TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/H Series

TMP95C061B

TOSHIBA CORPORATION

Preface

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

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)

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

CMOS 16-bit Microcontroller
TMP95C061BF

1. Outline and Device Characteristics

TMP95C061BF is high-speed advanced 16-bit microcontroller developed for controlling medium to large-scale equipment. TMP95C061BF is housed in an 100-pin mini flat package (QFP100-P-1414-0.50). TMP95C061BEF is housed in QFP100-P-2222-0.80A package.

Device characteristics are as follows:

(1) Original High speed 16-bit CPU (900/H 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
- Micro DMA : 4 channels (640 ns / 2 bytes at 25 MHz)

(2) Minimum instruction execution time : 160 ns at 25 MHz

(3) Internal RAM : None
Internal ROM : None

(4) External memory expansion
- Can be expanded up to16 Mbytes (for both programs and data).
- AM8 / $\overline{16}$ pin (select the external data bus width)
- Can mix 8- and 16-bit external data buses. ··· Dynamic data bus sizing

(5) DRAM Controller

(6) 8-bit timer : 4 channels

(7) 16-bit timer : 2 channels

(8) Pattern generator : 4 bits, 2 channels

(9) Serial interface : 2 channels
(Only for channel 0, external clock can be used in UART mode.)
(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 ... 7-level priority can be set.
- 6 external interrupts .... 7-level priority can be set.

(14) I/O ports
56 pins
(15) Standby function : 3 HALT modes (RUN, IDLE, STOP)

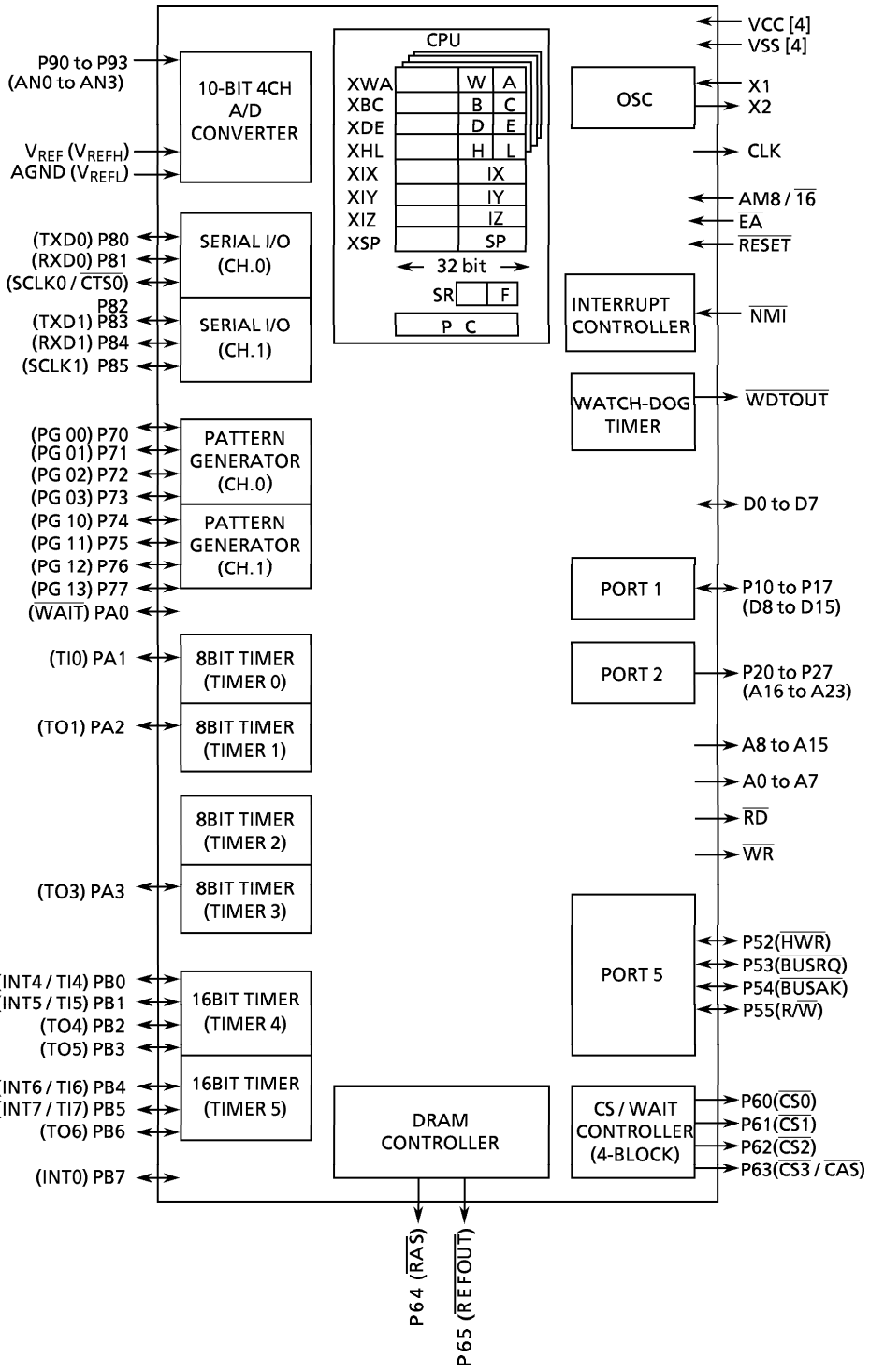


Figure 1 TMP95C061B Block Diagram

2. Pin Assignment and Functions

The assignment of input / output pins for TMP95C061B their name and outline functions are described below.

2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP95C061B.

