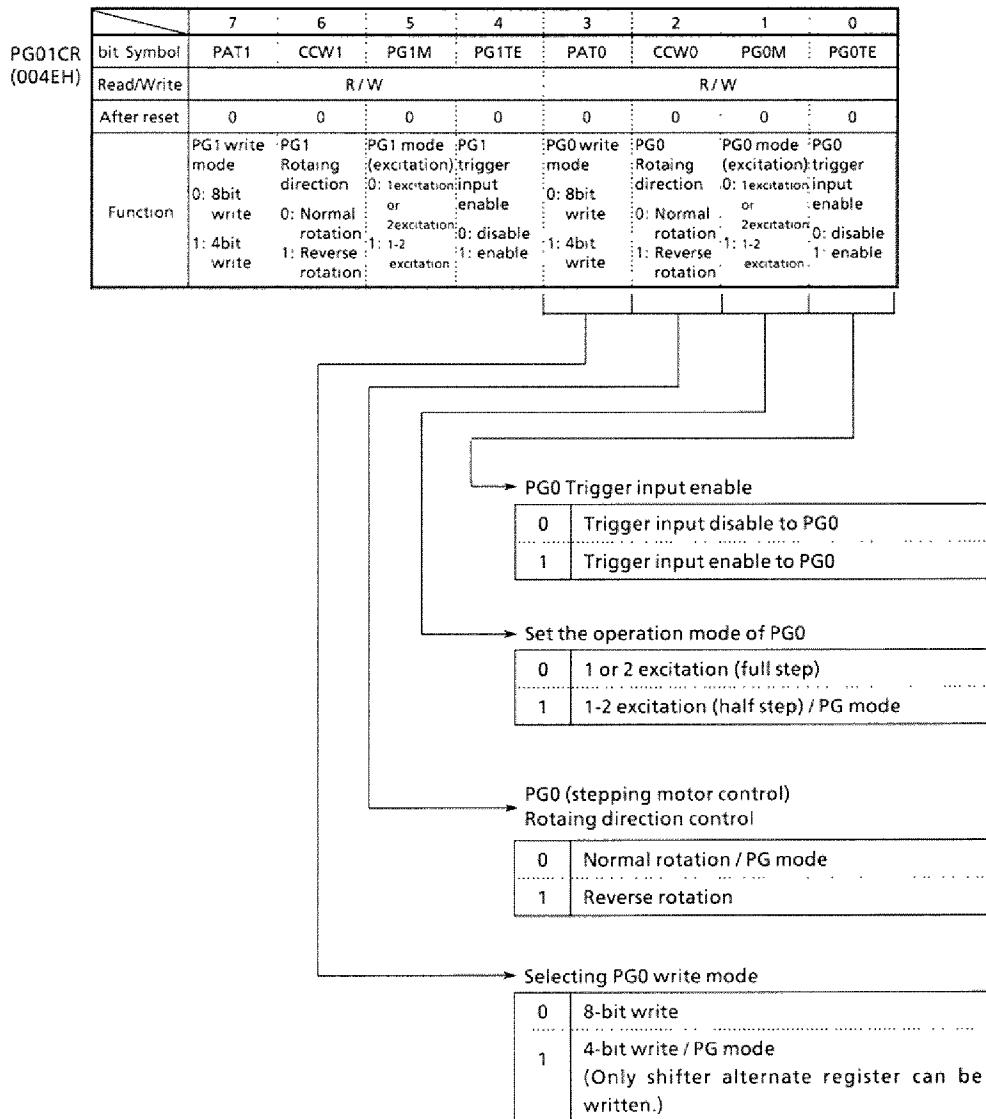


Figure 3.10 (2a). Pattern Generation Control Register (PG01CR)

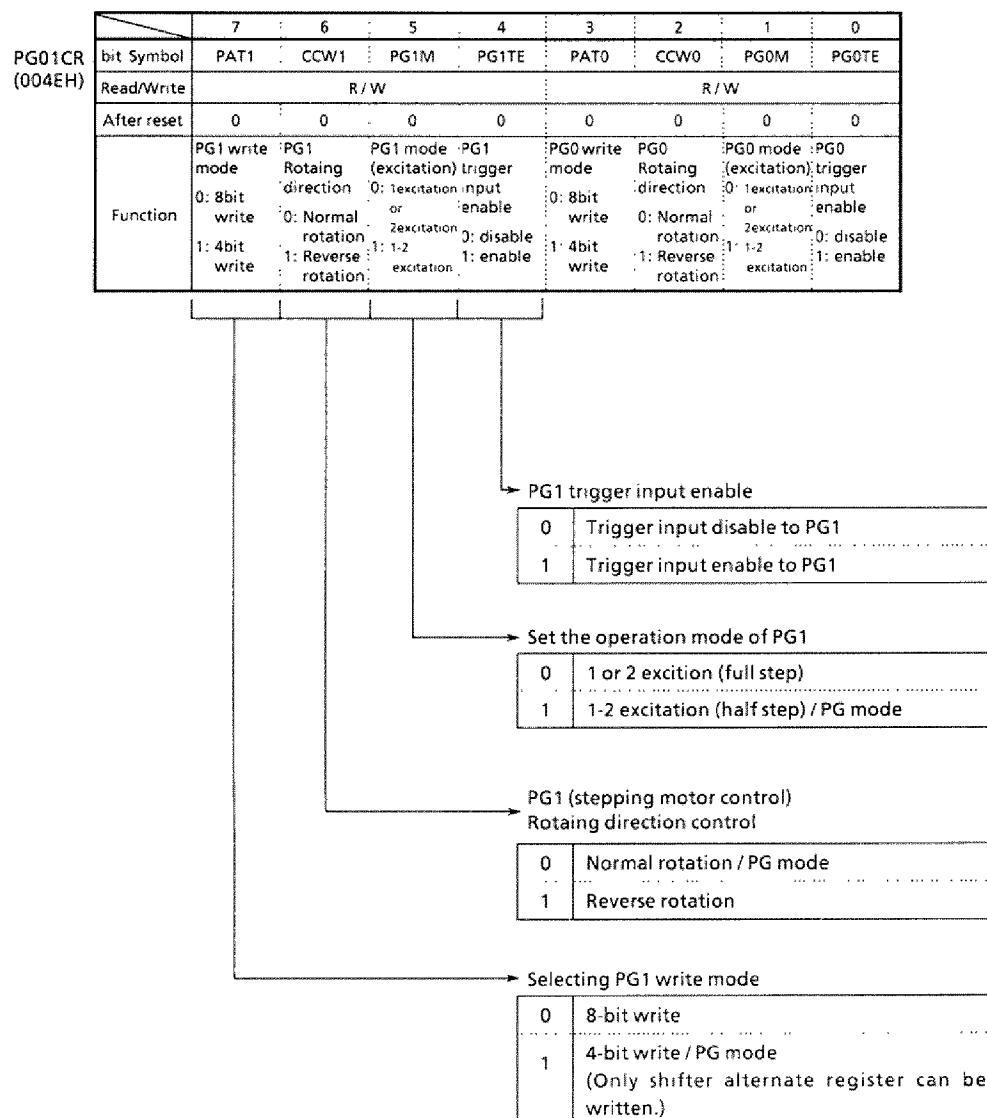


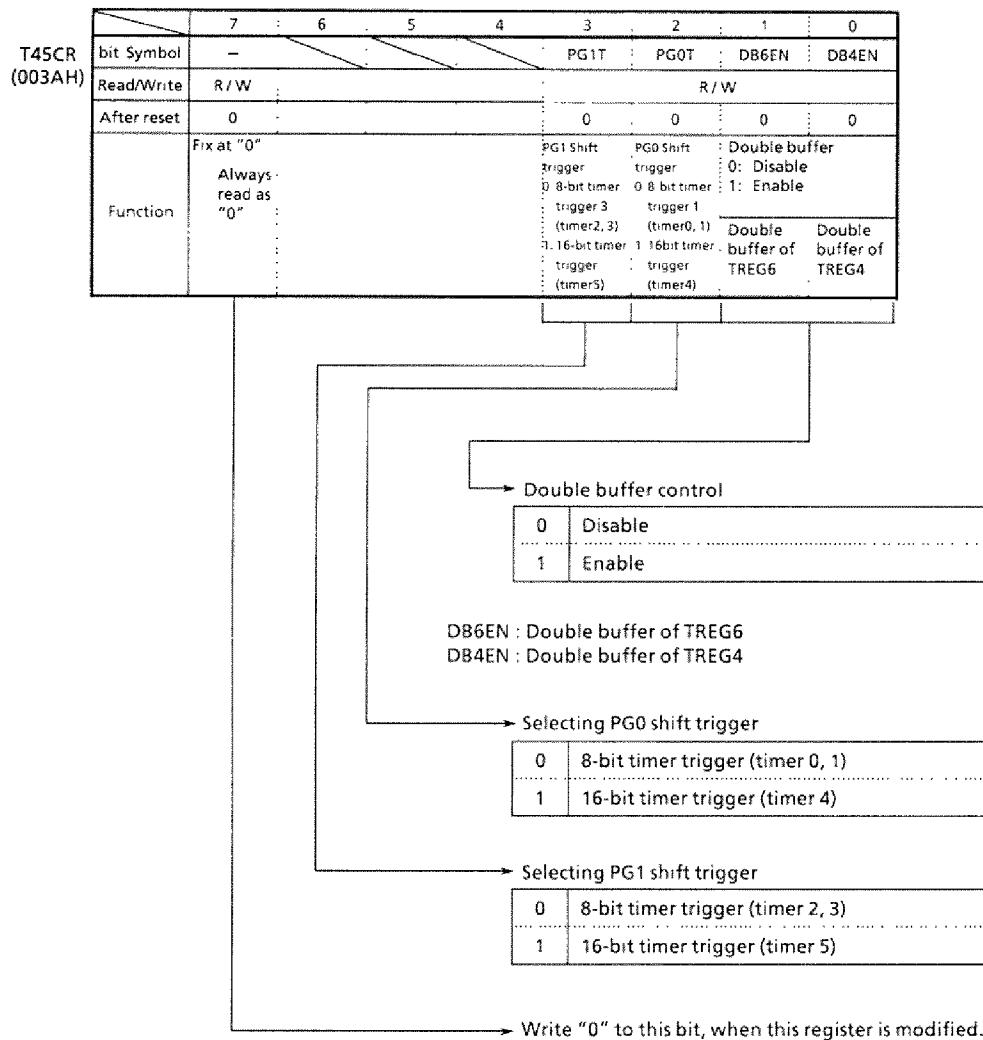
Figure 3.10 (2b). Pattern Generation Control Register (PG01CR)

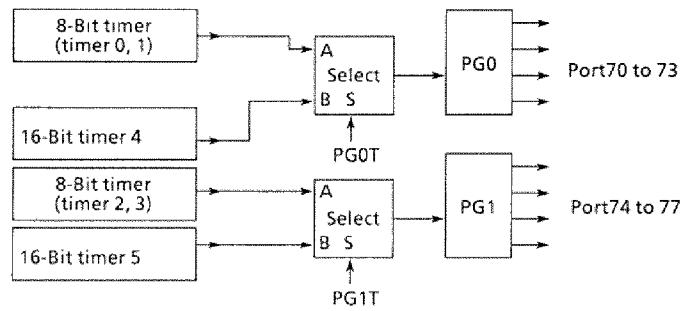
PG0REG (004CH)	7	6	5	4	3	2	1	0	
	bit Symbol	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
	Read/Write	W				R / W			
	After reset	0	0	0	0	Undefined			
		Pattern Generation 0 (PG0) output latch register				Shift alternate register 0 For the PG mode (4-bit write) register			
Function		(Reading the P7 that is set to the PG port allows to read-out.)							
Prohibit Read modify write									

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

PG1REG (004DH)	7	6	5	4	3	2	1	0	
	bit Symbol	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
	Read/Write	W				R / W			
	After reset	0	0	0	0	Undefined			
		Pattern Generation 1 (PG1) output latch register				Shift alternate register 1 For the PG mode (4-bit write) register			
Function		(Reading the P7 that is set to the PG port allows to read-out.)							
Prohibit Read modify write									

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

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



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

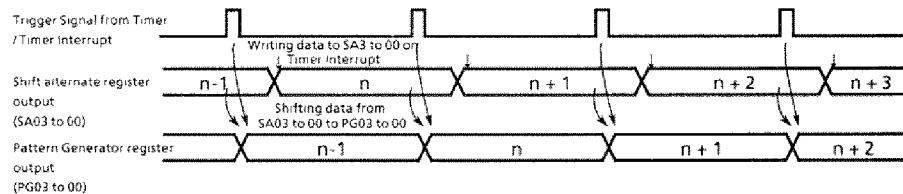
### (1) Pattern Generation Mode

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

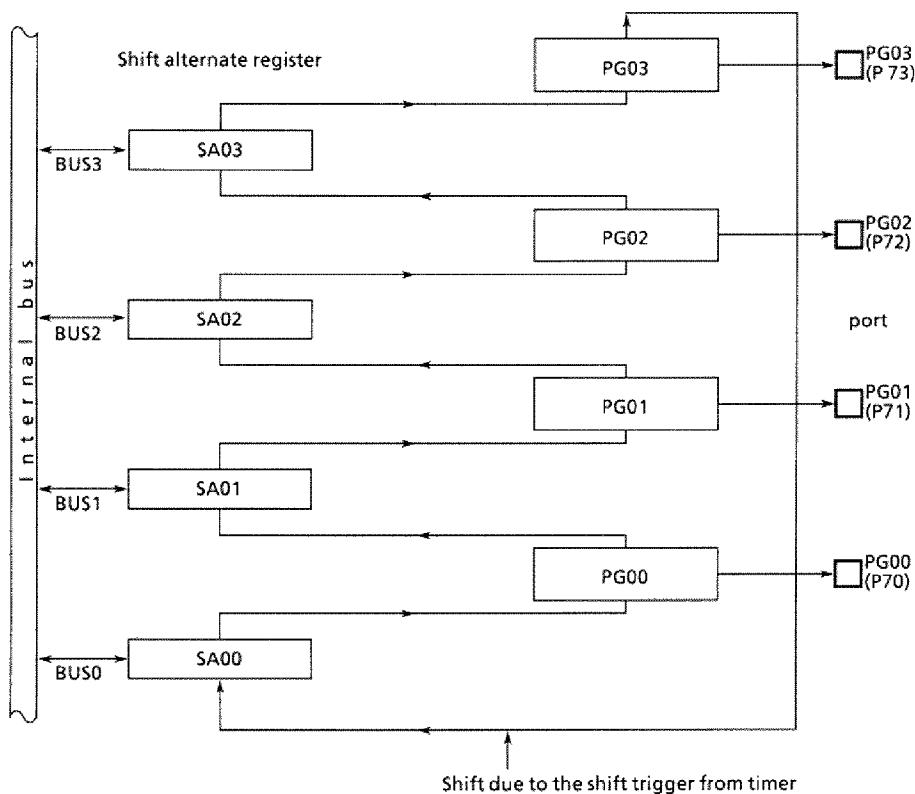
In this mode, set PG01CR <PG0M> and <PG1M> to 1, and PG01CR <CCW0> and <CCW1> to 0.

The output of this pattern generator is output to port 7; since port and functions can be switched on a bit basis using port 7 function control register P7FC, any port pin can be assigned to pattern generator output.

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



Example of pattern generation mode



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

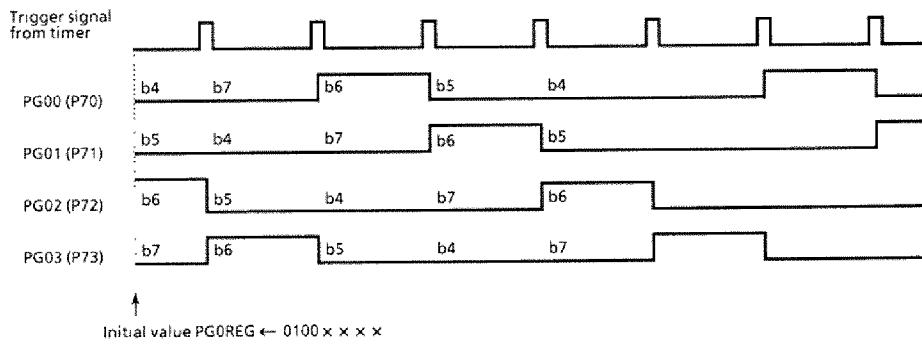
In this pattern generation mode, only writing the output latch is disabled by hardware, but other functions do the same operation as 1-2 excitation in stepping motor control port

mode. Accordingly, the data shifted by trigger signal from a timer must be written before the next trigger signal is output.

## (2) Stepping Motor Control Mode

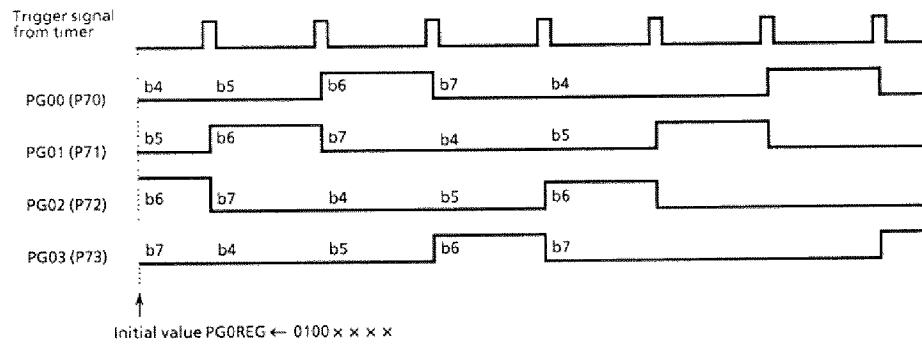
## ① 4-phase 1-Step/2-Step Excitation

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



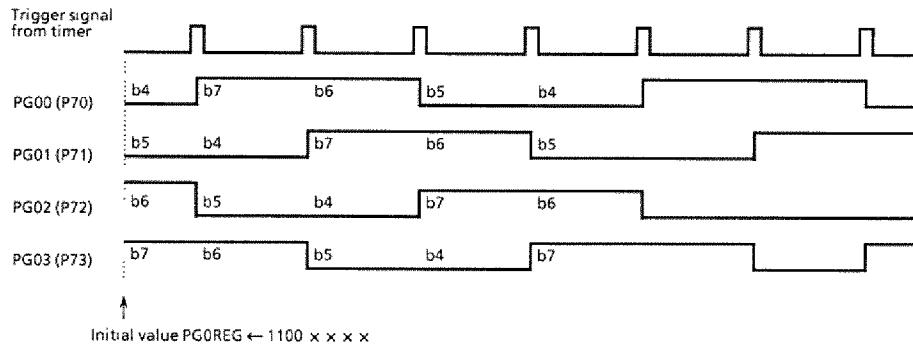
Note : bn indicates the initial value of PG0REG ← b7 b6 b5 b4 × × ×

## ① Normal Rotation



## ② Reverse Rotation

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



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

The operation when channel 0 is selected is explained below.

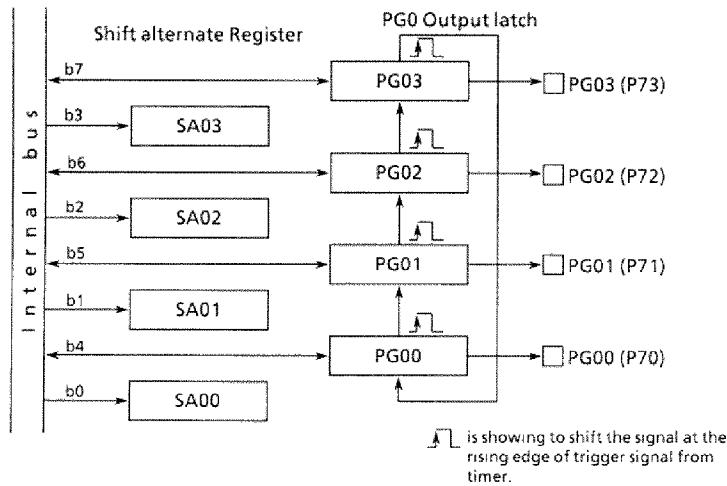
The output latch of PG0 (also used as P6) is shifted at the rising edge of the trigger signal from the timer to be output to the port.

The direction of shift is specified by PG01CR <CCW0>: Normal rotation (PG00 → PG01 → PG02 → PG03) when <CCW0> is set to "0"; reverse rotation (PG00 ← PG01 ← PG02 ← PG03) when "1". 4-phase

1-step excitation will be selected when only one bit is set to "1" during the initialization of PG, while 4-phase 2-step excitation will be selected when two consecutive bits are set to "1".

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

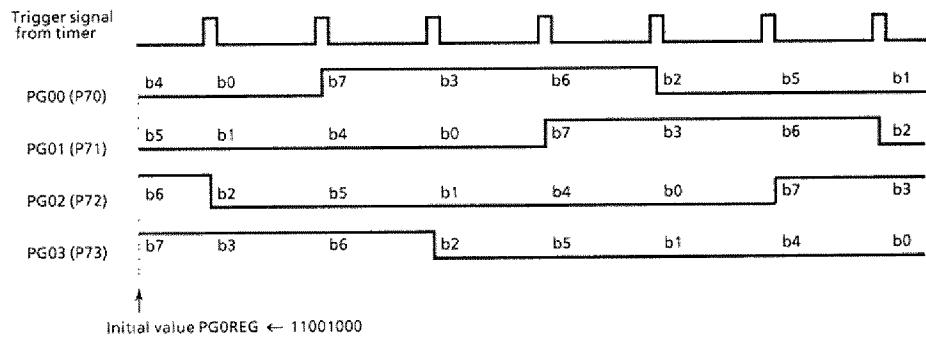
Figure 3.10 (10) shows the block diagram.



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

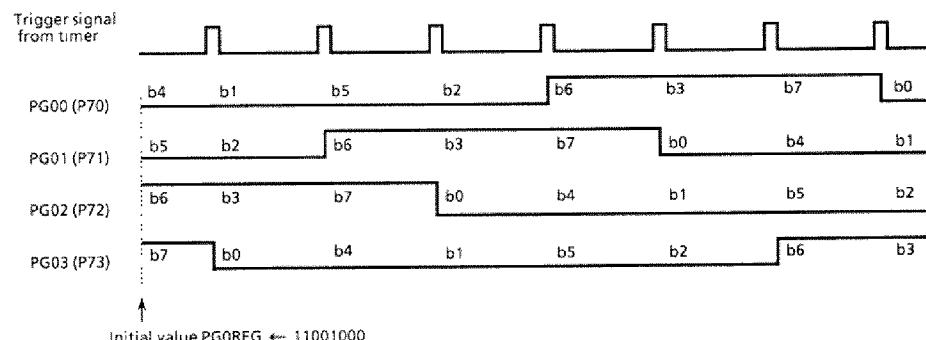
## ② 4-Phase 1-2 Step Excitation

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



Note: bn denotes the initial value  $\text{PG0REG} \leftarrow b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$

## ① Normal Rotation



## ② Reverse Rotation

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

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

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

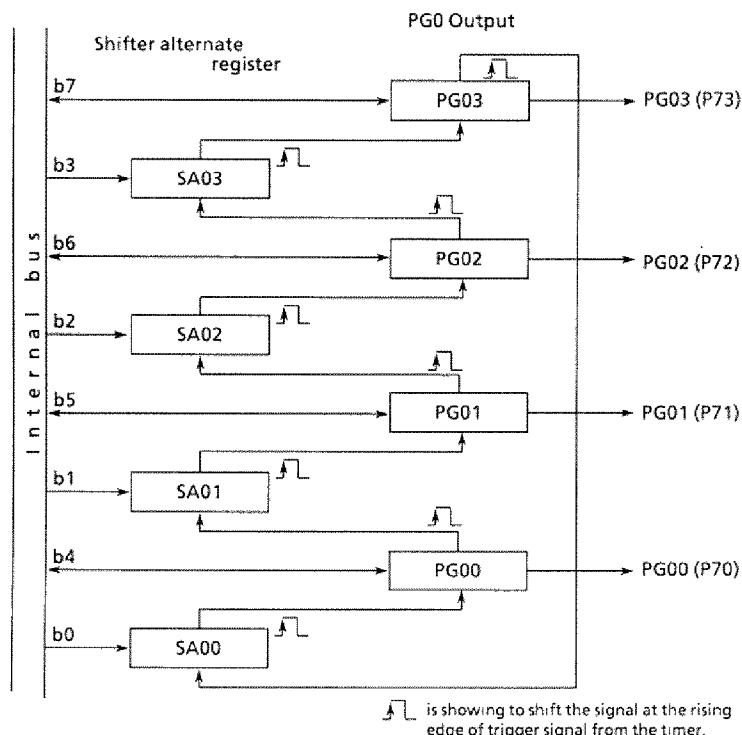
For example, if b7, b3, and b6 are set to "1", the initial value becomes "11001000", obtaining the output waveforms as shown in Figure 3.10 (11).

To get an output waveform of negative logic, set values 1's and 0's of the initial value should be inverted. For

example, to change the output waveform shown in Figure 3.10 (11) into negative logic, change the initial value to "00110111".

The operation will be explained below for channel 0. The output latch of PG0 (shared by P7) and the shifter alternate register (SA0) for Pattern Generation are shifted at the rising edge of trigger signal from the timer to be output to the port. The direction of shift is set by PG01CR <CCW0>.

Figure 3.10 (12) shows the block diagram.



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

Setting example: To drive channel 0 (PG0) by 4-phase 1-2 step excitation (normal rotation) when

timer 0 is selected, set each register as follows:

	7 6 5 4 3 2 1 0	
TRUN	← - X - - - - 0	Stop timer 0, and clear it to zero.
TMOD	← 0 0 X X - - 0 1	Set 8-bit timer mode and select $\phi T_1$ as the input clock of timer 0.
TFFCR	← X X X 0 1 0 1 0	Clear TFF1 to zero and enable the inversion trigger by timer 0.
TREG0	← * * * * * * * *	Set the cycle in timer register.
P7CR	← - - - - 1 1 1 1	Set P70 to P73 bits to the output mode.
P7FC	← - - - - 1 1 1 1	Set P70 to P73 bits to the PG output.
PG01CR	← - - - - 0 0 1 1	Select PG0 4-phase 1-2 step excitation mode and normal rotation .
PG0REG	← 1 1 0 0 1 0 0 0	Set an initial value.
TRUN	← 1 X - - - - 1	Start timer 0.

Note: X; don't care    - ; no change

### (3) Trigger Signal From Timer

The trigger signal from the timer which is used by PG is

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

**Table 3.10 (1) Select of Trigger Signal**

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

Note: To shift PG, TFFCR <FF1IE> must be set to "1" to enable TFF1 inversion.

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

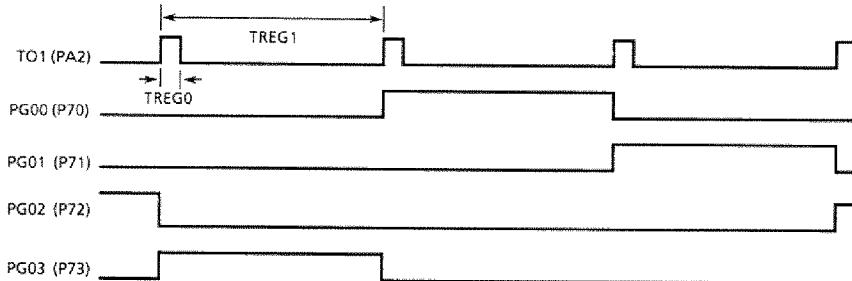
When using a trigger signal from Timer 4, set either T4FFCR <EQ5T4> or T4MOD <EQ5T5> to "1" and a

trigger is generated when the value in UC4 and the value in TREG5 match. When using a trigger signal from Timer 5, set T5FFCR <EQ7T6> to 1. Generates a trigger when the value in UC5 and the value in TREG7 match.

## (4) Application of PG and Timer Output

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

To drive a stepping motor, in addition to the value of each phase (PG output), synchronizing signal is often required at the timing when excitation is changed over. In this application, port 7 is used as a stepping motor control port to output a synchronizing signal to the TO1 pin (shared by PA2).



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

Setting example:

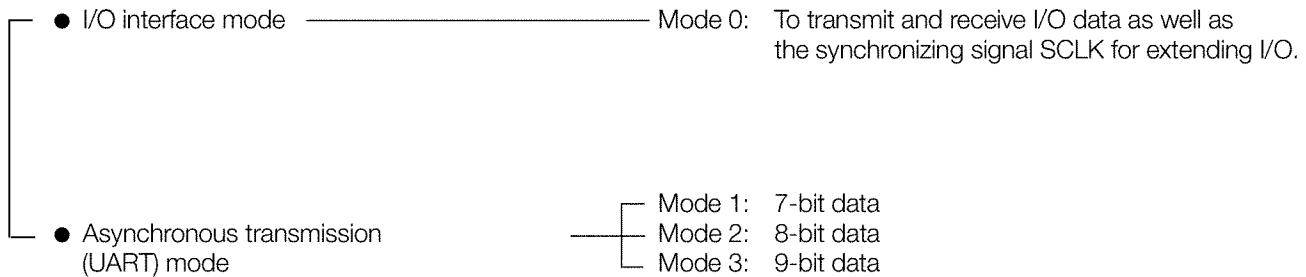
7 6 5 4 3 2 1 0	
TRUN ← - X - - - - 0 0	Stop timer 0, and clear it to zero.
TMOD ← 1 0 X X X X 0 1	Set timer 0 and timer 1 in PPG output mode and select φT1 as the input clock.
TFFCR ← X X X 0 0 1 1 X	Enable TFF1 inversion and set TFF1 to "1".
TREG0 ← * * * * * * * *	Set the duty of TO1 to TREG0.
TREG1 ← * * * * * * * *	Set the cycle of TO1 to TREG1.
PACR ← X X X X - - 1 -	
PAFC ← X X X X - - 1 X	} Assign PA2 as TO1.
P7CR ← - - - - 1 1 1 1	
P7FC ← - - - - 1 1 1 1	} Assign P70 to 73 as PG0.
PG01CR ← - - - - 0 0 0 1	Set PG0 in 4-phase 1-step excitation mode.
PG0REG ← * * * * * * * *	Set an initial value.
TRUN ← 1 X - - - - 1 1	Start timer 0 and timer 1.

Note: X ; don't care   - ; no change

### 3.11 Serial Channel

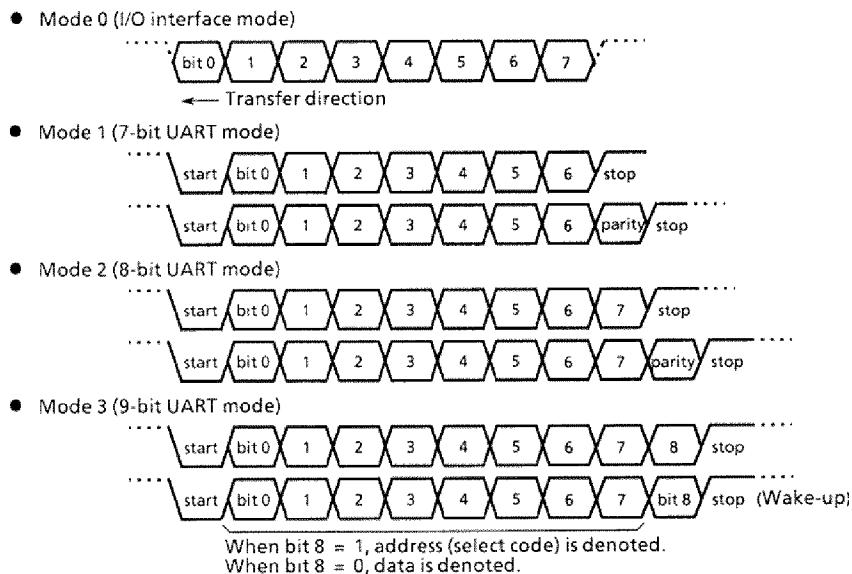
TMP96C061 contains two serial Input/Output channels.

The serial channel has the following operation modes:



In mode 1 and mode 2, a parity bit can be added. Mode 3 has wake-up function for making the master controller start slave controllers in serial link (multi-controller system).

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



**Figure 3.11 (1). Data Formats**

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

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

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

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

In the UART mode, a check function is added not to start the receiving operation by error start bits due to noise. The channel starts receiving data only when the start bit is

detected to be normal at least twice in three samplings.

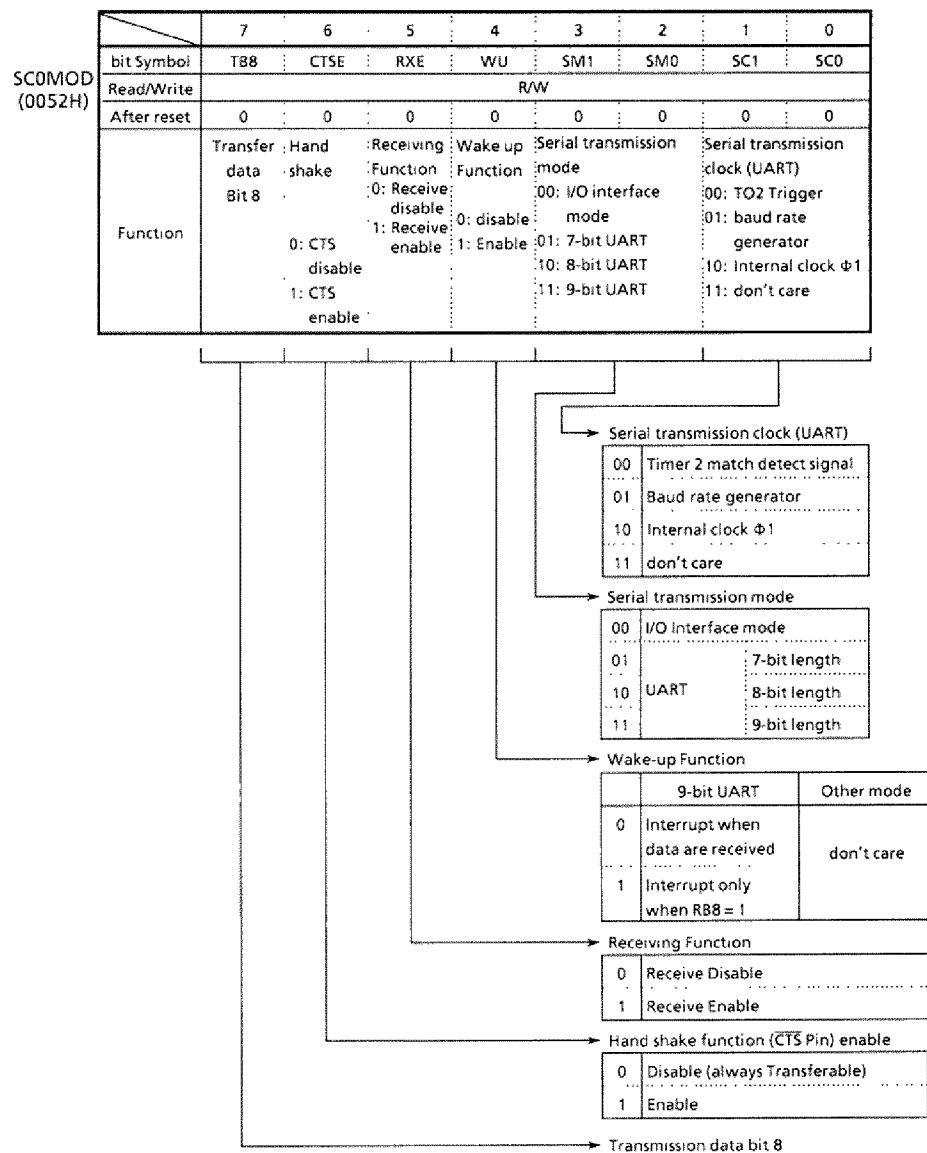
When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during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

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of four clocks ( $\phi T_0$ ,  $\phi T_2$ ,  $\phi T_8$ , and  $\phi T_{32}$ ) from the internal prescaler (shared by 8-bit/16-bit timer) by the value 2 to 16.

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

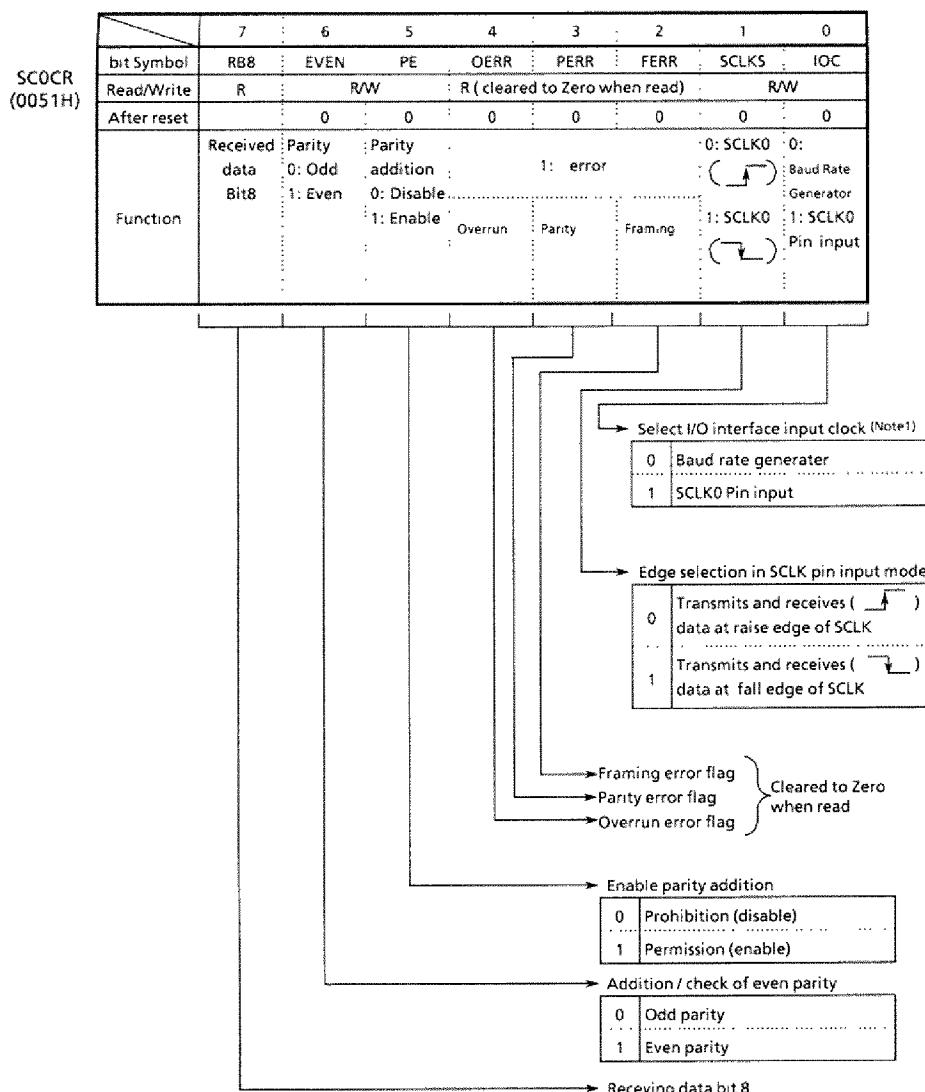
### 3.11.1 Control Registers

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



Note : There is SC1MOD (56H) in Channel1

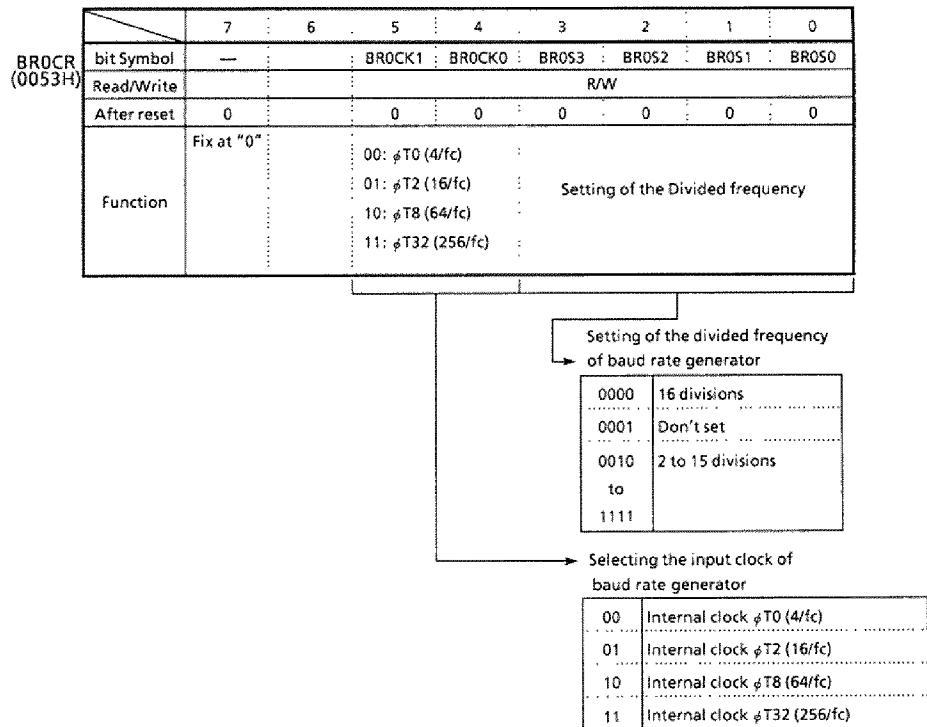
Figure 3.11 (2). Serial Mode Control Register (Channel 0, SC0MOD)



Note : Serial control register for channel 1 is SC1CR (55H).