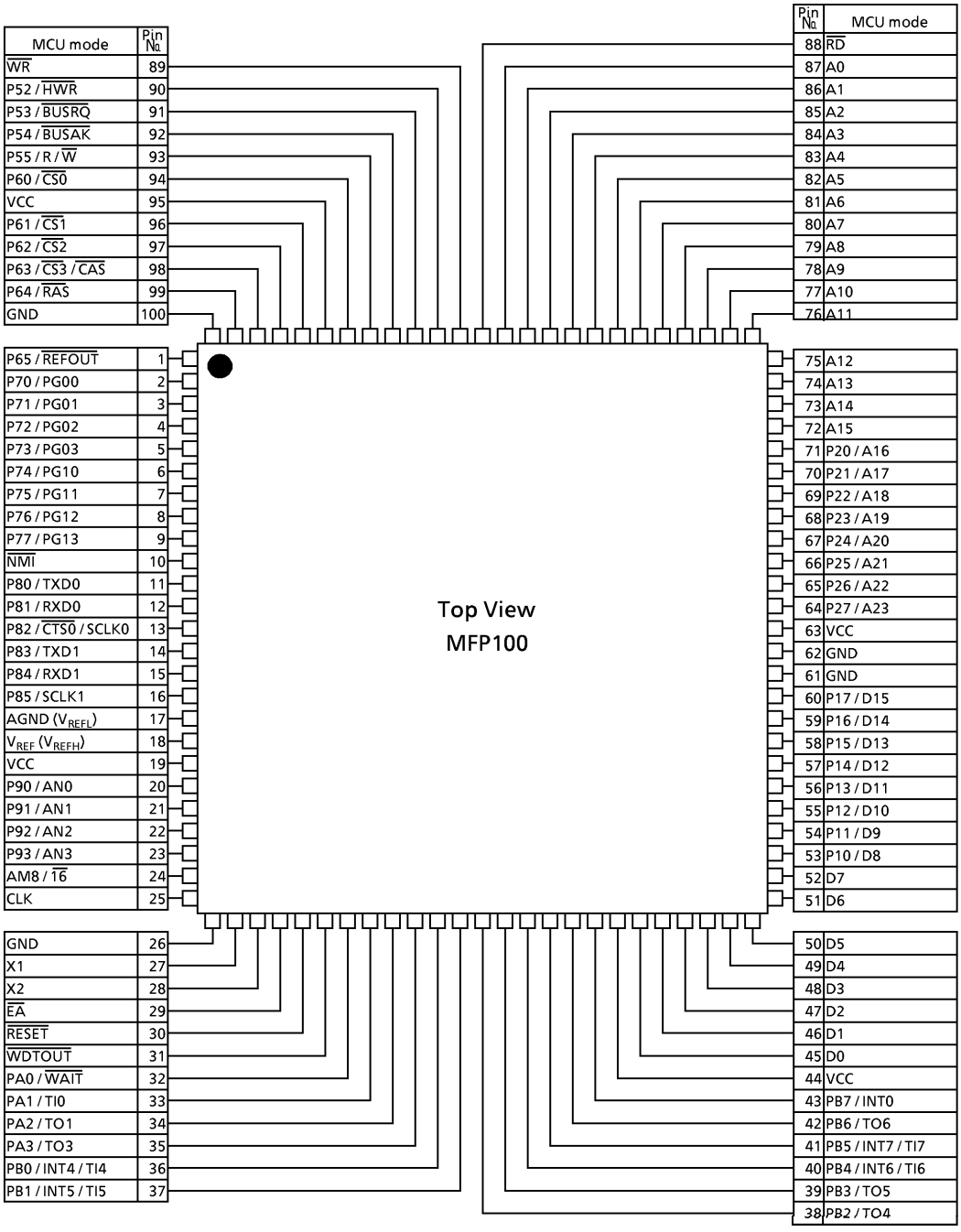


Figure 2.1 Pin Assignment (100-pin MFP)

2.2 Pin Names and Functions

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

Table 2.2 Pin Names and Functions.

Table 2.2

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 basls 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
$\overline{RD}$	1	Output	Read : Strobe signal for reading external memory
$\overline{WR}$	1	Output	Write : Strobe signal for writing data on pins D0 to 7
P52 $\overline{HWR}$	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 $\overline{BUSRQ}$	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, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RAS}$, $\overline{CAS}$ and $\overline{REFOUT}$ (*) pins. (for external DMAC)
P54 $\overline{BUSAK}$	1	I/O Output	Port 54 : I/O port (with pull-up resistor) Bus Acknowledge : Signal indicating that D0 to 15, A0 to 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RAS}$, $\overline{CAS}$ and $\overline{REFOUT}$ (*) pins are at high impedance after receiving $\overline{BUSRQ}$. (for external DMAC)
P55 R/$\overline{W}$	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 $\overline{CS0}$	1	Output Output	Port 60 : Output port Chip Select 0 : Outputs 0 when address is within specified address area
P61 $\overline{CS1}$	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 $\overline{BUSRQ}$ and $\overline{BUSAK}$ pins.