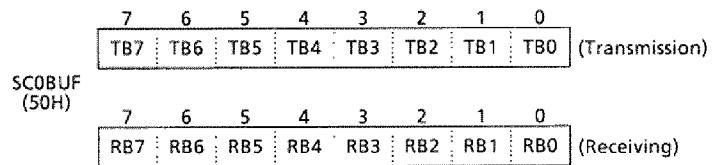
Note : As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

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



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

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



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

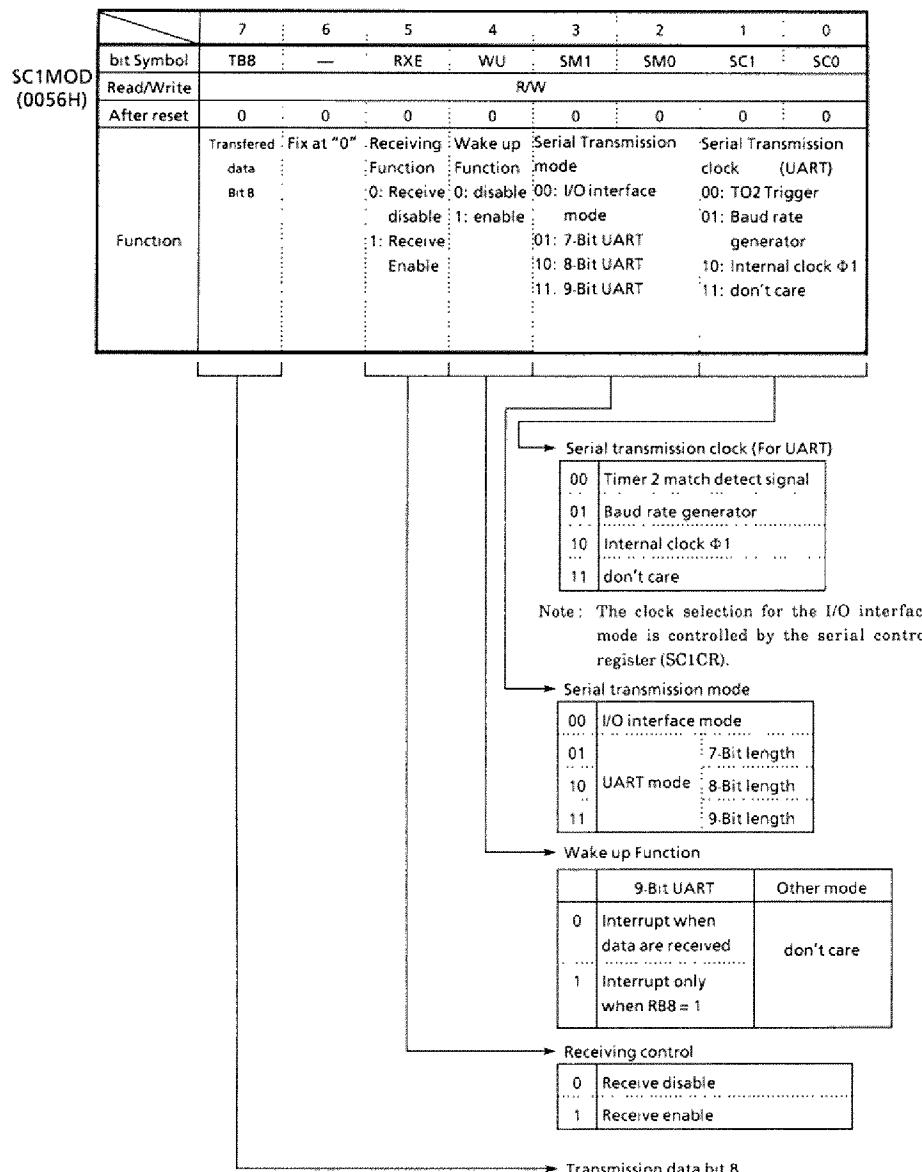
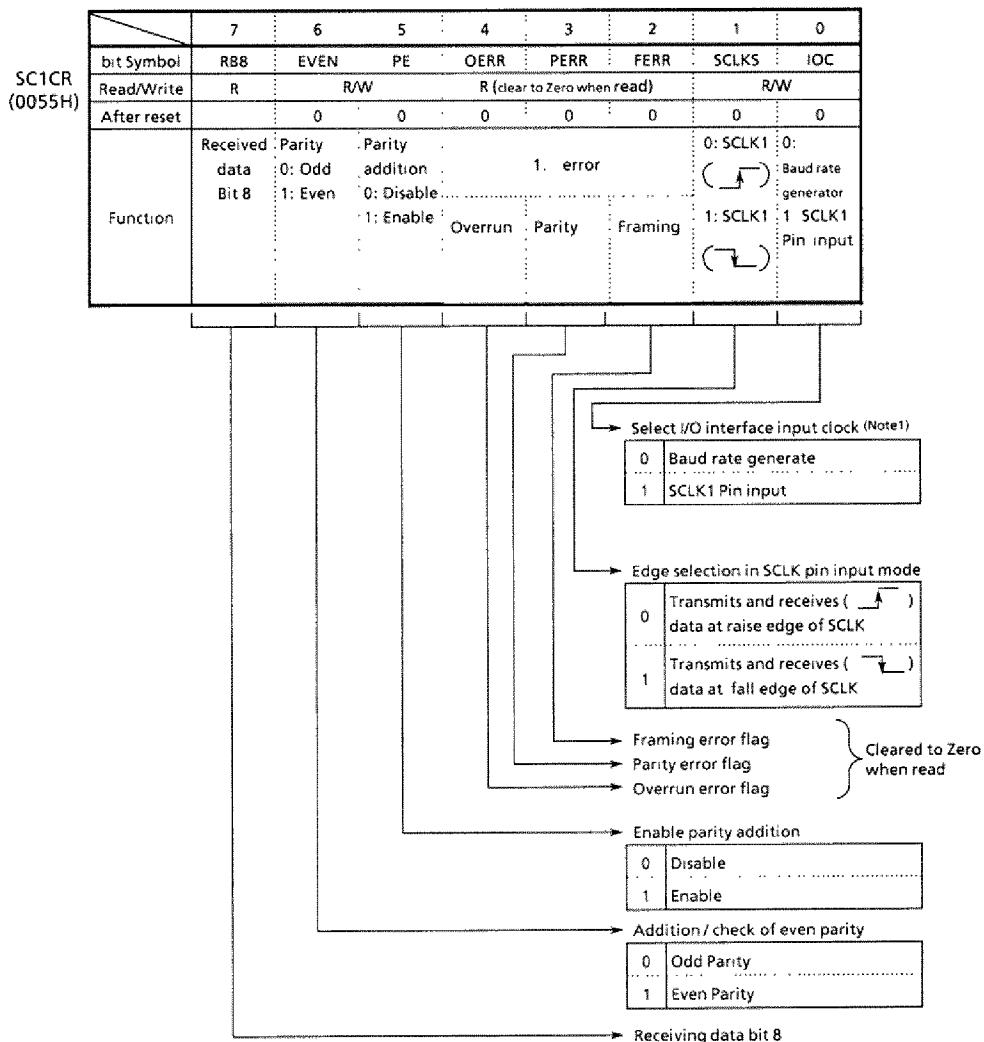
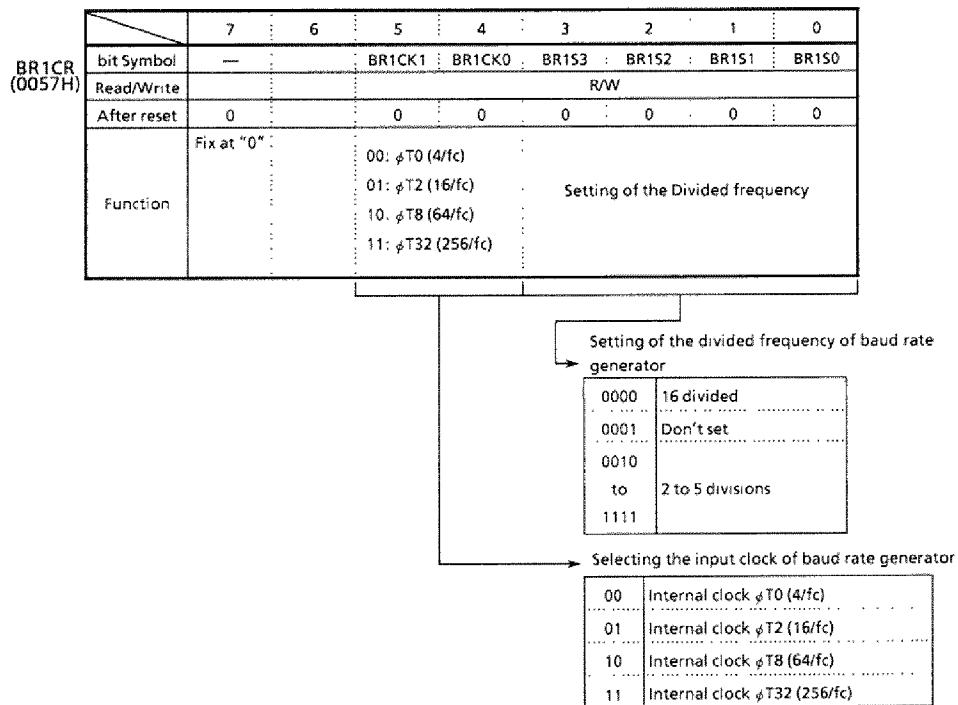


Figure 3.11 (6). Serial Mode Control Register (Channel 1, SC1MOD)



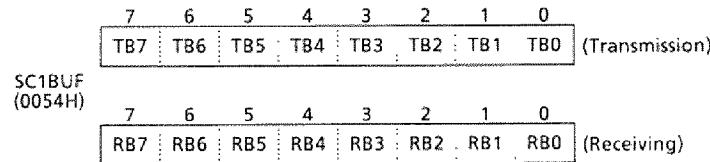
Note : As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

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



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

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



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

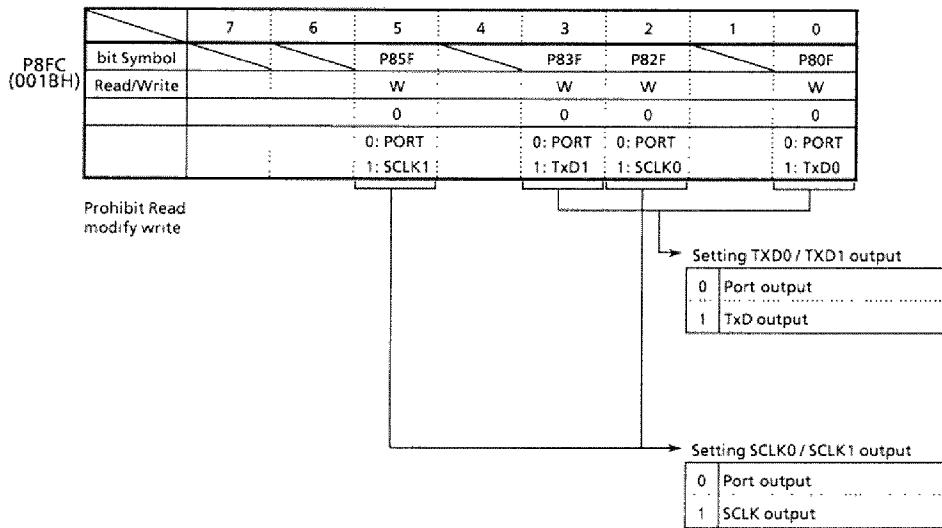
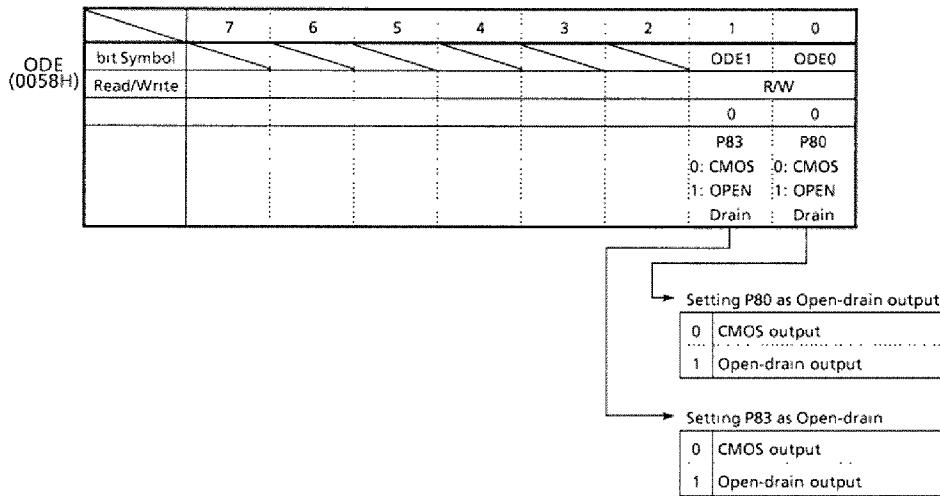


Figure 3.11 (10). Port 9 Function Register (P9FC)



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

### 3.11.2 Configuration

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

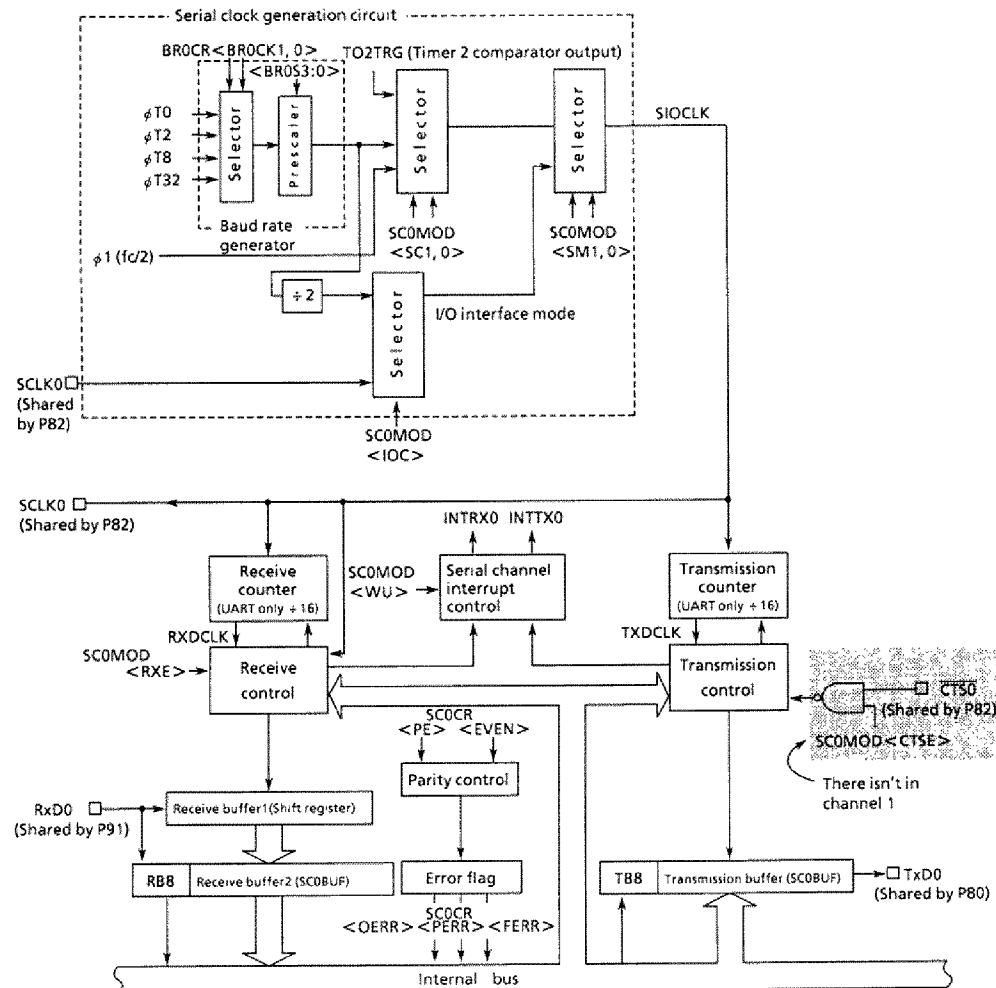
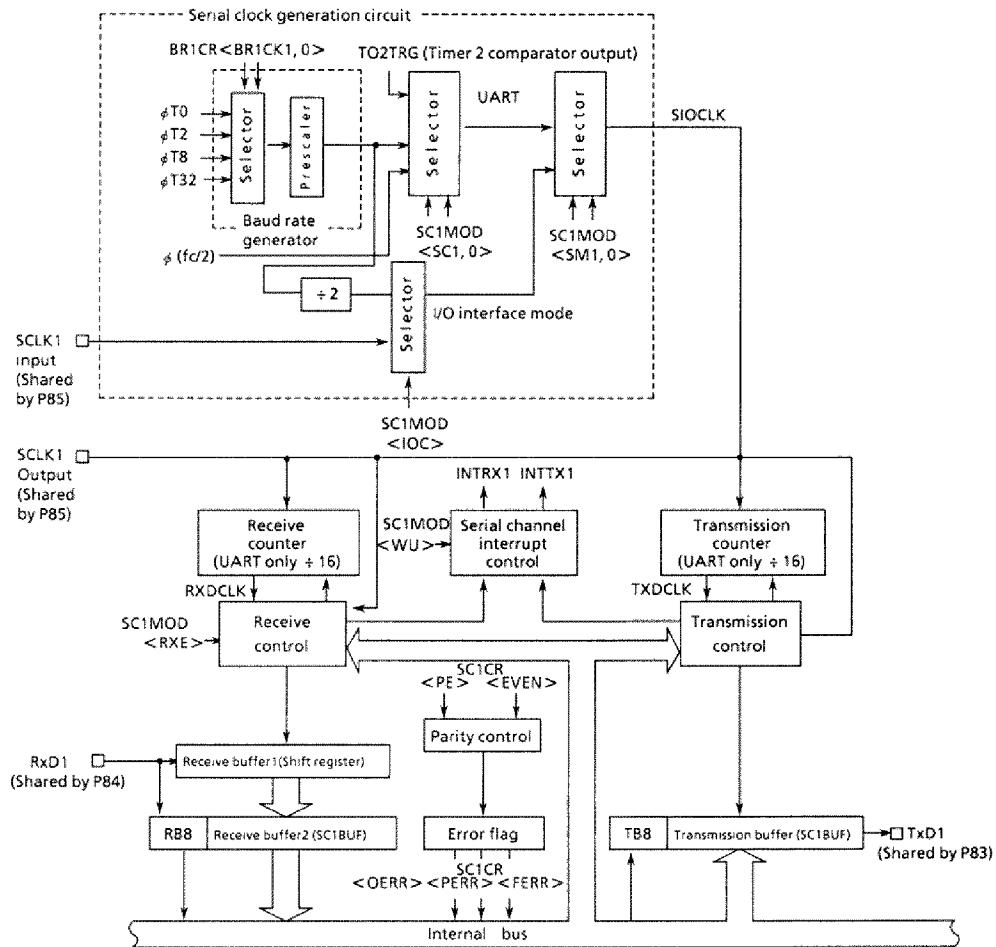


Figure 3.11 (12). Block Diagram of the Serial Channel 0

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



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

① Baud Rate Generator

Baud rate generator comprises a circuit that generates transmission and receiving clocks to determine the transfer rate of the serial channel.

The input clock to the baud rate generator,  $\phi T0$  ( $fc/4$ ),  $\phi T2$  ( $fc/16$ ),  $\phi T8$  ( $fc/64$ ), or  $\phi T32$  ( $fc/256$ ) 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 bit <BR0CK1/0>/<BR1CK1.0> of BR0CR/BR1CR.

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

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

The relation between the input clock and the source clock ( $fc$ ) is as follows:

$$\phi T0 = fc/4$$

$$\phi T2 = fc/16$$

$$\phi T8 = fc/64$$

$$\phi T32 = fc/256$$

Accordingly, when source clock  $fc$  is 12.288 MHz, input clock is  $\phi T2$  ( $fc/16$ ), and frequency divisor is 5, the transfer rate in UART mode becomes as follows:

$$\begin{aligned} \text{Transfer rate} &= \frac{fc/16}{5} \div 16 \\ &= 12.288 \times 10^6 / 16 / 5 / 16 = 9600 (\text{bps}) \end{aligned}$$

Table 3.11 (1) shows an example of the transfer rate in UART mode.

Also with 8-bit timer 2, the serial channel can get a transfer rate. Table 3.11 (2) shows an example of baud rate using timer 2.

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

$fc$ [MHz]	Frequency Divisor	Input Clock				Unit (kbps)
		$\phi T0$ ( $fc/4$ )	$\phi T2$ ( $fc/16$ )	$\phi T8$ ( $fc/64$ )	$\phi T32$ ( $fc/256$ )	
9.830400	2	76.800	19.200	4.800	1.200	
↑	4	38.400	9.600	2.400	0.600	
↑	8	19.200	4.800	1.200	0.300	
↑	16	9.600	2.400	0.600	0.150	
12.288000	5	38.400	9.600	2.400	0.600	
↑	A	19.200	4.800	1.200	0.300	
14.745600	3	76.800	19.200	4.800	1.200	
↑	6	38.400	9.600	2.400	0.600	
↑	C	19.200	4.800	1.200	0.300	

Note: Transfer rate in I/O interface mode is 8 times as fast as the values given in the above table.

**Table 3.11 (2) Selection of Transfer Rate (1) (When Timer 2 (Input Clock  $\phi T1$ ) is Used)**

TREGO	fc	Unit (Kbps)			
		12.288MHz	12MHz	9.8304MHz	8MHz
1H	96			76.8	62.5
2H	48			38.4	31.25
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 2 is used):

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

↑ (When Timer 2 (input clock  $\phi T1$ ) is used)

Input clock of Timer 2

$$\phi T1 = 8/fc$$

$$\phi T4 = 32/fc$$

$$\phi T16 = 128/fc$$

Note 1: Timer 2 match detect signal cannot be used as the transfer clock in I/O interface mode.

## ② Serial Clock Generation Circuit

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

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

When in SCLK output mode with the setting of SC1CR <IOC> = "0", the basic clock will be generated by dividing by 2 the output of the baud rate generator as described before. When in SCLK input mode with the setting of SC0CR <IOC> = "1", the rising edge or falling edge will be detected according to the setting of SC0CR <SCLKS> register to generate the basic clock.

### 2) Asynchronous Communication (UART) mode

According to the setting of SC0MOD/SC1MOD <SC1, 0>, the above baud rate generator clock, internal clock  $\phi 1$  (max. 500 Kbps @ fc = 16MHz), or the match detect signal from timer 0 will be selected to generate the basic clock SIOCLK.

## ③ Receiving Counter

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

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

For example, if the sampled data bit is "1", "0" and "1" at 7th, 8th and 9th clock respectively, the received data is evaluated as "1". The sampled data "0", "0" and "1" is evaluated that the received data is "0".

## ④ Receiving Control

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

When in SCLK1 output mode with the setting of SC1CR <IOC> = "0", RxD1 signal will be sampled at the rising edge of shift clock which is output to SCLK pin.

When in SCLK input mode with the setting SC1CR <IOC> = "1", RxD1 signal will be sampled at the rising edge or falling edge of SCLK input according to the setting of SC1CR <SCLKS> register.

## 2) Asynchronous Communication (UART) mode

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

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

### ⑤ Receiving Buffer

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

Received data is stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data are stored in the receiving buffer 1, the stored data is transferred to another receiving buffer 2 (SC0BUF/SC1BUF), generating an interrupt INTRX0/INTRX1. The CPU reads only receiving buffer 2 (SC0BUF/SC1BUF). Even before the CPU reads the receiving buffer 2 (SC0BUF/SC1BUF), the received data can be stored in the receiving buffer 1. However, unless the receiving buffer 2 (SC0BUF/SC1BUF) is

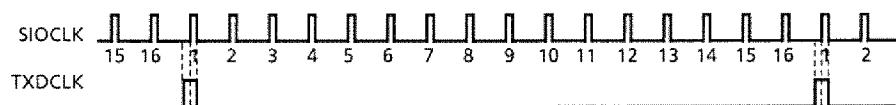
read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR <RB8> SC1CR <RB8> are still preserved. Reading data from receive data buffer 2 (SC0BUF/SC1BUF) clears interrupt request flags INTR0 <IRX0C> and INTRX1 <IRX1C7>.

The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SC0CR <RB8>/SC1CR <RB8>.

When in 9-bit UART mode, the wake-up function of the slave controllers is enabled by setting SC0MOD <WU>/SC1MOD <WU> to "1", and interrupt INTRX0/INTRX1 occurs only when SC0CR <RB8>/SC1CR <RB8> is set to "1".

### ⑥ Transmission Counter

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



**Figure 3.11 (14). Generation of Transmission Clock**

## ⑦ Transmission Controller

### 1) I/O interface mode

In SCLK0 output mode with the setting of SC1CR <IOC> = "0", the data in the transmission buffer are output bit by bit to TxD1 pin at the rising edge of shift clock which is output from SCLK1 pin.

In SCLK1 input mode with the setting SC1CR <IOC> = "1", the data in the transmission buffer are output bit

by bit to TxD1 pin at the rising edge or falling edge of SCLK input according to the setting of SC1CR <SCLKS> register.

### 2) Asynchronous Communication (UART) mode

When transmission data is written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxDCLK, generating a transmission shift clock TxDSFT.

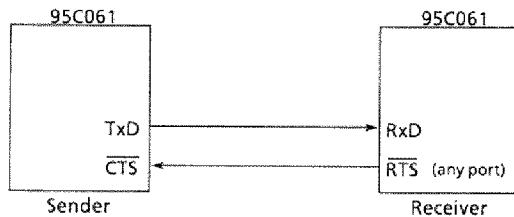
### Handshake function

Serial channel 0 has a CTS0 pin. Using this pin, data can be sent in units of one frame; thus, overrun errors can be avoided. The handshake function is enabled/disabled by SC0MOD <CTSE>.

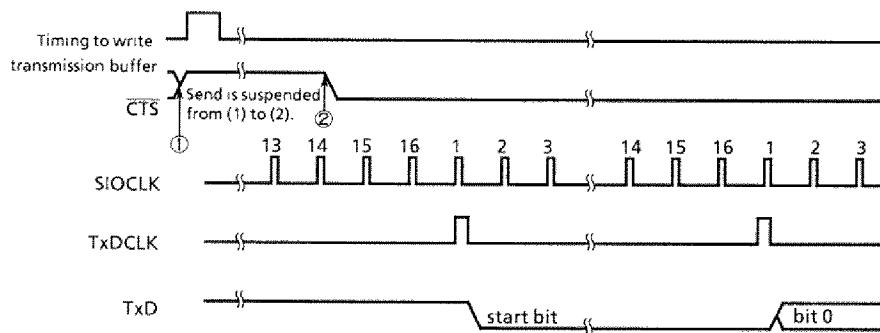
When the CTS0 pin goes high, after completion of the current data send, data send is halted until the CTS0 pin goes low again. The INTTX0 Interrupts are gener-

ated, requests the next send data to the CPU.

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



**Figure 3.11 (15). Handshake Function**



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

Note 2 : Transmission starts at the first TxDCLK clock fall after the CTS signal falls.

**Figure 3.11 (16). Timing of CTS (Clear to Send)**

### ⑧ Transmission Buffer

Transmission buffer (SC0BUF/SC1BUF) shifts to and sends the transmission data written from the CPU from the least significant bit (LSB) in order, using transmission shift clock TxDSFT which is generated by the transmission control. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0/INTTX1 interrupt.

### ⑨ Parity Control Circuit

When serial channel control register SC0CR <PE>/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 mode. With SC0CR <EVEN>/SC1CR <EVEN> register, even (odd) parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer (SC0BUF/SC2BUF), and data are transmitted after being stored in SC0BUF <TB7>/SC1BUF <TB7> when in 7-bit UART mode while in SC0MOD <TB8>/SC1MOD <TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before transmission data are written in the transmission buffer.

For receiving, data is shifted in the receiving buffer 1, and parity is added after the data is transferred in the receiving buffer 2 (SC0BUF/SC1BUF), and then compared with SC0BUF <RB7>/SC1BUF when in 7-bit

UART mode and with SC0MOD <RB8>/SC1MOD when in 8-bit UART mode. If they are not equal, a parity error occurs, and SC0CR <PERR>/SC1CR <PERR> flag is set.

### ⑩ 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 (SCBUFO/1), an overrun error will occur.

#### 2. Parity error <PERR>

The parity generated for the data shifted in receiving buffer 2 (SCBUFO/1) 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.

### ⑪ Generating Timing

#### 1) UART mode

##### Receiving

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit
Framing error timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit
Overrun error timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit

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

##### Transmitting

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

## 2) I/O Interface mode

Transmission interrupt timing	SCLK output mode	Immediately after rise of last SCLK signal. (See Figure 3.11 (19).)
	SCLK input mode	Immediately after rise of last SCLK signal (rising mode), or immediately after fall in falling mode. (See Figure 3.11 (20).)
Receiving interrupt timing	SCLK output mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after last SCLK. (See Figure 3.11 (21).)
	SCLK input mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after SCLK. (See Figure 3.11 (22).)

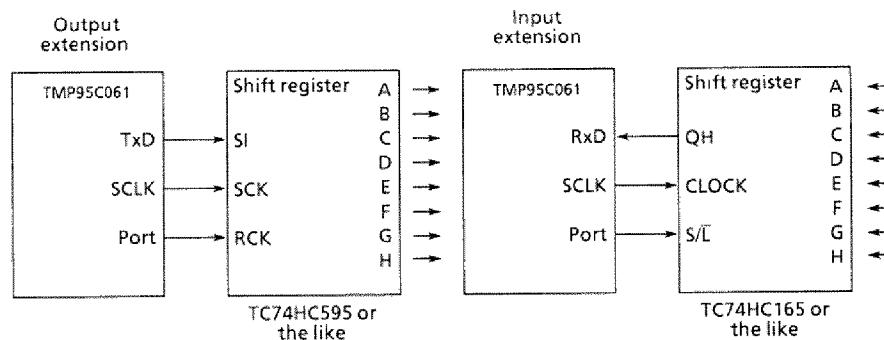
### 3.11.3 Operational Description

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

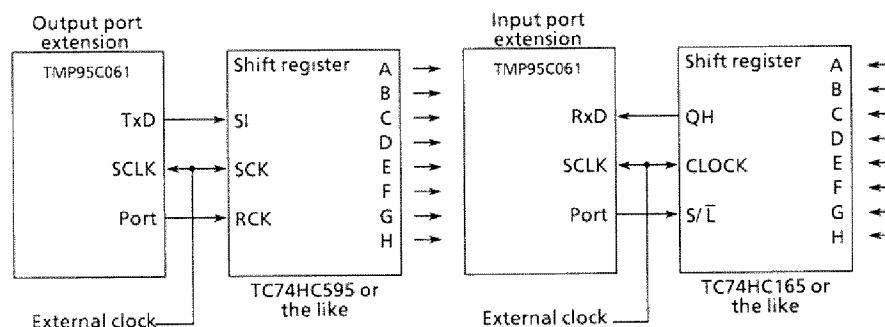
This mode is used to increase the number of I/O pins

for transmitting or receiving data to or from the external shifter register.

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



**Figure 3.11 (17). Example of SCLK Output Mode Connection**

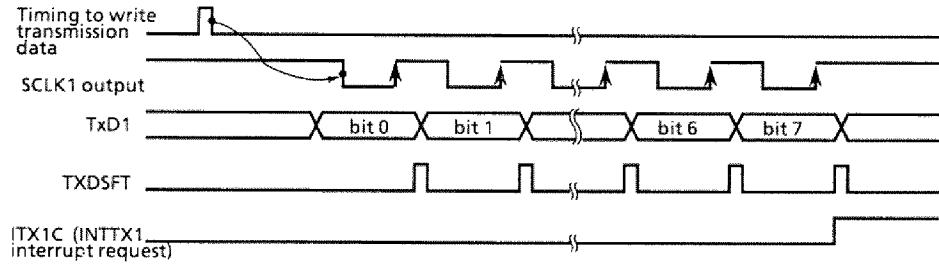


**Figure 3.11 (18). Example of SCLK Input Mode Connection**

## ① Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TxD pin and SCLK0 pin, respectively, each

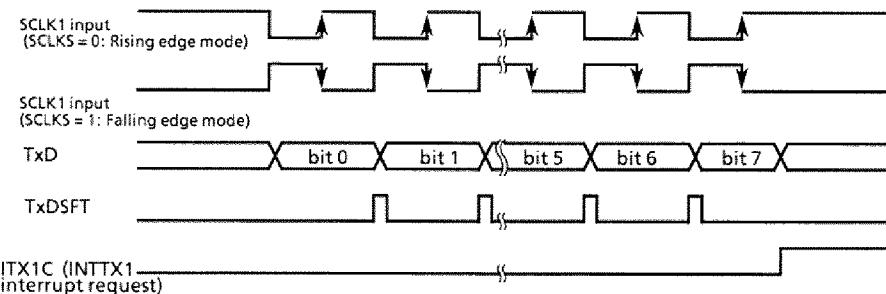
time the CPU writes data in the transmission buffer. When all data is output, INTES0 <ITX0C> will be set to generate INTTX0 interrupt.



**Figure 3.11 (19) Transmitting Operation in I/O Interface Mode (SCLK Output Mode) (Channel 1)**

In SCLK output mode, 8-bit data are output from TxD0 pin when SCLK0 input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES0 <ITX0C> will be set to generate INTTX0 interrupt.

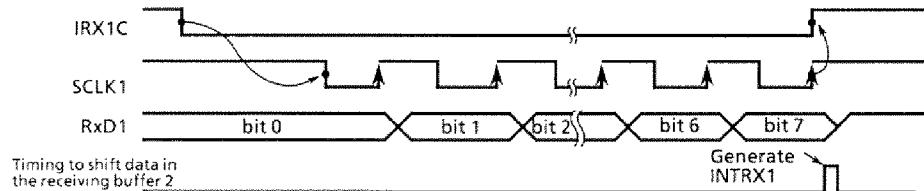


**Figure 3.11 (20). Transmitting Operation in I/O Interface Mode (SCLK Input Mode)**

## ② Receiving

In SCLK output mode, synchronous clock is output from SCLK pin and the data is shifted in the receiving buffer 1 whenever the receive interrupt flag INTES0

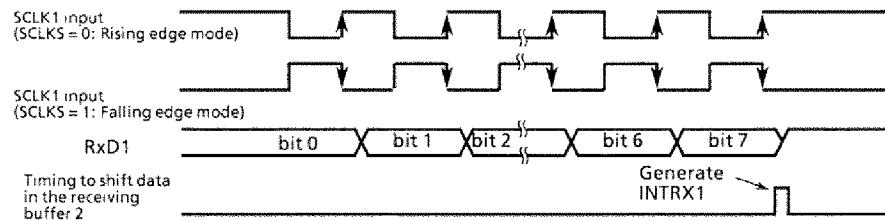
<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred in the receiving buffer 2 (SC1BUF) at the timing shown below, and INTES1 <IRX1C> will be set again to generate INTRX1 interrupt.



**Figure 3.11 (21). Receiving Operation in I/O Interface Mode (SCLK Output Mode)**

In SCLK input mode, the data is shifted in the receiving buffer 1 when SCLK input becomes active, while the receive interrupt flag INTES1 <IRX1C> is cleared by reading the received data. When 8-bit data is received, the

data will be shifted in the receiving buffer 2 (SC1BUF) at the timing shown below, and INTES1 <IRX1C> will be set again to generate INTRX interrupt.



**Figure 3.11 (22). 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")

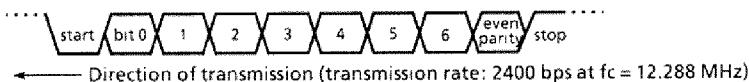
## (2) Mode 1 (7-bit UART Mode)

The 7-bit mode can be set by setting serial channel mode register SC0MOD <SM1,0> /SC1MOD <SM1,0> to "01".

In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register SC0CR <PE> /SC1CR <PE>, and even parity or odd parity is selected by SC0CR <EVEN> /SC1CR <EVEN> when <PE> is set to "1" (enable).

and even parity or odd parity is selected by SC0CR <EVEN> /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
P8CR	← X X - - - - 1
P8FC	← X X - X - - X 1
SC0MOD	← X 0 - X 0 1 0 1
SC0CR	← X 1 1 X X X 0 0
BROCR	← 0 X 1 0 0 1 0 1
TRUN	← 1 X - - - - -
INTES0	← 1 1 0 0 - - - -
SC0BUF	← * * * * * * *

} Select P80 as the TxD pin.  
Set 7-bit UART mode.  
Add an even parity.  
Set transfer rate at 2400 bps.  
Start the prescaler for the baud rate generator.  
Enable INTTX0 interrupt and set interrupt level 4.  
Set data for transmission.

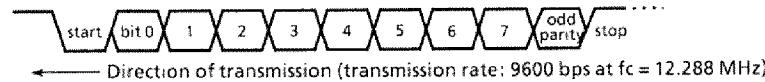
Note: X ; don't care      - ; no change

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

The 8-bit UART mode can be specified by setting SC0MOD <SM1,0> / SC1MOD <SM1,0> to "10". In this mode, parity bit can be added, the addition of a parity bit is enabled or disabled by SC0CR <PE>, and even parity or odd parity is selected by SC0CR <EVEN> /SC1CR <EVEN> when <PE> is set to "1" (enable).

even parity or odd parity is selected by SC0CR <EVEN> /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
P8CR	← X X - - - - 0 -
SC0MOD	← - 0 1 X 1 0 0 1
SC0CR	← X 0 1 X X X 0 0
BROCR	← 0 X 0 1 0 1 0 1
TRUN	← 1 X - - - - -
INTES0	← - - - - 1 1 0 0

Select P81 (RxD) as the input pin.  
Enable receiving in 8-bit UART mode.  
Add an odd parity.  
Set transfer rate at 9600 bps.  
Start the prescaler for the baud rate generator.  
Enable INTRX0 interrupt and set interrupt level 4.

## Interrupt processing

Acc ← SC0CR AND 00011100	} Check for error.
if Acc ≠ 0 then ERROR	

Acc ← SC0BUF      Read the received data.

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

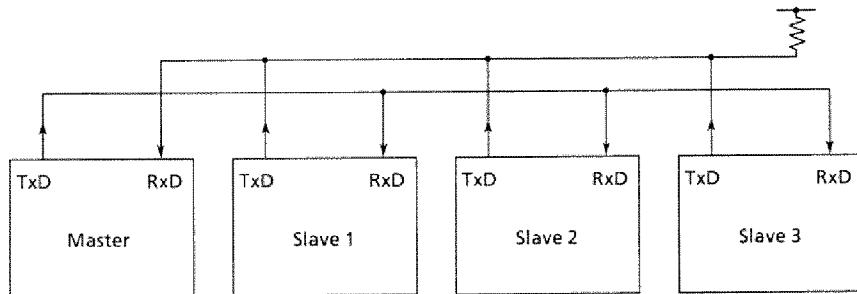
## (4) Mode 3 (9-bit UART Mode)

The 9-bit UART mode can be specified by setting SC0MOD <SM1,0> /SC1MOD <SM1,0> to “11”. In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SC0MOD <TB8>, while in receiving it is stored in SCCR <RB8>. For writing and reading the buffer, the MSB is read or written first, then SC0BUF/SC1BUF.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SC0MOD <WU> / SC1MOD <WU> to “1”. The interrupt INTRX1/INTRX0 occurs only when <RB8> = 1.



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

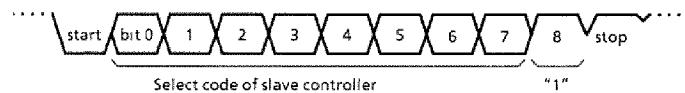
**Figure 3.11 (23). Serial Link Using Wake-Up Function**

## Protocol

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

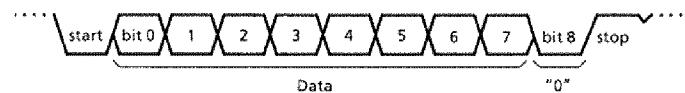
③ The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (bit 8) <TB8> is set to "1".

② Set SC0MOD <WU>/SC1MOD <WU> bit of each slave controller to "1" to enable data receiving.



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

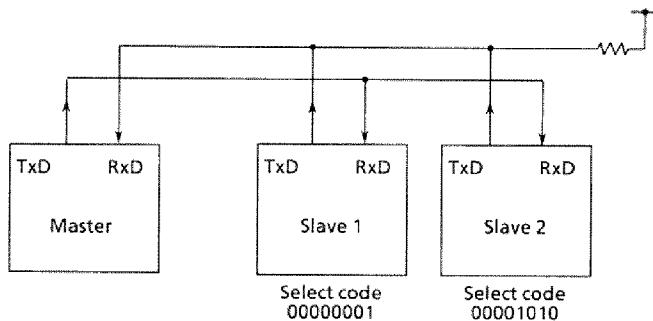
⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU>/SC1MOD <WU> bit is cleared to "0." The MSB (bit 8) <TB8> is cleared to "0".



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 INTRX0/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  $\phi_1$  ( $f_C/2$ ) as the transfer clock.



Since serial channels 0 and 1 operate in exactly the

same way, channel 0 is used for the purposes of explanation.