(*) $\overline{RAS}$, $\overline{CAS}$, and $\overline{REFOUT}$ 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 $\overline{CS2}$	1	Output Output	Port 62 : Output port Chip Select 2 : Outputs 0 if address is within specified address area
P63 $\overline{CS3}$ $\overline{CAS}$	1	Output Output Output	Port 63 : Output port Chip Select 3 : Outputs 0 if address is within specified address area Column address strobe : Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area
P64 $\overline{RAS}$	1	Output Output	Port 64 : Output port Low address strobe : Output $\overline{RAS}$ strobe for DRAM if address is within specified address area
P65 $\overline{REFOUT}$	1	Output Output	Port 65 : Output port Refresh Output : 0 : indicates priod of refresh cycle
P70 to P73 PG00 to PG03	4	I/O Output	Port 70 to 73 : I/O port that allow selection of I/O on a bit basis (with pull-up resister) Pattern generator Port : 00 to 03
P74 to P77 PG10 to PG13	4	I/O Output	Port 74 to 77 : I/O port that allow selection of I/O on a bit basis (with pull-up resister) Pattern generator Port : 10 to 13
P80 TXD0	1	I/O Output	Port 80 : I/O port (with pull-up resister) Serial send data 0
P81 RXD0	1	I/O Input	Port 81 : I/O port (with pull-up resister) Serial receive data 0
P82 $\overline{CTS0}$ SCLK0	1	I/O Input I/O	Port 82 : I/O port (with pull-up resister) Serial data send enable (clear to send) Serial Clock I/O 0
P83 TXD1	1	I/O Output	Port 83 : I/O port (with pull-up resister) Serial send data 1
P84 RXD1	1	I/O Input	Port 84 : I/O port (with pull-up resister) Serial receive data 1
P85 SCLK1	1	I/O I/O	Port 85 : I/O port (with pull-up resister) Serial clock I/O 1
P90 to P93 AN0 to AN3	4	Input Input	Port 9 : Input port Analog input : Input to A/D converter
PA0 $\overline{WAIT}$	1	I/O Input	Port A0 : I/O port (with pull-up resister) Wait : Pin used to request CPU us wait
PA1 TI0	1	I/O Input	Port A1 : I/O port (with pull-up resister) Timer input 0 : Timer 0 input
PA2 TO1	1	I/O Output	Port A2 : I/O port (with pull-up resister) 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 : Timer 2 or 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 86 : 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 progmnable rising / fulling 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 progmmable level / rising edge
V_{REF} (V_{REFH})	1	Input	Pin for refereme voltage input to A/D connecter
AGND (V_{REFL})	7	Input	Ground pin for A/D converter
$\overline{WDTOUT}$	1	Output	Watchdog timer output pin
$\overline{NMI}$	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「external input clock X1 ÷ 4」clock. Pulled-up during reset.
$\overline{EA}$	1	Input	fixed GND
AM8/$\overline{16}$	1	Input	Address mode : Selects external Data Bus width "0" should be inputted with fixed 16 bit Bus width or 16 bit Bus interlorded with 8 bit Bus. "1" should be inputted with fixed 8 bit Bus width
$\overline{RESET}$	1	Input	Reset : Initializes LSI (with pull-up resister)
X1 / X2	2	I/O	Oscillator connecting pin
VCC	4		Power supply pin (+ 5 V) (All Vcc pins should be connected with the power supply pin.)
VSS	4		GND pin (0 V) (All Vss pins should be connected with GND (0 V).)

Note 1 : Pull-up resistor can be released from the pin by software.
Note 2 : Connect all VCC pins to power supply and all VSS pins to GND.

3. Operation

This section describes in blocks the functions and basic operations of TMP95C061B devices.

Check the 「7. Care Points and Restriction 」because of the Care Points etc are described.

3.1 CPU

TMP95C061B devices 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 TMP95C061B that are not described in the previous section.

3.1.1 Reset

To reset the TMP95C061B, the $\overline{\text{RESET}}$ input must be kept at 0 for at least 10 system clocks (10 states: 0.8 μs at 25 MHz) within an operating voltage range and with a stable oscillation.

When reset is accepted, the CPU sets as follows:

- Program Counter (PC) according to Reset Vector that is stored 0FFFF00H to 0FFFF02H.
  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.)
- Sets the MAX bit of the status register (SR) to 1 (this sets maximum mode).
  (Note : This product does not support minimum mode. Do not use the MIN instruction.)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

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

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

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

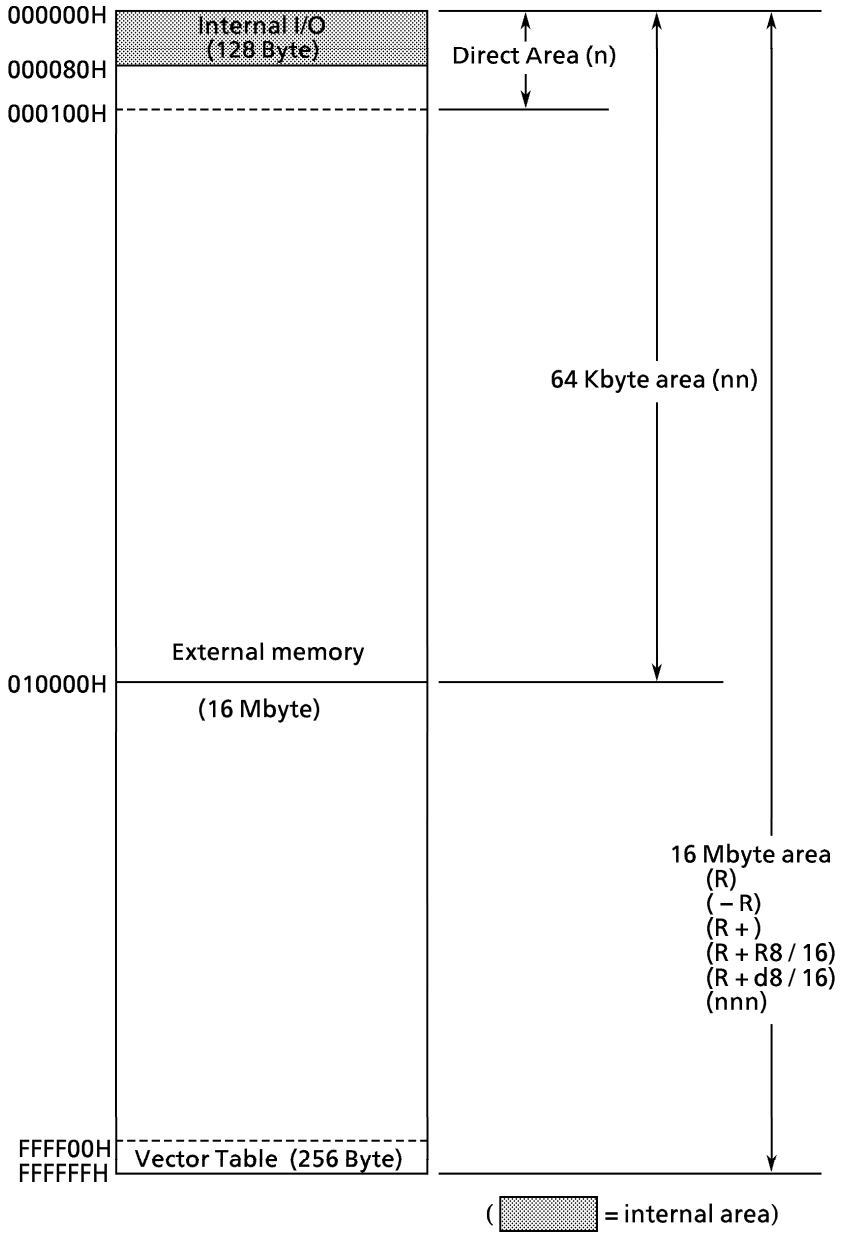
3.1.2 External data width selection pin (AM8 / $\overline{16}$)

After Reset operation, TMP95C061B operates 8 bits or 16 bits external data width according to input to AM8 / $\overline{16}$ pin.

- In case with fixed 16 bit bus or 16 bit bus interlarded with 8 bit bus
  "0" should be inputted. In this case, 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 resister.
- In case with fixed 8 bit bus
  "1" should be inputted. In this case, Port 1 (P10 to P17) operate as 8 bit I/O ports. And the value set in Chip Select / Wait Control resister <B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> and <BEXBUS> are neglected.

3.2 Memory Map

Figure 3.2 shows a memory map of the TMP95C061B.



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

Figure 3.2 Memory Map

3.2.1 Operation at internal I/O area access

TMP95C061B uses 128 bytes of address space (0H to 7FH) as an internal I/O area. Internal I/O registers are mapped on this area.

Operation of the internal I/O area access is different from that of the other address area access about following two points.

(1) In the internal I/O area access, $\overline{RD}$, and $\overline{WR}$ ($\overline{HWR}$) strobe signals are nonactive and fixed to high level.

However, in PSRAM mode set by P5<RDE>register, $\overline{RD}$ strobe signal becomes active also in the internal I/O area access. (See 3.5.3 Port5 (P52 to P55).)

(2) In the internal I/O area access, the number of waits becomes zero or one depending on the internal state of the CPU. This wait can't be controlled by chip select / wait controller (see 3.6 Chip Select / Wait Controller, AM8/$\overline{16}$ pin). When the specified address area overlaps with the internal I/O area, the operation as the internal I/O area takes priority of the specified address area.

3.3 Interrupts

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

TMP95C061B has the following 26 interrupt sources:

- Interrupts from the CPU··· 2
  (Software interrupts, and Illegal (undefined) instruction execution)
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0, and INT4 to 7)··· 6
- Interrupts from built-in I/Os··· 14
- Interrupts from micro DMA··· 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 interupt source to the CPU. When more than one interupt is generated simultaneously, the interrupt controller sends the value of the highest priority (7 for non-maskable interrupts is the highest) to the CPU.

The CPU compares the value of the priority sent with the value in the CPU interrupt mask register (IFF2 to 0). If the value is higher or equal to that of the CPU interrupt mask register, the interrupt is accepted. However, software interrupts and illegal instruction interrupts generated by the CPU are processed without comparison with the IFF<2:0>value.

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 micro DMA processing mode . Micro DMA 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.