#### ● Setting the master controller

Main

```
P8CR ← X X - - - 0 1      } Select P80 as TxDO pin and P81 as RxDO pin.
P8FC  ← X X - X - X X 1
INTES0 ← 1 1 0 0 1 1 0 1    Enable INTTX0 and set the interrupt level 4.
                             Enable INTTX0 and set the interrupt level 5.
SC0MOD ← 1 0 1 0 1 1 1 0    Set  $\phi_1$  as the transmission clock in 9-bit UART mode.
SC0BUF ← 0 0 0 0 0 0 0 1    Set the select code for slave controller 1.
```

INTTX0 interrupt

```
SC0MOD ← 0 - - - - -      Sets TB8 to "0".
SC0BUF ← * * * * * * *     Set data for transmission.
```

#### ● Setting the slave controller 2

Main

```
P8CR ← X X - - - 0 1      } Select P81 as RxDO pin and P90 as TxDO pin (open drain
P8FC  ← X X - X - X X 1
ODE   ← X X X X X X - 1
INTES0 ← 1 1 0 1 1 1 1 0    output).
                             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  $\phi_1$  ( $f_C/2$ ).
```

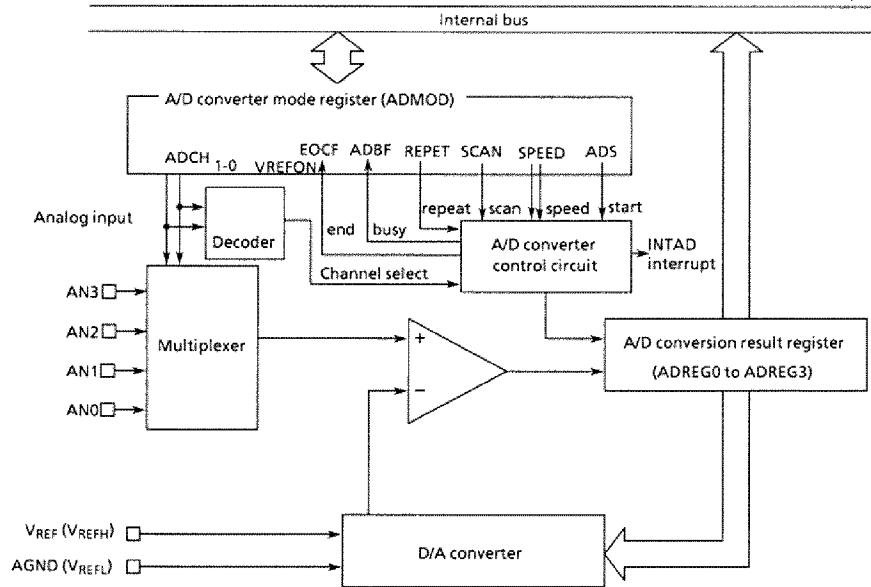
INTRX0 interrupt

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

### 3.12 Analog/Digital Converter

TMP95C061 contains a high-speed analog/digital converter (A/D converter) with 4-channel analog input that features 10-bit successive approximation.

Figure 3.12 (1) shows the block diagram of the A/D converter. The 4-channel analog input pins (AN3 to AN0) are shared by input-only P9 and so can be used as input port.



**Figure 3.12 (1). Block Diagram of A/D Converter**

Note 1: This A/D converter does not have a built-in sample and hold circuit. Therefore, when A/D converting high-frequency signals, connect a sample and hold circuit externally.

Note 2: To lower the power supply current in IDLE or STOP mode, depending on the timing, standby mode can be entered with the internal comparator in enable state. Thus, stop A/D conversion before executing the HALT instruction.

The ladder resistor between  $V_{REF} (V_{REFH})$  - AGND ( $V_{REFL}$ ) cannot be disconnected internally.

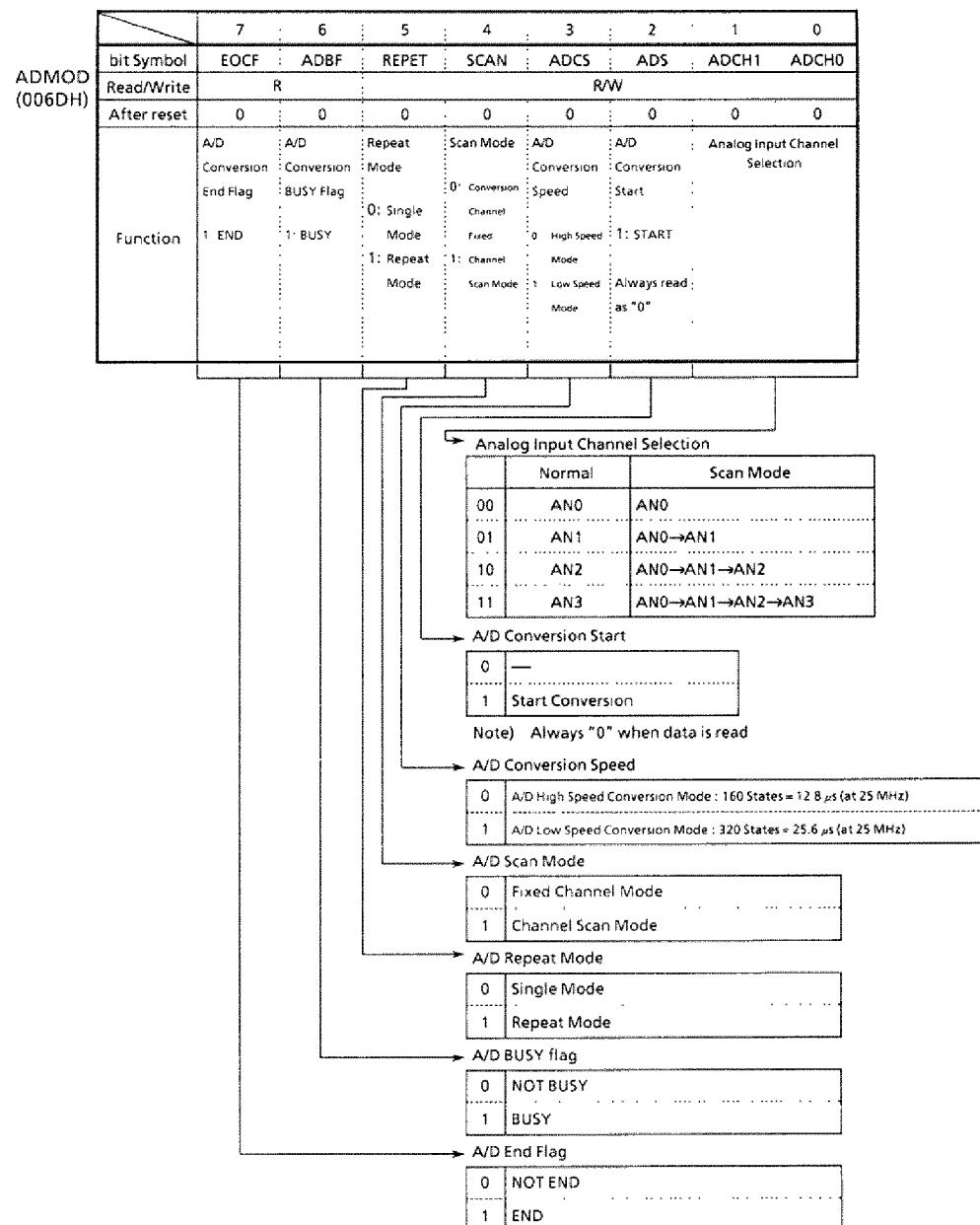


Figure 3.12 (2). A/D Control Register

ADREG04L (0060H)	7	6	5	4	3	2	1	0	
	bit Symbol	ADR01	ADR00						
	Read/Write				R				
	After reset	Undefined		1	1	1	1	1	
Function									
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									
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	
Function									
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									

Figure 3.12 (3-1). A/D Conversion Result Register (ADREG0, 1)

ADREG2L (0064H)	7	6	5	4	3	2	1	0	
	bit Symbol	ADR21	ADR20						
	Read/Write				R				
	After reset	Undefined	1	1	1	1	1	1	
	Function	Lower 2 bits of A/D result for AN2 are stored.							
ADREG2H (0065H)	7	6	5	4	3	2	1	0	
	bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write				R				
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN2 are stored.							
ADREG3L (0066H)	7	6	5	4	3	2	1	0	
	bit Symbol	ADR31	ADR30						
	Read/Write				R				
	After reset	Undefined	1	1	1	1	1	1	
	Function	Lower 2 bits of A/D result for AN3 are stored.							
ADREG3H (0067H)	7	6	5	4	3	2	1	0	
	bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write				R				
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN3 are stored.							

Figure 3.12 (3-2). A/D Conversion Result Register (ADREG2, 3)

### 3.12.1 Operation

#### (1) Analog Reference Voltage

High analog reference voltage is applied to the VREF ( $V_{REFH}$ ) pin, and low analog reference voltage is applied to AGND ( $V_{REFL}$ ) pin.

The reference voltage between VREG ( $V_{REFH}$ ) and AGND ( $V_{REFL}$ ) is divided by 1024 using ladder resistance, and compared with the analog input voltage for A/D conversion.

#### (2) Analog Input Channels

Analog input channel is selected by ADMOD <ADCH1, 0>. However, which channel to select depends on the operation mode of the A/D converter.

In fixed analog input mode, one channel is selected by ADMOD <ADCH1, 0> among four pins: AN0 to AN3.

In analog input channel scan mode, the number of channels to be scanned from AN0 is specified by ADMOD <ADCH1, 0>, such as AN0 → AN1, AN0 → AN1 → AN2, and AN0 → AN1 → AN2 → AN3.

When reset, A/D conversion channel register will be initialized to ADMOD <ADCH1, 0> = 00, so that AN0 pin will be selected.

The pins which are not used as analog input channel can be used as ordinary input port P9.

#### (3) Starting A/D Conversion

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

#### (4) A/D Conversion Mode

Both fixed A/D conversion channel mode and A/D conversion channel scan mode have two conversion modes, i.e., single and repeat conversion modes.

In fixed channel repeat mode, conversion of specified one channel is executed repeatedly.

In scan repeat mode, scanning from AN0, ⋯ → AN3 is executed repeatedly.

A/D conversion mode is selected by ADMOD <REPET, SCAN>.

#### (5) A/D Conversion Speed Selection

There are four A/D conversion speed modes. The selection is executed by ADMOD <ADMCDSPED> register.

When reset, ADMOD <ADCS> will be initialized to "0," so that high speed conversion mode will be selected.

#### (6) A/D Conversion End and Interrupt

- A/D conversion single mode

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

- A/D conversion repeat mode

For both fixed conversion channel mode and conversion channel scan mode, INTAD should be disabled when in repeat mode. Always set the INTE0AD at "000," that disables the interrupt request.

Write "0" to ADMOD <REPET> to end the repeat mode. Then, the repeat mode will be exited as soon as the conversion in progress is completed.

#### (7) Storing the A/D Conversion Result

The results of A/D conversion are stored in ADREG0 to ADREG3 registers for each channel. In repeat mode, the registers are updated whenever conversion ends. ADREG0 to ADREG3 are read-only registers.

#### (8) Reading the A/D Conversion Result

The results of A/D conversion are stored in ADREG0 to ADREG3 registers. When the contents of one of ADREG0 to ADREG3 registers are read, ADMOD <EOCF> will be cleared to "0".

Reading data from the upper 8 bits of the register (ADREG0H, ADREG1H, ADREG2H, ADREG3H) for one of the channels in use clears interrupt request flag INTE0AD <IADC>.

Sample : ① When the analog input voltage of the AN3 pin is A/D converted in high speed mode (160 states) and the results is stored in the memory address 0100H by A/D interrupt INTAD routine.

Main setting

INTE0AD ← 1 1 0 0 - - - -	Enable INTAD and set interrupt level 4.
ADMOD ← X X 0 0 0 1 1 1	Specify AN3 pin as an analog input channel and starts A/D conversion in 160 states speed mode.

INTAD routine

WA ← ADREG3	Read ADREG3L and ADREG3H values and write to WA (16 bit)
WA >> 6	Right-shifts WA six times and writes 0 in upper bits.
(00FF10H)← WA	Writes contents of WA in memory at 0FF10H

② When the analog input voltage of AN0 to AN2 pins is A/D converted in high speed / channel scan mode.

INTE0AD ← 1 0 0 0 - - - -	Disable INTAD
ADMOD ← X X 1 1 0 1 1 0	specify AN0 to AN2 as an analog input channel and start A/D conversion in high speed / channel scan mode.

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

### **3.13 Watchdog Timer (Runaway Detecting Timer)**

TMP95C061 is containing watchdog timer of Runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the

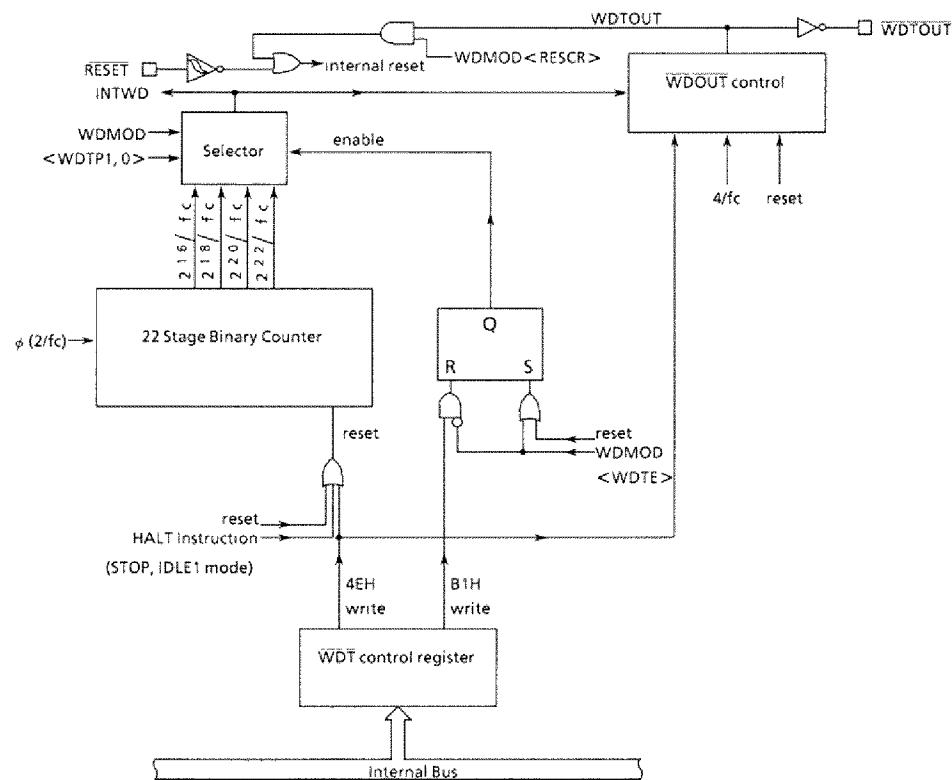
watchdog timer detects a malfunction, it generates a non-maskable interrupt to notify the CPU of the malfunction, and outputs 0 externally from watchdog timer out pin WDTOUT to notify the peripheral devices of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

### 3.13.1 Configuration

(WDT).

Figure 3.13 (1) shows the block diagram of the watchdog timer



**Figure 3.13 (1). Block Diagram of Watchdog Timer**

The watchdog timer is a 22-stage binary counter which uses  $\phi$  (fc/2) as the input clock. There are four outputs from the binary counter:  $2^{16}/fc$ ,  $2^{18}/fc$ ,  $2^{20}/fc$ , and  $2^{22}/fc$ . Selecting one of the outputs with the WDMOD register generates a watchdog interrupt, and outputs watchdog timer out when an overflow occurs.

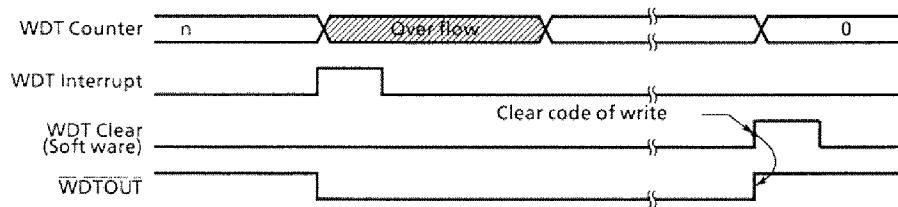
Since the watchdog timer out pin (WDTOUT) outputs "0" due to a watchdog timer overflow, the peripheral devices can be reset. The watchdog timer out pin is set to "1" after disabling WDT and clearing the watchdog timer (by writing a clear code 4EH in the WDCR register).

(Example)

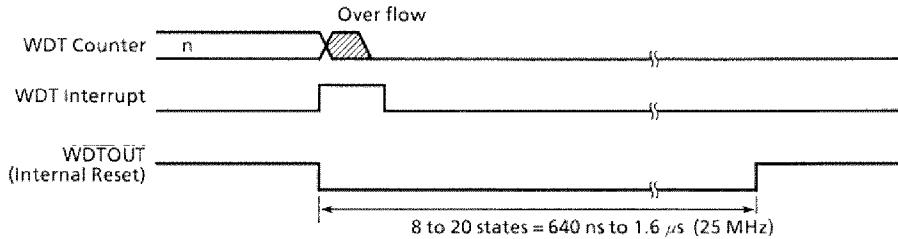
LDW	(WDMOD), B100H	; disable
LD	(WDCR), 4EH	; write clear
SET	7, (WDMOD)	; enable again

In other words, the WDTOUT continues to output "0" until the clear code is written.

The watchdog timer out pin (WDTOUT) outputs 0 to 8 to 20 states (640ns to 1.6 $\mu$ s @ 25MHz) and resets itself.



**Figure 3.13 (2). Normal Mode**



**Figure 3.13 (3). Reset Mode**

### 3.13.2 Control Registers

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

(1) Watchdog Timer Mode Register (WDMOD)

- ① Setting the detecting time of watchdog timer <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD <WDTP1, 0> = 00 when reset, and therefore  $2^{16}/f_{\text{Sys}}$  is set. (The number of states is approximately 32,768).

- ② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD <WDTE> is initialized to "1" to enable the watchdog timer.

- Disable control

WDMOD	$\leftarrow$	0	-	-	-	-	-	x	x
WDCR	$\leftarrow$	1	0	1	1	0	0	0	1

Clear WDMOD <WDTE> to "0".

Write the disable code (B1H).

- Enable control

Set WDMOD <WDTE> to "1".

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

- Watchdog timer clear control

WDCR	$\leftarrow$	0	1	0	0	1	1	1	0
------	--------------	---	---	---	---	---	---	---	---

Write the clear code (4EH).

To disable, it is necessary to clear this bit to "0" and write the disable code (B1H) in the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.

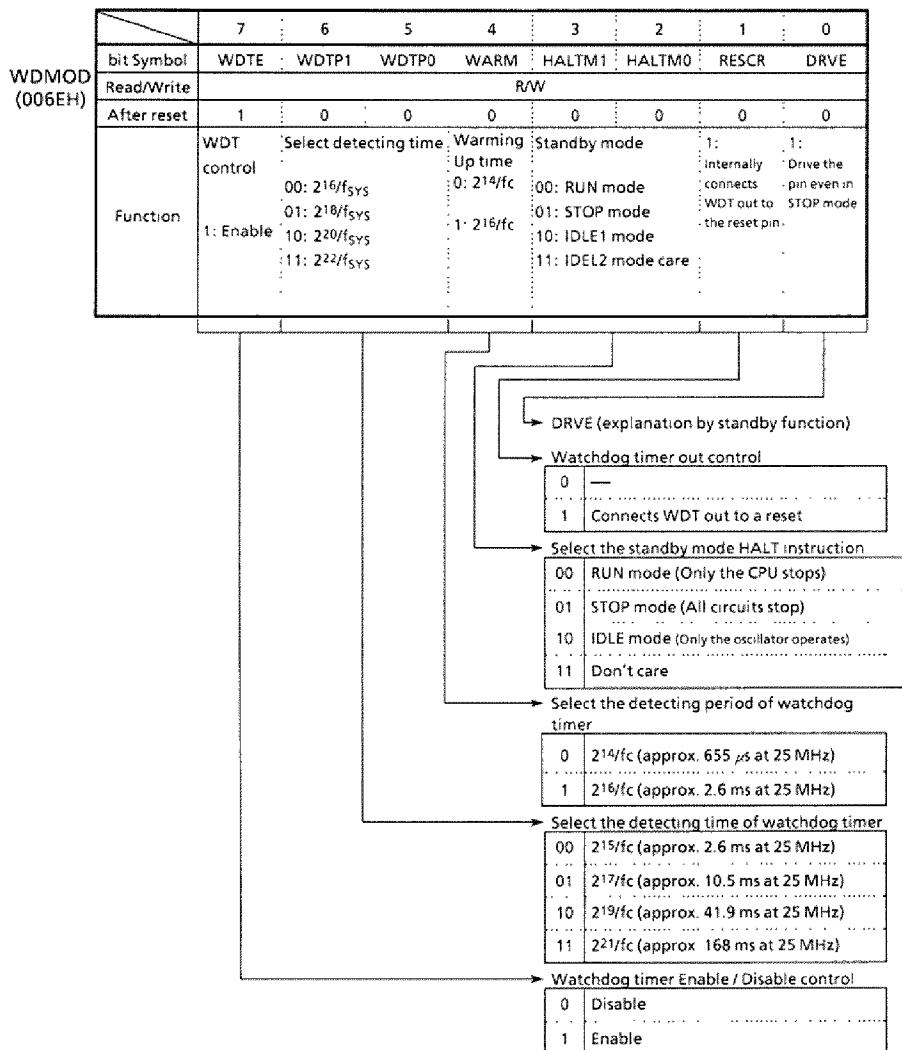
However, it is possible to return from the disable state to enable state by merely setting <WDTE> to "1".