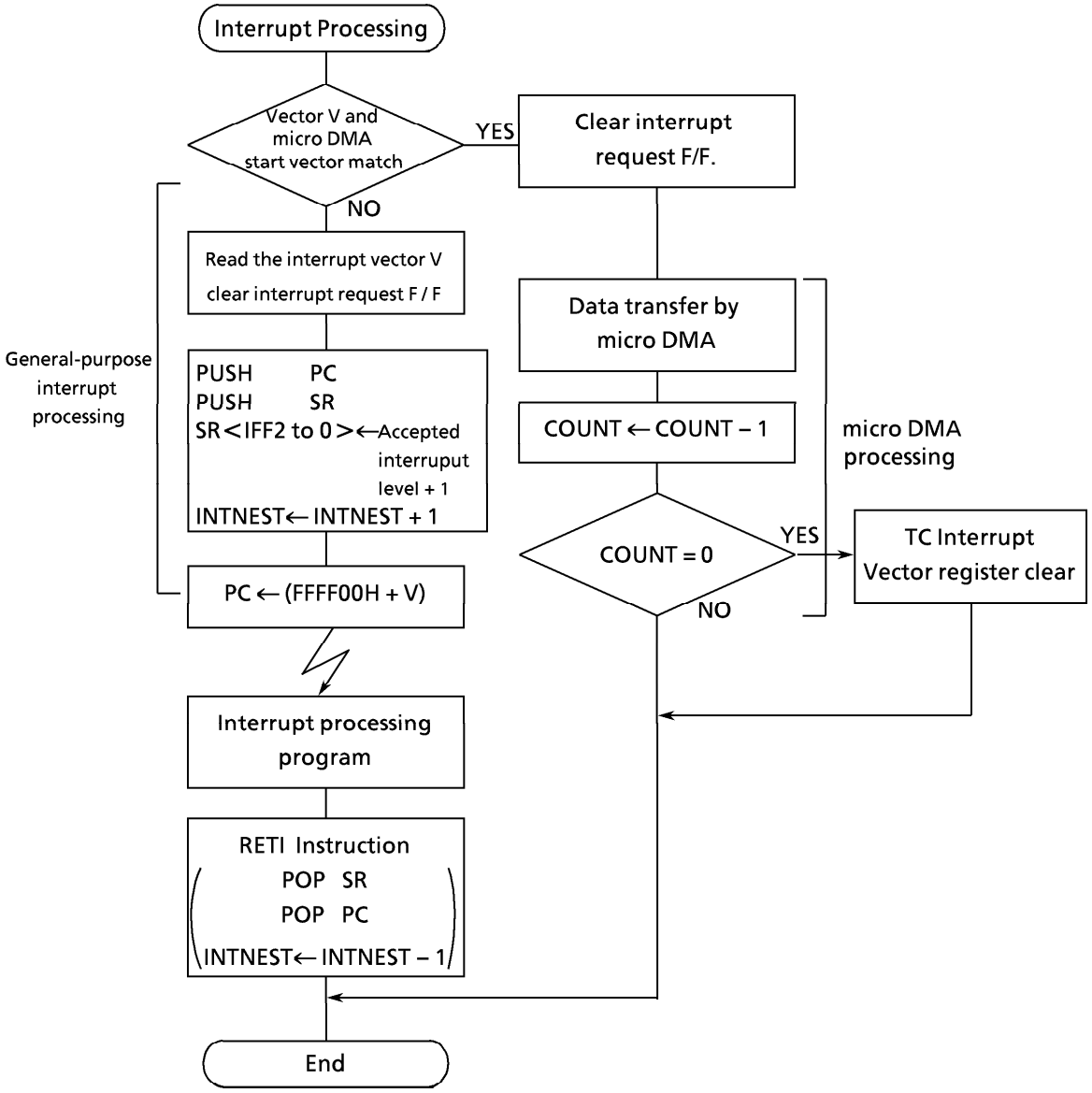


Figure 3.3 (1) Interrupt Processing Flowchart

3.3.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows:

However, in the case of software interrupts and illegal instruction interrupts generated by the CPU, the CPU skips (1) and (3) and executes steps (2), (4), and (5).

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

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

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

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

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

The following diagram shows all the above processing state number.

Bus Width of Stack Area	Bus Width of Interrupt Vector Area	Interrupt processing state number
8 bit	8 bit	28
	16 bit	24
16 bit	8 bit	22
	16 bit	18

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

Though acceptance of non-maskable interrupts cannot be disabled by program, acceptance of maskable interrupts can. A priority can be set for each source of maskable interrupts. The CPU accepts an interrupt request with a priority higher or equal to 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 routine 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 0FFFF00H to 0FFFFFFH (256 bytes) of the TMP95C061B are assigned for interrupt vector area.

Table3.3 (1) TMP95C061B Interrupt Table

Default priority	Type	Interrupt source	Vector value "V"	Address refer to vector	Micro DMA start vector
1		Reset, or SWI0 instruction	0000H	FFFF00H	–
2		SWI 1 instruction	0004H	FFFF04H	–
3		INTUNDEF : Illegal instruction, or SWI2	0008H	FFFF08H	–
4	Non-	SWI 3 instruction	000CH	FFFF0CH	–
5	maskable	SWI 4 instruction	0010H	FFFF10H	–
6		SWI 5 instruction	0014H	FFFF14H	–
7		SWI 6 instruction	0018H	FFFF18H	–
8		SWI 7 instruction	001CH	FFFF1CH	–
9		$\overline{\text{NMI}}$ Pin	0020H	FFFF20H	–
10		INTWD : Watchdog timer	0024H	FFFF24H	–
–		(Micro DMA)	–	–	–
11		INT0 pin	0028H	FFFF28H	0AH
12		INT4 pin	002CH	FFFF2CH	0BH
13		INT5 pin	0030H	FFFF30H	0CH
14		INT6 pin	0034H	FFFF34H	0DH
15		INT7 pin	0038H	FFFF38H	0EH
–		(Reserved)	003CH	FFFF3CH	–
16		INTT0 : 8-bit timer0	0040H	FFFF40H	10H
17		INTT1 : 8-bit timer1	0044H	FFFF44H	11H
18		INTT2 : 8-bit timer2	0048H	FFFF48H	12H
19		INTT3 : 8-bit timer3	004CH	FFFF4CH	13H
20	Maskable	INTTR4 : 16-bit timer4 (TREG4)	0050H	FFFF50H	14H
21		INTTR5 : 16-bit timer4 (TREG5)	0054H	FFFF54H	15H
22		INTTR6 : 16-bit timer5 (TREG6)	0058H	FFFF58H	16H
23		INTTR7 : 16-bit timer5 (TREG7)	005CH	FFFF5CH	17H
24		INTRX0 : Serial receive (Channel.0)	0060H	FFFF60H	18H
25		INTTX0 : Serial send (Channel.0)	0064H	FFFF64H	19H
26		INTRX1 : Serial receive (Channel.1)	0068H	FFFF68H	1AH
27		INTTX1 : Serial send (Channel.1)	006CH	FFFF6CH	1BH
28		INTAD : A/D conversion completion	0070H	FFFF70H	1CH
29		INTTC0 Micro DMA completion (channel.0)	0074H	FFFF74H	–
30		INTTC1 Micro DMA completion (channel.1)	0078H	FFFF78H	–
31		INTTC2 Micro DMA completion (channel.2)	007CH	FFFF7CH	–
32		INTTC3 Micro DMA completion (channel.3)	0080H	FFFF80H	–
–		(Reserved)	0084H	FFFF84H	–
to		to	to	to	to
–		(Reserved)	00FCH	FFFFFCH	–

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, $\overline{\text{NMI}}$ Vector : 9ABCH, INTAD Vector : 123456h.

```
ORG     8100H
LD      A, B
        :
ORG     9ABCH
LD      B, C
        :
ORG     123456H
LD      C, A
        :
ORG     0FFFF00H
DL      008100H         ; Reset = 8100H

ORG     0FFFF20H
DL      009ABCH         ; NMI = 9ABCH

ORG     0FFFF70H
DL      123456H         ; INTAD = 123456H
```

(cf)

ORG, DL are the Assembler Directive.

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

3.3.2 Micro DMA

In addition to conventional interrupt processing, TMP95C061B supports the micro DMA function. For interrupt requests set for micro DMA, micro DMA processing is performed at the highest priority for maskable interrupts (level 6), regardless of the actual interrupt level set for the interrupt.

Because the function of micro DMA has been implemented with the cooperative operation of CPU, when CPU is a state of stand-by by HALT instruction, the requirement of micro DMA will be ignored (pending).

(1) Micro DMA Operation

When an interrupt request occurs for an interrupt specified by the micro DMA start vector register, micro DMA sends the micro DMA request to the CPU with the highest priority for maskable interrupts (level 6), regardless of the actual interrupt level set for the interrupt, and starts micro DMA. The micro DMA function has four channels. This allows micro DMA to be set for up to four interrupts at the same time.

When micro DMA is accepted, the interrupt request F-F for the micro DMA channel is cleared, data are automatically transferred from the transfer source address to the transfer destination address (the addresses are set in the control register), and the transfer count is decremented. If the decremented result is other than zero, micro DMA processing terminates. If the decremented result is zero, the CPU sends a micro DMA transfer end interrupt (INTTCn) to the interrupt controller, clears the micro DMA start vector register to 0, disables the next micro DMA startup, and terminates micro DMA processing.

If an interrupt request for the interrupt source used is received between the time that the micro DMA start vector is cleared and the time that it is reset, the CPU performs general-purpose processing at the specified interrupt level. Therefore, if the interrupt source is only being used for starting micro DMA (not used as an interrupt), set the interrupt level to zero.

When simultaneously using the same interrupt resource for both the micro DMA and general-purpose interrupts as described above, set the level of the interrupt source used to start micro DMA lower than the levels of all other interrupt sources. In this case, the cause of general interrupt is limited to the edge interrupt.

Example : When using timers 0 to 3 for running micro DMA 0 to 3
Set the interrupt level of timers 0 to 3 to 1
Set other interrupt levels to 2 to 6

Like other maskable interrupts, the priority of the micro DMA transfer end interrupt is determined by the interrupt level and default priority.
If multiple-channel micro DMA requests occur at the same time, the priority is determined by the channel numbers, not the interrupt levels. The lower the channel number, the higher the priority. (CH0 (high) → CH3 (low))

The transfer source and transfer destination addresses are set in 32-bit control registers. However, as only 24-bit addresses are output, the address space available to micro DMA is 16M bytes.

Three transfer modes are supported: 1-byte transfer, 1-word transfer (= two bytes), and 4-byte transfer. For each transfer mode, it is possible to specify whether to increment, decrement, or fix source and destination addresses after transfer. These modes facilitate data transfer from I/O to memory, from memory to I/O, and from I/O to I/O. For transfer mode details, see "Transfer Mode Register Details" later in this manual.

As a 16-bit transfer counter is used, micro DMA can perform a maximum of 65536 transfers (initializing the counter to 0000H specifies the maximum number of transfers).

The 18 interrupt sources with micro DMA start vectors (as listed in Table 3.3 (1)) can be used to start micro DMA processing.

Figure 3.3.2 (1) shows the micro DMA cycle for 1-word transfer in transfer destination address INC mode (the same apart from counter mode). (The conditions for this cycle are based on a 16-bit bus, 0 waits, and transfer source/transfer destination addresses both even-numbered values.).

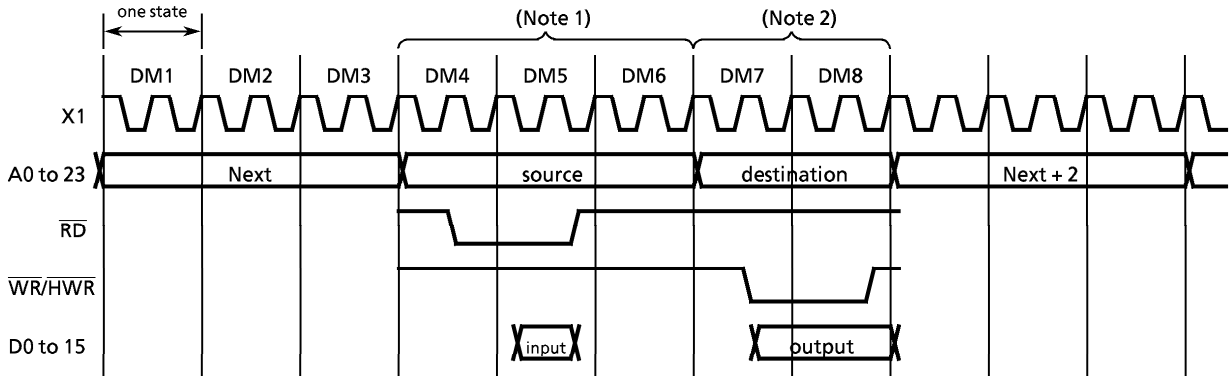


Figure 3.3.2 (1) Micro DMA Cycle Diagram

States 1 to 3 : Instruction fetch cycle (prefetches the next instruction code)
If the instruction cue buffer has three or more bytes of instruction code, the cycles are dummy cycles.