- ③ Watchdog timer out reset connection <RESCR>

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

(2) Watchdog Timer Control Register (WDCR)

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



**Figure 3.13 (4). Watchdog Timer Mode Register**

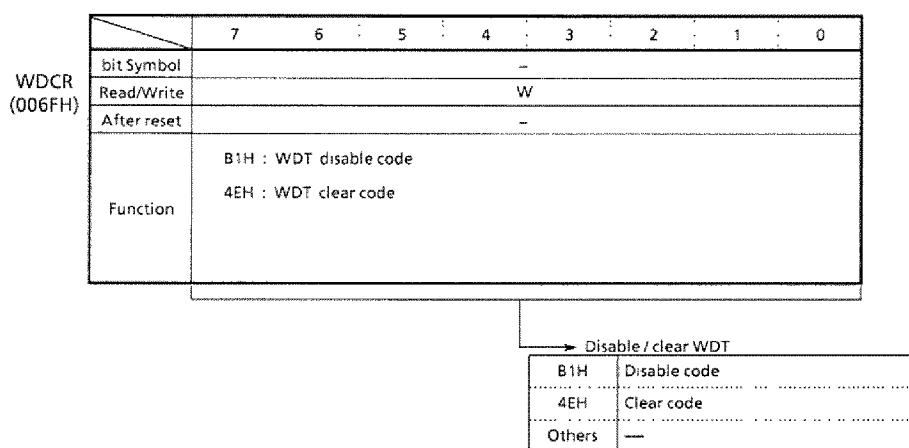


Figure 3.13 (5). Watchdog Timer Control Register

### 3.13.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD <WDTP1, 0> register and outputs a low level signal. The watchdog timer must be zero-cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (runaway) due to the INTWD Interrupt and it is possible to return to normal oper-

ation by an anti-malfunction program. By connecting the watchdog timer out pin to peripheral devices' resets, a CPU malfunction can also be acknowledged to other devices.

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE and STOP modes. In the RUN mode, the watchdog timer is enabled.

However, the function can be disabled when entering the RUN 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  $2^{18}/fc$

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 IDLE 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:  $2^{16}/fc$ )

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

(note)X : don't care   - : no change

### 3.14 Bus Release Function

The TMP95C061 supports a bus request pin (BUSRQ: also used as P53) and a bus acknowledge pin (BUSAk: also used as P54). Set these pins using P5CR and P5FC.

### 3.14 (1) Operation Description

When 0 is input to the BUSRQ pin, the TMP95C061 acknowledges a bus request. When the current bus cycle ends, the TMP95C061 sets the address bus (A23 to A0) and bus control signals (RD, WR, R/W, CS0 to 3) to high, then sets these signals and the data bus (D15 to D0) output buffer to off, sets the

BUSAk pin to low to indicate the bus is released. For bus release timing and DRAM dedicated pin state when the DRAM controller is in use, see 3.7 (5) Bus release mode.

During bus release, the TMP95C061 cannot access internal I/Os and internal I/Os keep functioning. Therefore, the watchdog timer continues counting. To use the bus release function, set runaway detect time with bus release time in consideration.

### 3.14 (2) Pin States as Bus Release

Table 3.14 shows pin states at bus release.

**Table 3.14 Pin States as Bus Release**

Pin name	PIN status as bus release	
	Port mode	Function mode
D0 to D7	---	Becomes high impedance.
P10 to P17 (D8 to 15)	No status change.	Becomes high impedance.
P20 to P27 (A16 to A23)	No status change.	First sets all bits to high, then sets them to high impedance.
A0 to A15 <u>RD</u> <u>WR</u>	---	First sets all bits to high, then sets them to high impedance.
P52 (HWR) P55 (R/ <u>W</u> )	No status change.	First sets all bits to high, then sets output buffer to off. Internal pull-up is added regardless of output latch value.
P60 (CS0) P61 (CS1) P62 (CS2) P63 (CS3)	No status change.	First sets all bits to high, then sets them to high impedance.

For P63 (CAS), P64 (RAS), P65 (REFOUT), see the description of "3.7 (5) Bus release mode".

## 4. Electrical Characteristics

### 4.1 Absolute Maximum

Symbol	Parameter	Rating	Unit
$V_{CC}$	Power Supply Voltage	-0.5 ~ 6.5	V
$V_{IN}$	Input Voltage	-0.5 ~ $V_{CC} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	102	mA
$\Sigma I_{OH}$	Output Current (total)	-120	mA
$PD$	Power Dissipation ( $T_a = 70^\circ C$ )	600	mW
$T_{SOLDER}$	Soldering Temperature (10s)	260	°C
$T_{STG}$	Storage Temperature	-65 ~ 150	°C
$T_{OPR}$	Operating Temperature	-20 ~ 70	°C

### 4.2 DC Characteristics

$V_{CC} = 5V \pm 10\%$ ,  $T_a = -20$  to  $70^\circ C$  (8 to 25MHz) (Typical values are for  $T_a = 25^\circ C$  and  $V_{CC} = 5V$  unless otherwise noted)

Symbol	Parameter	Min	Max	Unit	Test Condition
$V_{IL}$	Input Low Voltage (AD0-15)	-0.3	0.8	V	
$V_{IL1}$	P5, P7, P8, P9, PA, PB	-0.3	$0.3V_{CC}$	V	
$V_{IL2}$	RESET, NMI, INTO (PB7)	-0.3	$0.25V_{CC}$	V	
$V_{IL3}$	$\overline{EA}$ , AM8/ $\overline{16}$	-0.3	0.3	V	
$V_{IL4}$	X1	-0.3	$0.2V_{CC}$	V	
$V_{IH}$	Input High Voltage (D0 to 15)	2.2	$V_{CC} + 0.3$	V	
$V_{IH1}$	P5, P7, P8, P9, PA, PB	$0.7V_{CC}$	$V_{CC} + 0.3$	V	
$V_{IH2}$	RESET, NMI, INTO (PB7)	$0.75V_{CC}$	$V_{CC} + 0.3$	V	
$V_{IH3}$	$\overline{EA}$ , AM8/ $\overline{16}$	$V_{CC} - 0.3$	$V_{CC} + 0.3$	V	
$V_{IH4}$	X1	$0.8V_{CC}$	$V_{CC} + 0.3$	V	
$V_{OL}$	Output Low Voltage		0.45	V	$I_{OL} = 1.6mA$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -400\mu A$
$V_{OH1}$		$0.75V_{CC}$		V	$I_{OH} = -100\mu A$
$V_{OH2}$		$0.9V_{CC}$		V	$I_{OH} = -20\mu A$
$I_{DAR}$	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	$V_{EXT} = 1.5V$ $R_{EXT} = 1.1K\Omega$
$I_{LI}$	Input Leakage Current	0.02 (Typ)	$\pm 5$	$\mu A$	$0.0 \leq V_{in} \leq V_{CC}$
$I_{LO}$	Output Leakage Current	0.05 (Typ)	$\pm 10$	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
$I_{cc}$	Operating Current (RUN)	26 (Typ)	50	mA	$f_C = 25MHz$
	IDLE	1.7 (Typ)	10	mA	
	STOP ( $T_a = -20$ ~ $70^\circ C$ )	0.2 (Typ)	50	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
	STOP ( $T_a = 0$ ~ $50^\circ C$ )		10	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
$V_{STOP}$	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	$V_{IL2} = 0.2V_{CC}$ $V_{IH2} = 0.8V_{CC}$
$R_{RST}$	RESET Pull Up Register	50	150	$K\Omega$	
$C_{IO}$	Pin Capacitance		10	pF	$f_C = 1MHz$
$V_{TH}$	Schmitt Width RESET, NMI, INTO (PB7)	0.4	1.0 (Typ)	V	
$R_K$	Pull Down/Up Register	50	150	$K\Omega$	

Note: I-DAR is guaranteed for a total of up to 8 ports.

## 4.3 AC Electrical Characteristics

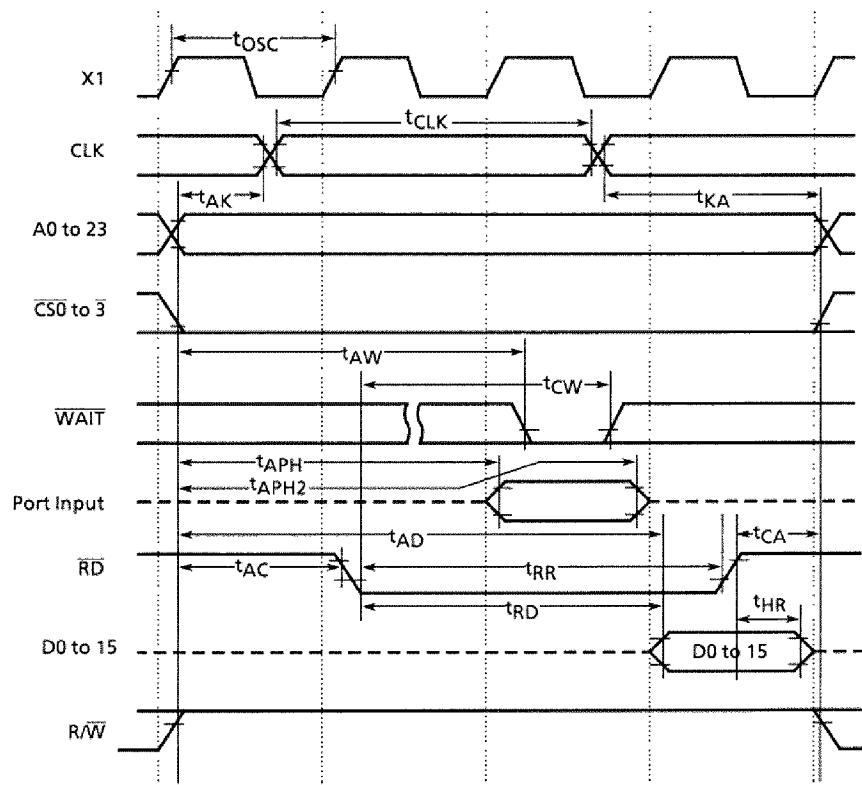
 $V_{cc} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ C$  (8MHz to 25MHz)

No.	Symbol	Parameter	Variable		20MHz		25MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$t_{OSC}$	Osc. Period (=x)	40	125	50		40		ns
2	$t_{CLK}$	CLK width	2x - 40		85		40		ns
3	$t_{AK}$	A0 to 23 Valid $\rightarrow$ CLK Hold	0.5x - 20		11		0		ns
4	$t_{KA}$	CLK Valid $\rightarrow$ A0 to 23 Hold	1.5x - 60		24		0		ns
5	$t_{AC}$	A0 to 23 Valid $\rightarrow$ RD/WR fall	0.5x - 20		16		20		ns
6	$t_{CA}$	RD/WR rise $\rightarrow$ A0 to 23 Hold	0.5x - 20		11		0		ns
7	$t_{AD}$	A0 to 23 Valid $\rightarrow$ D0 - 15 input		3.0x - 35		140		105	ns
8	$t_{RD}$	RD fall $\rightarrow$ D0 - 15 input		3.5x - 40		85		60	ns
9	$t_{RR}$	RD Low width	2.0x - 40		85		60		ns
10	$t_{HR}$	RD rise $\rightarrow$ D0 to 15 Hold	0		0		0		ns
11	$t_{WW}$	WR Low width	2.5x - 40		85		60		ns
12	$t_{DW}$	D0 to 15 Valid $\rightarrow$ WR rise	2.0x - 40		60		40		ns
13	$t_{WD}$	WR rise $\rightarrow$ D0 to 15 Hold	0.5x - 10		15		10		ns
14	$t_{AW}$	A0 to 23 Valid $\rightarrow$ WAIT input (1WAIT + n mode)		3.5x - 90		85		50	ns
15	$t_{CW}$	WR Low width	2.5x - 0		125		100		ns
16	$t_{APH}$	A0 to 23 Valid $\rightarrow$ PORT input		2.5x - 90		35		10	ns
17	$t_{APH2}$	A0 to 23 Valid $\rightarrow$ PORT Hold	2.5x - 50		175		150		ns
18	$t_{CP}$	WR rise $\rightarrow$ PORT Valid		200		200		200	ns

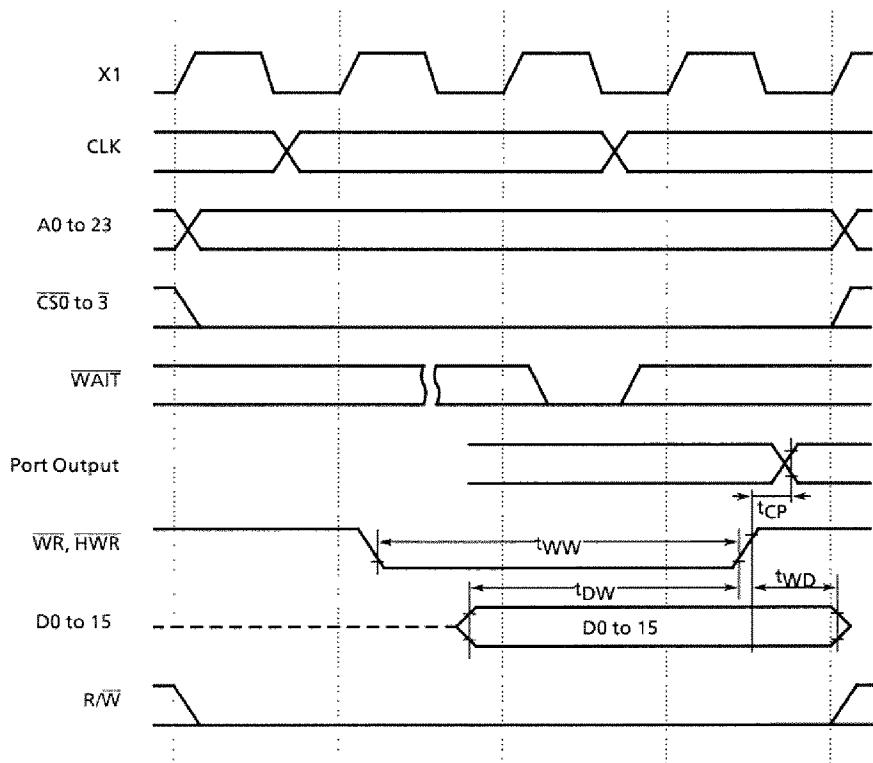
## AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF  
(However, D0 to D15, A0 to A23, ALE, RD, WR, HWR, R/W, CLK, CS0 to CS3, CL = 100pF)
- Input Level: High 2.4V /Low 0.45V (D0 to D15)  
High 0.8Vcc /Low 0.2Vcc (except for D0 to D15)

## (1) Read Cycle



## (2) Write Cycle



## 4.4 DRAM Control AC Characteristics

 $V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C (8MHz to 25MHz)

No.	Symbol	Parameter	Variable		20MHz		25MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$t_{RC}$	$\overline{RAS}$ cycle time	4x		200		160		ns
2	$t_{RAC}$	$\overline{RAS}$ access time		3x - 40		110		80	ns
3	$t_{CAC}$	$\overline{CAS}$ access time		1.5x - 25		40		25	ns
4	$t_{AA}$	Column address access time		2.5x - 35		70		45	ns
5	$t_{OFF}$	Input data hold time	0		0		0		ns
6	$t_{RP}$	$\overline{RAS}$ precharge time	1.5x - 10		65		50		ns
7	$t_{RAS}$	$\overline{RAS}$ low pulse width	2.5x - 30		65		70		ns
8	$t_{RSH}$	$\overline{RAS}$ hold time	1x - 15		65		25		ns
9	$t_{CSH}$	$\overline{CAS}$ hold time	3x - 35		65		85		ns
10	$t_{CAS}$	$\overline{CAS}$ low pulse width	1.5x - 15		65		45		ns
11	$t_{RCD}$	$\overline{RAS}$ - $\overline{CAS}$ delay time	1.5x - 40	1.5x	35	75	20	60	ns
12	$t_{RAD}$	$\overline{CAS}$ column address delay time	0.5x - 5	0.5x - 20	20	45	15	40	ns
13	$t_{CRP}$	$\overline{RAS}$ - $\overline{CAS}$ precharge time	1x - 35		15		5		ns
14	$t_{CP}$	$\overline{CAS}$ precharge time	2.5x - 35		90		65		ns
15	$t_{ASR}$	Low address setup time	0.5x - 15		10		5		ns
16	$t_{RAH}$	Low address hold time	0.5x - 5		20		15		ns
17	$t_{ASC}$	Column address setup time	1x - 25		25		15		ns
18	$t_{CAH}$	Column address hold time	2x - 35		65		45		ns
19	$t_{RAL}$	Column address $\overline{RAS}$ read time	2x - 30		70		50		ns
20	$t_{CWL}$	Write command $\overline{CAS}$ read time	2.5x - 35		90		65		ns
21	$t_{DS}$	Data output setup time	0.5x - 5		10		5		ns
22	$t_{DH}$	Data output hold time	2x - 35		65		45		ns
23	$t_{WCS}$	Write command setup time	1x - 30		20		10		ns
24	$t_{CHR^*1}$	$\overline{CAS}$ hold time	2x - 50		50		30		ns
25	$t_{RPC^*}$	$\overline{RAS}$ precharge $\overline{CAS}$ active time	1.5x - 30		45		30		ns
26	$t_{CSR^*}$	$\overline{CAS}$ setup time	0.5x - 10		15		10		ns
27	$t_{RPS^*2}$	$\overline{RAS}$ precharge time	6x - 50		250		190		ns
28	$t_{CHS^*2}$	$\overline{CAS}$ hold time	0		0		0		ns
29	$t_{CFL}$	Refresh setup time	1x - 5		45		35		ns
30	$t_{CFH}$	Refresh hold time	1x - 10		40		30		ns

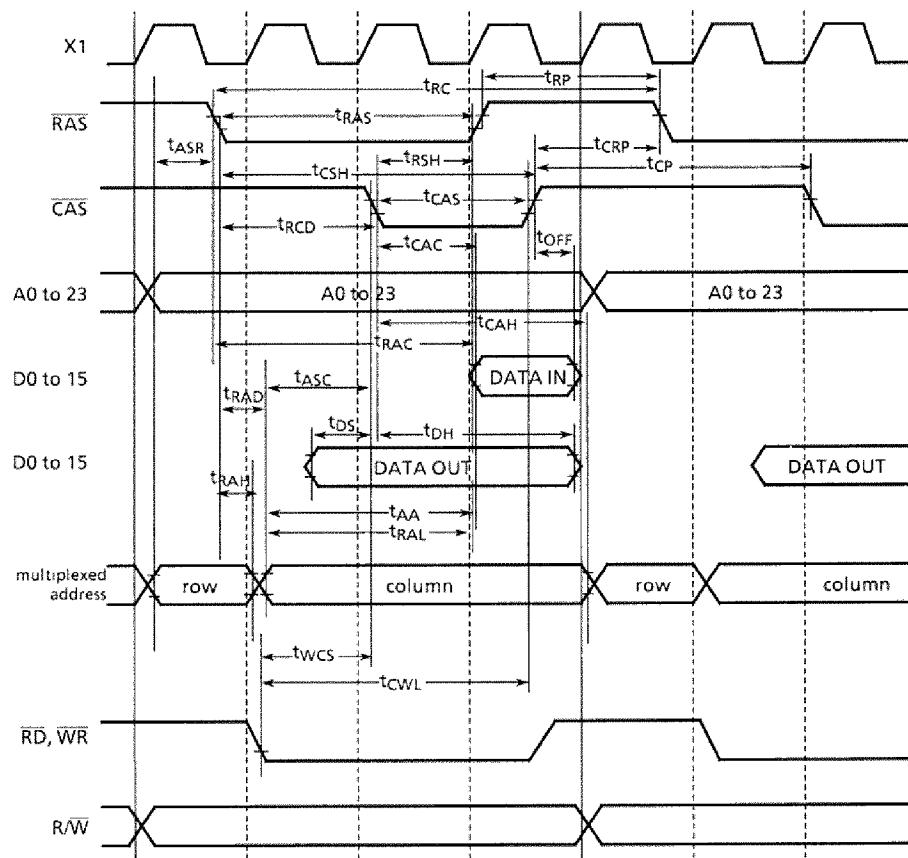
\*1  $\overline{CAS}$  before  $\overline{RAS}$  interval refresh mode\*2  $\overline{CAS}$  before  $\overline{RAS}$  self-refresh mode

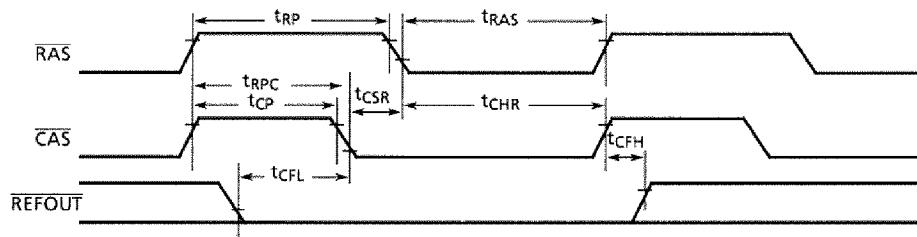
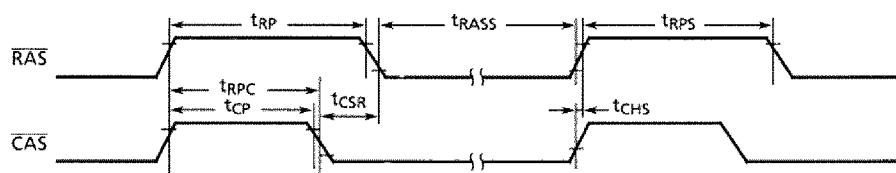
\* Both refresh modes