States 4 to 5 : Micro DMA read cycles

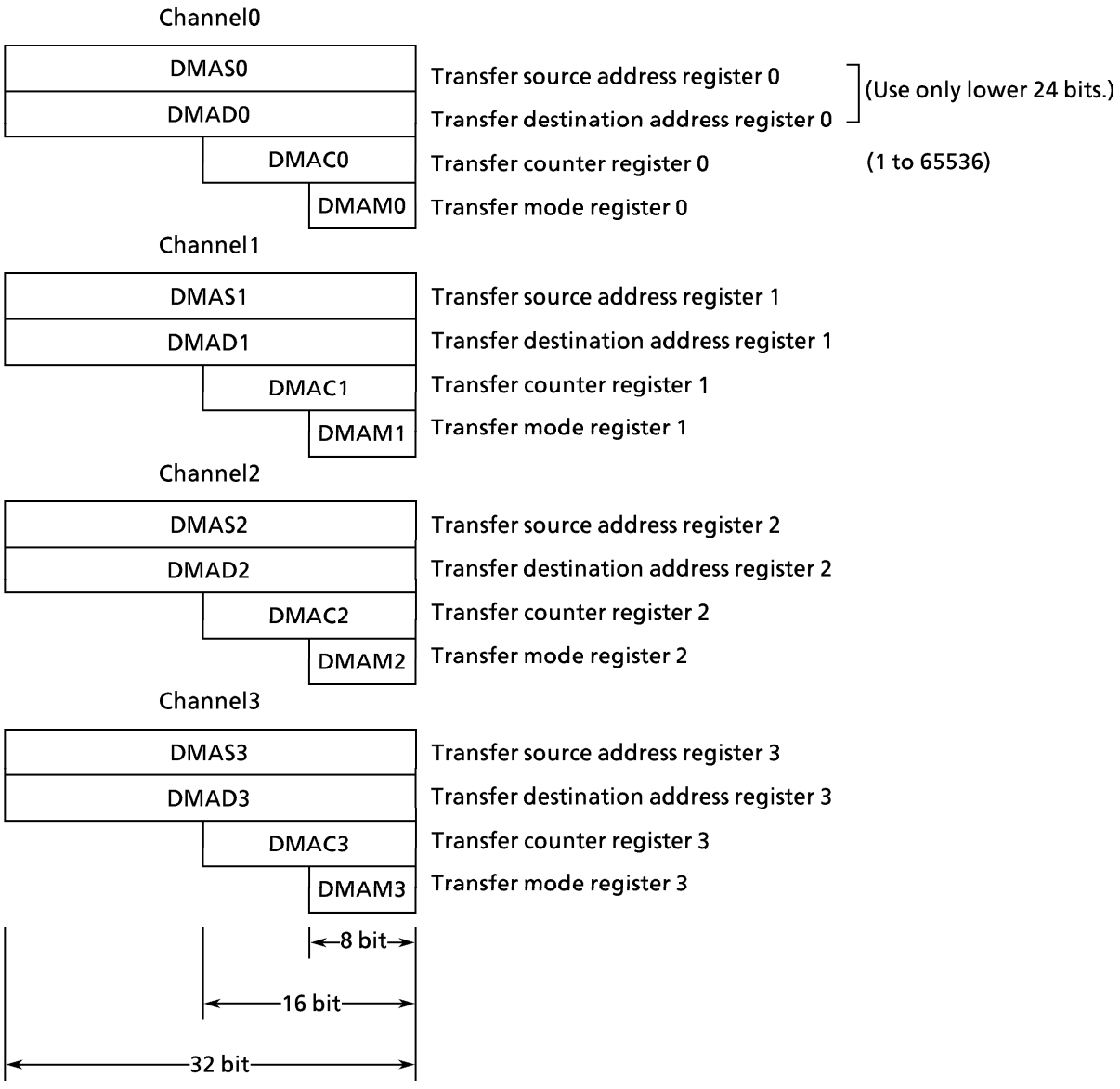
State 6 : Dummy cycle (address bus remains the same as in state 5)

States 6 to 8 : Micro DMA write cycle

Note 1 : If the source address area uses an 8-bit bus, two states are added.
If also the source address area uses a 16-bit bus and the source address is an odd-numbered address, two states are added.

Note 2 : If the destination address area uses an 8-bit bus, two states are added.
If also the destination address area uses a 16-bit bus and the destination address is an odd-numbered address, two states are added.

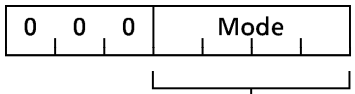
(2) Register configuration (CPU control register)



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

(3) Transfer mode register details

(DMAM0 to 3)



Note : When setting values for this register, set the upper 3 bits to 0.