## AC Measuring Conditions

- Output Level: High 2.2V / Low 0.8V, CL50pF  
(However CL = 100pF for D0 to D15, A0 to A23, RD, WR, HWR, R/W  $\overline{RAS}$ )
- Input Level: High 2.4V / Low 0.45V (AD0 ~ AD15)  
High 0.8Vcc/Low 0.2Vcc (except for D0 to D15)

## (1) Read/Write Access Cycle



(2) CAS Before RAS Interval Refresh Cycle(3) CAS Before RAS Self-Refresh Cycle

**4.5 A/D Conversion Characteristics** $V_{CC} = 5V \pm 10\% \ TA = -20 \sim 70^\circ C \ (8 \text{ to } 25MHz)$ 

Symbol	Parameter		Min	Typ	Max	Unit
$V_{REF}$	Analog reference voltage		$V_{CC} - 1.5$	$V_{CC}$	$V_{CC}$	V
$A_{GND}$	Analog reference voltage		$V_{SS}$	$V_{SS}$	$V_{SS}$	
$V_{AIN}$	Analog input voltage range		$V_{SS}$		$V_{CC}$	
$I_{REF}$	Analog current for analog reference voltage			0.5	1.5	mA
Error (Quantize error of $\pm 0.5$ LSB not included)	$4 \leq f_C \leq 16MHz$	Slow mode		$\pm 1.5$	$\pm 4.0$	LSB
		Fast mode		$\pm 3.0$	$\pm 6.0$	
	$16 \leq f_C \leq 25MHz$	Slow mode		$\pm 1.5$	$\pm 4.0$	
		Fast mode		$\pm 4.0$	$\pm 8.0$	

**4.6 Serial Channel Timing - I/O Interface Mode** $V_{CC} = 5V \pm 10\% \ TA = -20 \text{ to } 70^\circ C \ (8 \text{ to } 25MHz)$ 

## (1) SCLK Input Mode

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{SCY}$	SCLK cycle	16x		0.8		0.64		μs
$t_{OSS}$	Output Data→rising edge of SCLK	$t_{SCY}/2 - 5x - 50$		100		70		ns
$t_{OHS}$	SCLK rising edge→output data hold	5x - 100		150		100		ns
$t_{HSR}$	SCLK rising edge→input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge→effective data input		$t_{SCY} - 5x - 100$		450		340	ns

 $V_{CC} = 5V \pm 10\% \ TA = -20 \text{ to } 70^\circ C \ (8 \text{ to } 25MHz)$ 

## (2) SCLK Output Mode

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{SCY}$	SCLK cycle (programmable)	16x	$8192x$	0.8	512	0.64	409.6	μs
$t_{OSS}$	Output Data→rising edge of SCLK	$t_{SCY} - 2x - 150$		550		410		ns
$t_{OHS}$	SCLK rising edge→output data hold	$2x - 80$		20		0		ns
$t_{HSR}$	SCLK rising edge→input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge→effective data input		$t_{SCY} - 2x - 150$		550		410	ns

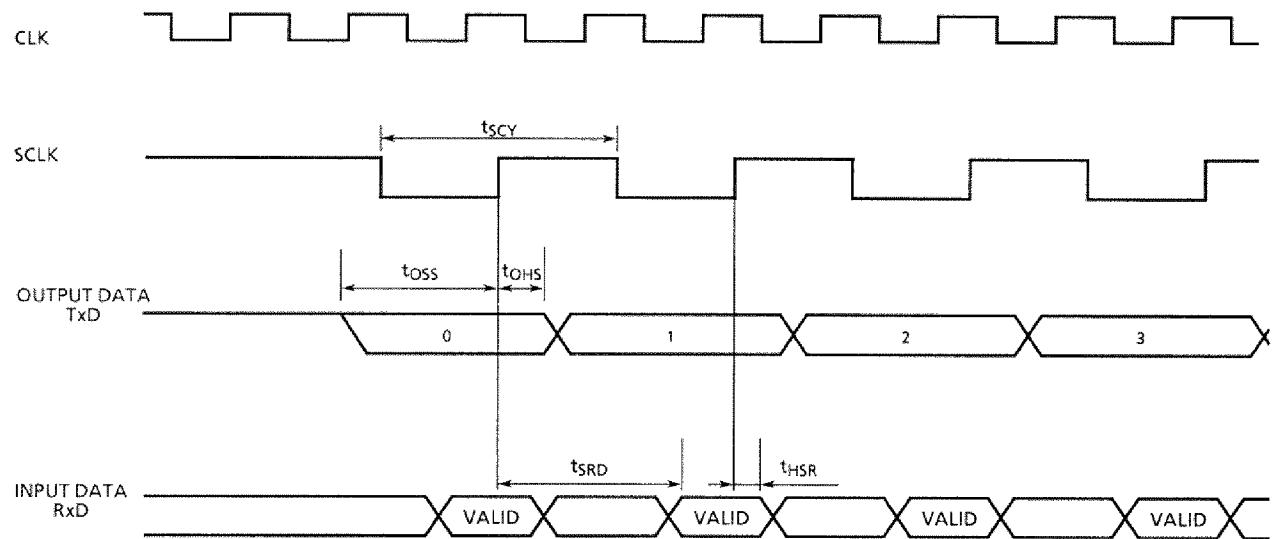
**4.7 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7)** $V_{CC} = 5V \pm 10\% \ TA = -20 \text{ to } 70^\circ C \ (8 \text{ to } 25MHz)$ 

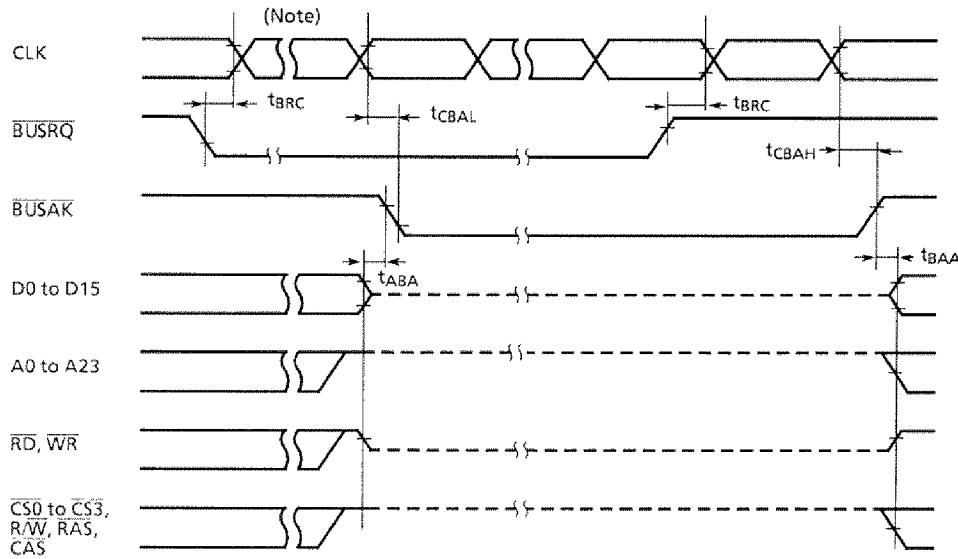
Symbol	Parameter	Variable		20MHz		25MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{VCK}$	Clock cycle	$8x + 100$		500		420		ns
$t_{VCKL}$	Low level clock pulse width	$4x + 40$		240		200		ns
$t_{VCKH}$	High level clock pulse width	$4x + 40$		240		200		ns

**4.8 Interrupt Operation** **$V_{cc} = 5V \pm 10\%$   $T_a = -20$  to  $70^\circ C$  (8 to 25MHz)**

<b>Symbol</b>	<b>Parameter</b>	<b>Variable</b>		<b>20MHz</b>		<b>25MHz</b>		<b>Unit</b>
		<b>Min</b>	<b>Max</b>	<b>Min</b>	<b>Max</b>	<b>Min</b>	<b>Max</b>	
$t_{INTAL}$	NMI, INTO Low level pulse width	4x		200		160		ns
$t_{INTAH}$	NMI, INTO High level pulse width	4x		200		160		ns
$t_{INTBL}$	INT4 ~ INT7 Low level pulse width	$8x + 100$		500		420		ns
$t_{INTBH}$	INT4 ~ INT7 High level pulse width	$8x + 100$		500		420		ns

#### 4.9 Timing Chart for I/O Interface Mode



4.10 Timing Chart for Bus Request (BUSRQ)/BUS Acknowledge (BUSAK)

Symbol	Parameter	Variable		20MHz		25MHz		Unit
		Min	Max	Min	Max	Min	Max	
t <sub>BRC</sub>	BUSRQ setup time for CLK	120		120		120		ns
t <sub>CBAL</sub>	CLK→BUSAK falling edge		1.5x + 120		220		200	ns
t <sub>CBAH</sub>	CLK→BUSAK rising edge		0.5x + 40		65		60	ns
t <sub>ABA</sub>	Floating time to BUSAK fall	0	80	0	80	0	80	ns
t <sub>BAA</sub>	Floating time to BUSAK rise	0	80	0	80	0	80	ns

Note: The bus will be released after the WAIT request is inactive, when the BUSRQ is set to "0" during "wait" cycle.

## 5. Table of Special Function Registers (SFRs)

### (SFR; Special Function Register)

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

- (1) I/O port
- (2) I/O port control

- (3) Timer control
- (4) Pattern Generator control
- (5) Watch Dog Timer control
- (6) Serial Channel control
- (7) A/D converter control
- (8) Interrupt control
- (9) Chip Select/Wait Control
- (10) DRAM Control

Configuration of the table

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

**Table 5 I/O Register Address Map**

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
000000H		20H	TRUN	40H	TREG6L	60H	ADREG0L
1H	P1	21H		41H	TREG6H	61H	ADREG0H
2H		22H	TREG0	42H	TREG7L	62H	ADREG1L
3H		23H	TREG1	43H	TREG7H	63H	ADREG1H
4H	P1CR	24H	T01MOD	44H	CAP3L	64H	ADREG2L
5H		25H	TFFCR	45H	CAP3H	65H	ADREG2H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG3L
7H		27H	TREG3	47H	CAP4H	67H	ADREG3H
8H	P2FC	28H	T23MOD	48H	T5MOD	68H	B0CS
9H		29H	TRDC	49H	T5FFCR	69H	B1CS
AH		2AH		4AH		6AH	B2CS
BH		2BH		4BH		6BH	B3CS
CH		2CH	PACR	4CH	PG0REG	6CH	BEXCS
DH	P5	2DH	PAFC	4DH	PG1REG	6DH	ADMOD
EH		2EH	PBCR	4EH	PG01CR	6EH	WDMOD
FH		2FH	PBFC	4FH		6FH	WDCR
10H	PSCR	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H	PSFC	31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BROCR	73H	INTET10
14H		34H	CAP1L	54H	SC1BUF	74H	INTET32
15H	P6FC	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P7CR	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	INTETC01
1AH	PBCR	3AH	T45CR	5AH	DREFCR	7AH	INTETC23
1BH	PBFC	3BH		5BH	DMEMCR	7BH	IIMC
1CH		3CH	MSAR0	5CH	MSAR2	7CH	DMA0V
1DH		3DH	MAMR0	5DH	MAMR2	7DH	DMA1V
1EH	PA	3EH	MSAR1	5EH	MSAR3	7EH	DMA2V
1FH	PB	3FH	MAMR1	5FH	MAMR3	7FH	DMA3V

## (1) I/O Port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
R/W										
Input mode										
			0	0	0	0	0	0	0	0
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
R/W										
Output mode										
			1	1	1	1	1	1	1	1
PS	PORTS	0DH			P55	P54	P53	P52		RDE
R/W										
Input mode (Pulled-up)										
			1	1	1	1	1			1
P6	PORT6	12H			P65	P64	P63	P62	P61	P60
R/W										
Output mode										
			1	1	1	1	0	1	1	1
P7	PORT7	13H	P77	P76	P75	P74	P73	P72	P71	P70
R/W										
Input mode (Pulled-up)										
			1	1	1	1	1	1	1	1
P8	PORT8	18H			P85	P84	P83	P82	P81	P80
R/W										
Input mode (Pulled-up)										
			1	1	1	1	1	1	1	1
P9	PORT9	19H					P93	P92	P91	P90
R										
Input mode										
PA	PORTA	1EH					PA3	PA2	PA1	PA0
R/W										
Input mode (Pulled-up)										
			1	1	1	1	1	1	1	1
PB	PORTB	1FH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W										
Input mode (Pulled-up)										
			1	1	1	1	1	1	1	1

Note : Clearing "RDE" to "0" outputs the RD strobe from RD pin (for PSRAM), even when the internal address is accessed.

If "RDE" remains "1", the RD strobe is output only when the external address is accessed.

## Read/Write

R/W ; Either read or write possible

R ; Only read is possible

W ; Only write is possible

Prohibit RMW ; Prohibit Read Modify Write (Prohibit RES / SET / TSET / CHG / STCF / ANDCF / ORCF / XORCF Instruction)

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

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	PORT1 Control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
							W			
			0	0	0	0	0	0	0	0
P2FC	PORT2 Function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
							W			
			0	0	0	0	0	0	0	0
P5CR	PORT5 Control	10H (Prohibit RMW)			P55C	P54C	P53C	P52C		
							W			
					0	0	0	0		
P5FC	PORT5 Function	11H (Prohibit RMW)			0 : IN	1 : OUT				
					P55F	P54F	P53F	P52F		
					0 : PORT	0 : PORT	0 : PORT	0 : PORT		
P6FC	PORT6 Function	15H (Prohibit RMW)			1 : R/W	1 : BUSAK	1 : BUSRQ	1 : HWR		
					P65F	P64F	P63F	P62F	P61F	P60F
							W			
P7CR	PORT7 Control	16H (Prohibit RMW)			0	0	0	0	0	0
					0 : PORT	1 : CS/CAS, RAS, REFOUT				
			P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
P7FC	PORT7 Function	17H (Prohibit RMW)					W			
			P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
			0	0	0	0	0	0	0	0
P8CR	PORT8 Control	1AH (Prohibit RMW)			0 : PORT	1 : PG1-OUT		0 : PORT	1 : PG0-OUT	
					P85C	P84C	P83C	P82C	P81C	P80C
					0 : IN	1 : OUT				
P8FC	PORT8 Function	1BH (Prohibit RMW)			P85F	P83F	P82F			P80F
					W	W	W			W
					0	0	0			0
					0 : PORT	0 : PORT	0 : PORT	0 : PORT		
					1 : 5CLK1	1 : TxD1	1 : SCLK0		1 . TxD0	

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

Symbol	Name	Address	7	6	5	4	3	2	1	0
PACR	PORTA Control	2CH (Prohibit RMW)					PA3C	PA2C	PA1C	PA0C
								W		
							0	0	0	0
								0 : IN	1 : OUT	
PAFC	PORTA Function	2DH (Prohibit RMW)					PA3F	PA2F		
								W		
							0	0		
							0 : PORT	0 : PORT		
							1 : TO3	1 : TO1		
PBCR	PORTB Control	2EH (Prohibit RMW)	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
							W			
			0	0	0	0	0	0	0	0
							0 : IN	1 : OUT		
PBFC	PORTB Function	2FH (Prohibit RMW)		PB6F			PB3F	PB2F		
				W			W	W		
				0			0	0		
				0 : PORT			0 : PORT	0 : PORT		
				1 : TO6			1 : TO5	1 : TO4		

## (3) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
			PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W				R/W			
			0		0	0	0	0	0	0
			Prescaler & Timer Run / Stop CONTROL							
			0 : Stop & Clear 1 : Run (Count up)							
TRUN	Timer Control	20H								
TREG0	8 bit Timer Register 0	22H (Prohibit RMW)								
TREG1	8 bit Timer Register 1	23H (Prohibit RMW)								
T01 MOD	8 bit Timer 0,1 Source CLK & MODE	24H (Prohibit RMW)	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
TFFCR	8 bit Timer Flip-Flop Control	25H (Prohibit RMW)	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
TREG2	PWM Timer Register 2	26H (Prohibit RMW)								
TREG3	PWM Timer Register 3	27H (Prohibit RMW)								
T23 MOD	8 bit Timer 2,3 Source CLK & MODE	28H (Prohibit RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
TRDC	Timer Reg. Double Buffer Control Reg.	29H							TR2DE	TR0DE
									R/W	
									0	0
									0 : Double Buffer Disable 1 : Double Buffer Enable	

## (3) Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4L	16 bit Timer Register4L (Prohibit RMW)	30H	-		-		-		-	
			W		W		W		W	
			Undefined		Undefined		Undefined		Undefined	
TREG4H	16 bit Timer Register4H (Prohibit RMW)	31H	-		-		-		-	
			W		W		W		W	
			Undefined		Undefined		Undefined		Undefined	
TREG5L	16 bit Timer Register5L (Prohibit RMW)	32H	-		-		-		-	
			W		W		W		W	
			Undefined		Undefined		Undefined		Undefined	
TREG5H	16 bit Timer Register5H (Prohibit RMW)	33H	-		-		-		-	
			W		W		W		W	
			Undefined		Undefined		Undefined		Undefined	
CAP1L	Capture Register1L	34H	-		-		-		-	
			R		R		R		R	
			Undefined		Undefined		Undefined		Undefined	
CAP1H	Capture Register1H	35H	-		-		-		-	
			R		R		R		R	
			Undefined		Undefined		Undefined		Undefined	
CAP2L	Capture Register2L	36H	-		-		-		-	
			R		R		R		R	
			Undefined		Undefined		Undefined		Undefined	
CAP2H	Capture Register2H	37H	-		-		-		-	
			R		R		R		R	
			Undefined		Undefined		Undefined		Undefined	
T4MOD	16 bit Timer 4 Source CLK & MODE (Prohibit RMW)	38H	CAP2TS : EQ5TS	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0	
			R/W	W			R/W			
			0 : 0	1	0	0	0	0	0	0
T4FFCR	16 bit Timer 4 Flip-Flop Control (Prohibit RMW)	39H	TFF5C1 : TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0	
			W		R/W		W			
			0 : 0	0	0	0	0	0	0	0
T45CR	T4, T5 Control	3AH	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		
			R/W			R/W				
			0		0	0	0	0	0	0
			Fix at "0"			PG1shift trigger	PG0shift trigger	1 : Double Buffer Enable		
						0 : timer2, 3 1 : timer5	0 : timer0, 1 1 : timer4			

## (3) Timer Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG6L	16bit Timer Register6L (Prohibit RMW)	40H				—				
						W				
						Undefined				
TREG6H	16 bit Timer Register6H (Prohibit RMW)	41H				—				
						W				
						Undefined				
TREG7L	16 bit Timer Register7L (Prohibit RMW)	42H				—				
						W				
						Undefined				
TREG7H	16 bit Timer Register7H (Prohibit RMW)	43H				—				
						W				
						Undefined				
CAP3L	Capture Register3L	44H				—				
						R				
						Undefined				
CAP3H	Capture Register3H	45H				—				
						R				
						Undefined				
CAP4L	Capture Register4L	46H				—				
						R				
						Undefined				
CAP4H	Capture Register4H	47H				—				
						R				
						Undefined				
T5MOD	16 bit Timer 5 Source CLK & MODE	48H				CAP3IN	CAP34M1 : CAP34M0	CLE	T5CLK1	T5CLK0
						W	R/W			
						1	0	0	0	0
T5FFCR	16 bit Timer 5 Flip-Flop Control	49H					Capture Timming	Source Clock		
							0 : Soft-Capture	00 : Disable	00 : T16	
							01 : T16 ↑ T17 ↑	1 : UCS	01 : gT1	
							1 : Don't care	10 : T16 ↑ T16 ↓	Clear	10 : gT4
							11 : TFF1 ↑ TFF1 ↓	Enable	11 : gT16	
						CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1
							R/W			W
							0	0	0	0
							TFF6 Invert Trigger	00 : Invert TFF6		
							0 : Trigger Disable	01 : Set TFF6		
							1 : Trigger Enable	10 : Clear TFF6		
								11 : Don't care		

**(4) Pattern Generator**

Symbol	Name	Address	7	6	5	4	3	2	1	0
PGOREG	PG0 Register	4CH (Prohibit RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			0	0	0	0	W	R/W	Undefined	
			0	0	0	0	W	R/W	Undefined	
PG1REG	PG1 Register	4DH (Prohibit RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			0	0	0	0	W	R/W	Undefined	
			PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
PG01CR	PG0, 1 Control	4EH	0	0	0	0	0	0	0	0
			0: 8-bit write	0: Normal Rotation	0: 4-bit Step	PG1 trigger input	0: 8-bit write	0: Normal Rotation	0: 4-bit Step	PG0 trigger input
			1: 4-bit write	1: Reverse Rotation	1: 8-bit Step	enable	1: 4-bit write	1: Reverse Rotation	1: 8-bit Step	enable

**(5) Watch Dog Timer**

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode	6EH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTMO	RESCR	DRVE
			1	0	0	0	0	0	0	0
			00: 2 <sup>16</sup> /fc	Warming up Time	Standby Mode	00: RUN Mode	1: Connect internally	1: Drive the pin		
WDCR	Watch Dog Timer Control Register	6FH (Prohibit RMW)	1: WDT Enable	01: 2 <sup>18</sup> /fc	10: 2 <sup>20</sup> /fc	0: 2 <sup>14</sup> /fc	01: STOP Mode	10: IDLE Mode	1: WDT out pin to mode	
			11: 2 <sup>22</sup> /fc	1: 2 <sup>16</sup> /fc	11: Don't care	11: 2 <sup>16</sup> /fc	11: Don't care	Reset Pin		
			—	—	—	—	—	—	—	—

B1H: WDT Disable Code 4EH: WDT Clear Code

## (6) Serial Channel

Symbol	Name	Address	7	6	5	4	3	2	1	0									
SC0BUF	Serial Channel 0 Buffer	50H	RB7 TB7	R86 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RBO TBO									
			R (Receiving) /W (Transmission)																
			Undefined																
SC0CR	Serial Channel 0 Control	51H	RB8 R	EVEN R/W	PE	OERR R (Cleared to 0 by reading)	PERR	FERR	SCLK	IOC									
			Receiving data bit 8	0: Odd 1: Even	1: Parity Enable	0 Overrun	0 Parity	0 Framing	0: SCLK0 1: SCLK0	0: Input SCLK0 pin									
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0									
			R/W																
			Undefined	0	0	0	0	0	0	0									
			Transmission data bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: I/O Interface mode 01: UART 7bit 10: UART 8bit 11: UART 9bit	00: TO2 Trigger 01: Baud rate generator 10: Internal clock φ1 11: Don't care											
			-		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0									
			R/W																
BR0CR	Baud Rate Control	53H	0	0	0	0	0	0	0	0									
			Fix at "0"		00: $\#T_0$ (4/fc) 01: $\#T_2$ (16/fc) 10: $\#T_8$ (64/fc) 11: $\#T_{32}$ (256/fc)	Set frequency divisor 0 to F ("1" prohibited)													
SC1BUF	Serial Channel 1 Buffen	54H	RB7 TB7	R86 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RBO TBO									
			R (Receiving) /W (Transmission)																
			Undefined																
SC1CR	Serial Channel 1 Control	55H	RB8 R	EVEN R/W	PE	OERR R (Cleared to 0 by reading)	PERR	FERR	SCLKS	IOC									
			Receiving data bit 8	0: Odd 1: Even	1: Parity Enable	0 Overrun	0 Parity	0 Framing	0: SCLK1 1: SCLK1	0: 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									
			Transmission data bit 8	Fix at "0"	1: Receive Enable	1: Wake up Enable	00: I/O Interface mode 01: UART 7bit 10: UART 8bit 11: UART 9bit	00: TO2 Trigger 01: Baud rate generator 10: Internal clock φ1 11: Don't care											
			-		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0									
			R/W																
BR1CR	Baud Rate Control	57H	0	0	0	0	0	0	0	0									
			Fix at "0"		00: $\#T_0$ (4/fc) 01: $\#T_2$ (16/fc) 10: $\#T_8$ (64/fc) 11: $\#T_{32}$ (256/fc)	Set frequency divisor 0 to F ("1" prohibited)													
ODE	Serial Open Drain Enable	58H	ODE1 : ODE0																
			R/W																
									0	0									
									1:P83 Open-drain	1:P80 Open-drain									

## (7) A/D Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
*1) AD REG0L	AD Result Reg 0 low	60H	ADR01	ADR00						
										R
			Undefined	1	1	1	1	1	1	1
AD REG0H	AD Result Reg 0 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
										R
*1) AD REG1L	AD Result Reg 1 low	62H	ADR11	ADR10						
										R
			Undefined	1	1	1	1	1	1	1
AD REG1H	AD Result Reg 1 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
										R
*1) AD REG2L	AD Result Reg 2 low	64H	ADR21	ADR20						
										R
			Undefined	1	1	1	1	1	1	1
AD REG2H	AD Result Reg 2 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
										R
*1) AD REG3L	AD Result Reg 3 low	66H	ADR31	ADR30						
										R
			Undefined	1	1	1	1	1	1	1
AD REG3H	AD Result Reg 3 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
										R
										Undefined
ADMOD	A/D Converter Mode reg	6DH	EOCF	ADB#	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0
			R							R/W
			0	0	0	0	0	0	0	0
			1: End	1: Busy	1: Repeat mode	1: Scan mode	1: Slow mode	1: START		Analog Input Channel Select

\*1) Data to be stored in A/D Result Reg Low are the lower 2 bits of the conversion result. The contents of the lower 6 bits of this register are always read as "1".

## (8) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTAD										
INTE-OAD	INTerrupt Enable 0 & A/D	70H	IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INT5										
INTE45	INTerrupt Enable 4/5	71H	I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INT7										
INTE67	INTerrupt Enable 6/7	72H	I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTT1 (Timer 1)										
INTET01	INTerrupt Enable Timer 1/0	73H	IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTT3 (Timer 3)										
INTET23	INTerrupt Enable Timer 3/2	74H	IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTR5 (TREG5)										
INTET45	INTerrupt Enable Treg 5/4	75H	IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTR7 (TREG7)										
INTET67	INTerrupt Enable Treg 7/6	76H	IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTX0										
INTES0	INTerrupt Enable Serial 0	77H	ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTX1										
INTES1	INTerrupt Enable Serial 1	78H	ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTC1										
INTETC01	INTerrupt Enable TC 0/1	79H	ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTTC3										
INTETC23	INTerrupt Enable TC 2/3	7AH	ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

Function (Write)			
IxxM2	IxxM1	IxxM0	
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
Function (Read)			
IxxC	Function (Read)	Function (Write)	
0	Indicate no interrupt request.	Clear interrupt request flag.	
1	Indicate interrupt request	----- Don't care -----	

## (8) Interrupt Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Control	7BH (Prohibit RMW)						IOIE	IOLE	NMIREE
								W	W	W
								0	0	0
								1: INT0 input enable	0: INT0 edge mode	1: Operate even at NMI rise edge
									1: INT0 level mode	
DMA0V	DMA 0 request Vector	7CH (Prohibit RMW)								HDMA0 Start vector
								DMA0V8	DMA0V7	DMA0V6
										DMA0V5
										DMA0V4
								W		
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)								HDMA1 Start vector
								DMA1V8	DMA1V7	DMA1V6
										DMA1V5
										DMA1V4
								W		
								0	0	0
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)								HDMA2 Start vector
								DMA2V8	DMA2V7	DMA2V6
										DMA2V5
										DMA2V4
								W		
								0	0	0
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)								HDMA3 Start vector
								DMA3V8	DMA3V7	DMA3V6
										DMA3V5
										DMA3V4
								W		
								0	0	0
										0

## (9) Chip Select/Wait Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS / WAIT control register	68H (Prohibit RMW)			BOE	-	BOBUS	B0C1	B0C0	
					W	-	W	W	W	
					0	-	0	0	0	
					0: B0CS	-	0: 16 BIT	00: 2WAIT		
					1: master		1: 8 BIT	01: 1WAIT		
					bit			10: 1WAIT + n		
								11: 0WAIT		
B1CS	Block 1 CS / WAIT control register	69H (Prohibit RMW)			B1E	-	B1BUS	B1W1	B1W0	
					W	-	W	W	W	
					0	-	0	0	0	
					0: B1CS	-	0: 16 BIT	00: 2WAIT		
					1: master		1: 8 BIT	01: 1WAIT		
					bit			10: 1WAIT + n		
								11: 0WAIT		
B2CS	Block 2 CS / WAIT control register	6AH (Prohibit RMW)			B2E	B2M	B2BUS	B2W1	B2W0	
					W	W	W	W	W	
					1	0	0	0	0	
					0: B2CS	0: 16M	0: 16 BIT	00: 2WAIT		
					1: master	area	1: 8 BIT	01: 1WAIT		
					bit	1: MREG	setting	10: 1WAIT + n		
								11: 0WAIT		
B3CS	Block 3 CS / WAIT control register	6BH (Prohibit RMW)			B3E	B3CAS	B3BUS	B3W1	B3W0	
					W	W	W	W	W	
					0	0	0	0	0	
					0: B3CS	0: CS3	0: 16 BIT	00: 2WAIT		
					1: master	output	1: 8 BIT	01: 1WAIT		
					bit	1: CAS	output	10: 1WAIT + n		
								11: 0WAIT		
BEXCS	External CS / WAIT control register	6CH (Prohibit RMW)			-	-	BEXBUS	BEXW1	BEXW0	
					-	-	W	W	W	
					-	-	0	0	0	
					-	-	0: 16 BIT	00: 2WAIT		
							1: 8 BIT	01: 1WAIT		
								10: 1WAIT + n		
								11: 0WAIT		
MSAR0	Memory Start Address Reg. 0	3CH	S23	S22	S21	S20	S19	S18	S17	S16
							R/W			
			1	1	1	1	1	1	1	1
							A23 to A16			
								Memory start address setting		
MAMR0	Memory Start Address Mask Reg. 0	3DH	V20	V19	V18	V17	V16	V15	V14~9	V8
							R/W			
			1	1	1	1	1	1	1	1
							0 : Comparison is valid			
							1 : Comparison is invalid			
MSAR1	Memory Start Address Reg. 1	3EH	S23	S22	S21	S20	S19	S18	S17	S16
							R/W			
			1	1	1	1	1	1	1	1
							A23 to A16			
								Memory start address setting		
MAMR1	Memory Start Address Mask Reg. 1	3FH	V21	V20	V19	V18	V17	V16	V15~9	V8
							R/W			
			1	1	1	1	1	1	1	1
							0 : Comparison is valid			
							1 : Comparison is invalid			

## (9) Chip Select/Wait Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0		
MSAR2	Memory Start Address Reg. 2	5CH	S23	S22	S21	S20	S19	S18	S17	S16		
			R/W									
			1	1	1	1	1	1	1	1		
			A23 to A16 Memory start address setting									
MAMR2	Memory Start Address Mask Reg. 2	5DH	V22	V21	V20	V19	V18	V17	V16	V15		
			R/W									
			1	1	1	1	1	1	1	1		
			0 : Comparison is valid 1 : Comparison is invalid									
MSAR3	Memory Start Address Reg. 3	5EH	S23	S22	S21	S20	S19	S18	S17	S16		
			R/W									
			1	1	1	1	1	1	1	1		
			A23 to A16 Memory start address setting									
MAMR3	Memory Start Address Mask Reg. 3	5FH	V22	V21	V20	V19	V18	V17	V16	V15		
			R/W									
			1	1	1	1	1	1	1	1		
			0 : Comparison is valid 1 : Comparison is invalid									

## (10) DRAM Control

Symbol	Name	Address	7	6	5	4	3	2	1	0		
DREFCR	Refresh Control Reg.	5AH	DMI	RS2	RS1	R50	RW2	RW1	RW0	RC		
			R/W									
			0	0	0	0	0	0	0	0		
			Dummy cycle	Refresh cycle insertion interval				Refresh cycle width		Refresh cycle		
DMEMCR	Memory Access Control Reg (Prohibit RMW)	5BH	0: Prohibit	000: 31 states				000: 2 states	0: Not inserted			
			1: Execute	001: 62 states	001: 3 states				010: 4 states	1: inserted		
			010: 78 states	010: 5 states				011: 6 states	0: Disable			
			011: 97 states	011: 7 states				101: 8 states	1: Enable			
			100: 109 states	100: 9 states				110: 11 states	:			
			101: 124 states	101: 11 states				111: 12 states	:			
			110: 154 states	110: 13 states				111: 14 states	:			
			111: 195 states	111: 15 states				111: 16 states	:			
			SRFC	BRM		MACM	MUXE	MUXW1	MUXW0	MAC		
			W	-		R/W						
			1	-	0	0	0	0	0	0		
			Self refresh	DRAM pin Bus	0: Normal access	Address multiplex	Multiplexed address length		Memory access control			
			0: Execute	-	1: Release	1: Slow	0: Disable	00: 8 bit	01: 9 bit	0: Disable		
			1: Release	-	0: Release	access	1: Enable	10: 10 bit	11: 11 bit	1: Enable		
			1: Not release	-	1: Not release	-	-	-	-	-		
			1: Release	-	1: Release	-	-	-	-	-		

## 6. Port Section Equivalent Circuit Diagram

- Reading The Circuit Diagram

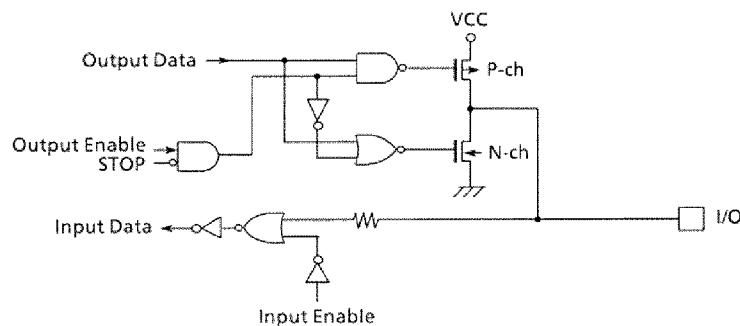
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

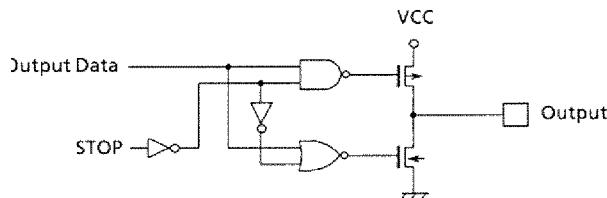
**STOP:** This signal becomes active “1” when the hold mode setting register is set to the STOP mode ( $WDMOD <HALTM1,0> = 0,1$ ) and the CPU executes the HALT instruction. When the drive enable bit [DRVE] is set to “1”, however, STP remains at “0”.

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

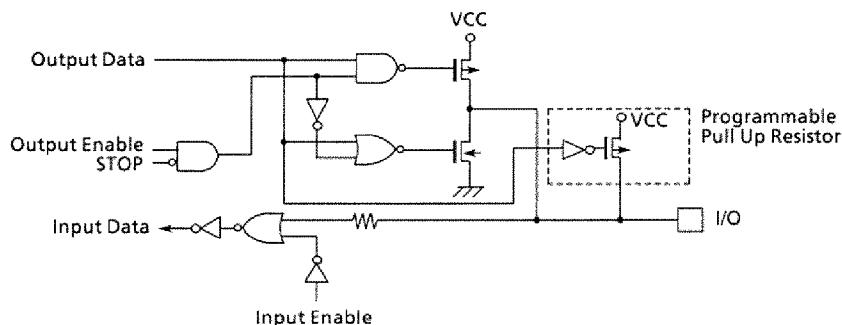
- D0 to D7, P1 (D8 to 15)



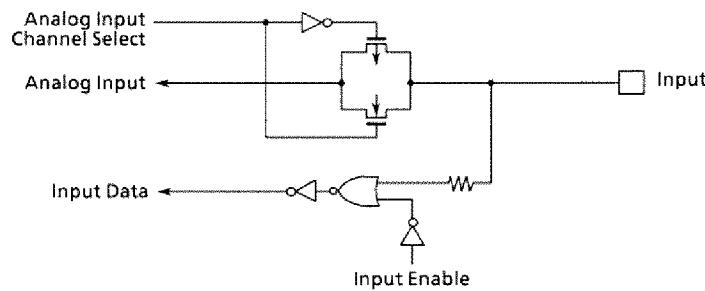
- P2 (A16 to A23), A0 to A15,  $\overline{RD}$ ,  $\overline{WR}$ , P6



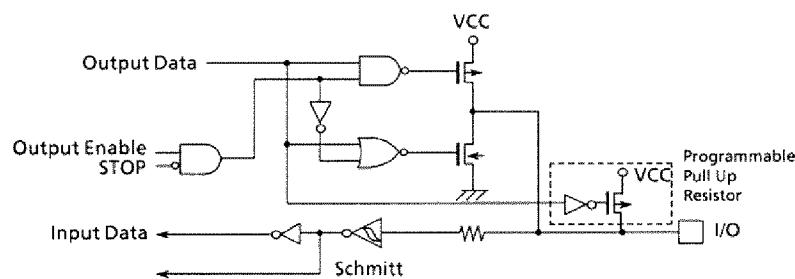
- P52 52, P7, P81, P82, P84, P85, PA, PB6 ~ B0



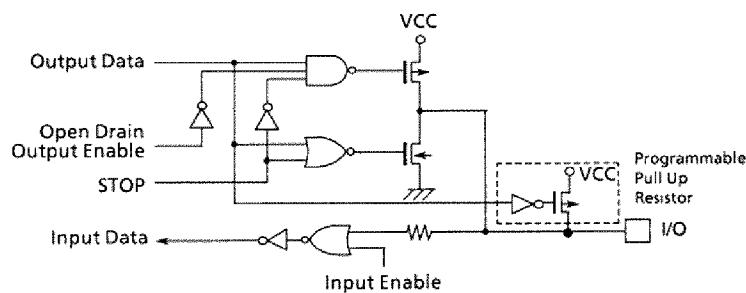
- P9 (AN0 to 3)



- P87 (INT0)



- P80 (TXD0), P83 (TXD1)



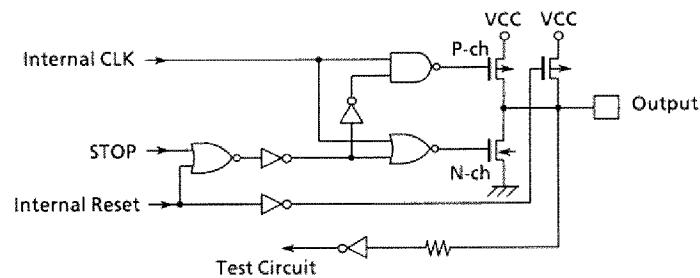
- NMI



- WDTOUT



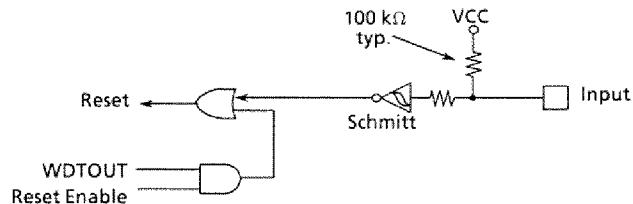
- CLK



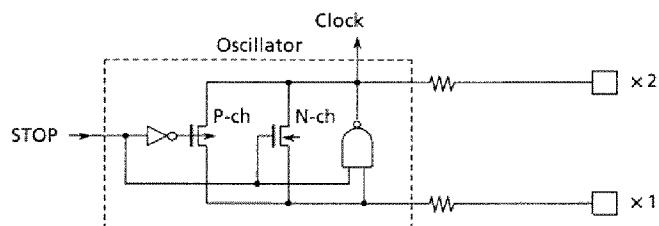
- $\overline{EA}$ , AM8/ $\overline{16}$



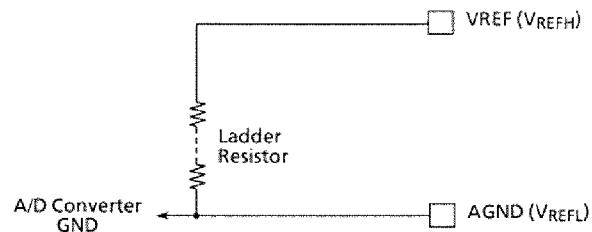
- $\overline{RESET}$



- X1, X2



- VREF (VREFH), AGND (VREFL)



## 7. Care Points and Restriction

### (1) Special Expression

- ① Explanation of a built-in I/O register: Register

Symbol <Bit Symbol>

ex) TRUN <TRUN> ··· Bit T0RUN of Register TRUN

- ② Read, Modify and Write Instruction

An instruction which CPU executes following by one instruction.

1. CPU reads data of the memory.
2. CPU modifies the data.
3. CPU writes the data to the same memory.

ex1) SET 3, (TRUN) ··· set bit3 of TRUN  
ex2) INC1, (100H) ··· increment the data of 100H

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

SET	imm, mem,	RES	imm, mem
CHG	imm, mem,	TSET	imm, mem
INC	imm, mem,	DEC	imm, mem
RLD	A, mem,	ADD	imm, reg

- ③ 1 state

One cycleclock divided by 2 oscillation frequency is called 1 state.

ex) Oscillation frequency is 25MHz.

$2/25\text{MHz} = 80\text{ns} = 1 \text{ state}$

### (2) Care Points

- ①  $\overline{\text{EA}}$ , pin, AM/ $\overline{16}$  pin

Fix these pins VCC or GND unless changing voltage.

- ② Warming-up Counter

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

- ③ Programmable Pull Up/Down Resistance

The programmable pull up/down resistors can be selected ON/OFF by program when they are used as the input ports. The case of they are used as the output ports, they cannot be selected ON/OFF by program.

- ④ 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.

- ⑤ Watch Dog Timer

The watch dog timer starts operation immediately after the reset is released. When the watch dog timer is not used, set watch dog timer to disable.

- ⑥ CPU (HDMA)

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

**Notes**