ZZ: 0 = byte transfer, 1 = word transfer, 2 = 4-byte transfer, 3 = reservation

		execution time (Min.at 25 MHz)
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 micro DMA channels 0 to 3.
DMADn + / DMASn + : Post-increment (Increments register value after transfer.)
DMADn – / DMASn – : Post-decrement (Decrement 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 micro DMA start vector. The interrupt request fip-flop is used to latch interrupt requests from peripheral devices. The flip-flop is cleared to 0 at reset, when the CPU reads the interrupt channel vector after the acceptance of interrupt, when the CPU accepts the micro DMA request 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 (eg, INTE0AD, INTE45, etc.) provided for each interrupt source. Interrupt levels to be set are from 1 to 6. Writing 0 or 7 as the interrupt priority disables the corresponding interrupt request. The priority of the non-maskable interrupt ($\overline{\text{NMI}}$ pin, watchdog timer, etc.) is fixed to 7. If interrupt requests with the same interrupt level are generated simultaneously, interrupts are accepted in accordance with the default priority (the smaller the vector value, the higher the priority).

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the interrupt mask register <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 valve in the interrupt mask register saved in the stack before the interrupt was generated to the CPU SR<IFF2 to 0>.

The interrupt controller also has four registers used to store the micro DMA start vector. These are I/O registers; unlike other micro DMA registers (DMAS, DMAD, DMAM, and DMAC). Writing the start vector of the interrupt source for the micro DMA processing (see Table 3.3.(1)), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (eg, DMAS and DMAD) prior to the micro DMA processing.

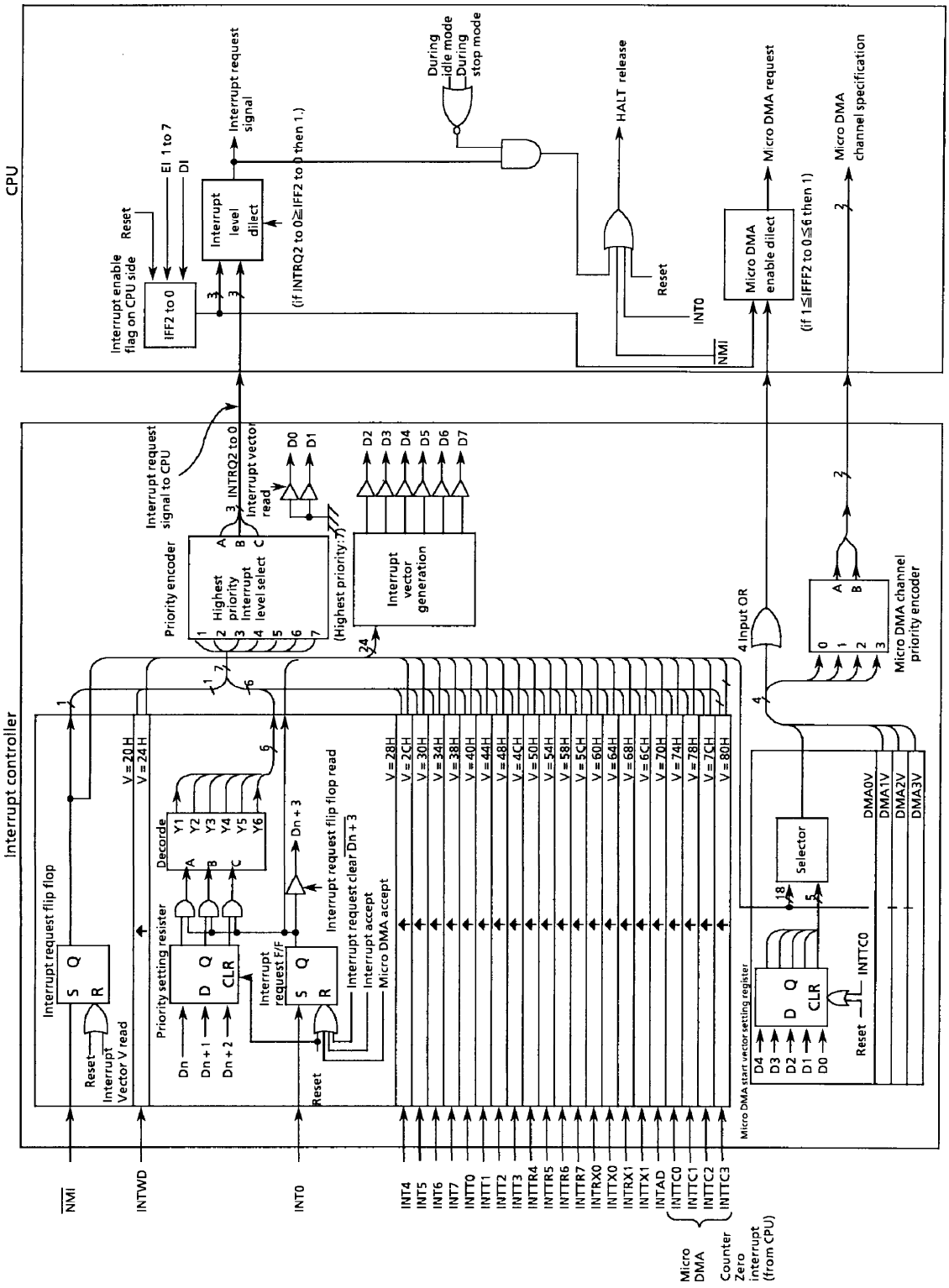


Figure 3.3.3 (1) Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

(Read-modify-write is prohibited.)

Symbol	Address	7	6	5	4	3	2	1	0
INTE0AD	0070H	INTAD				INT0			
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE45	0071H	INT5				INT4			
		I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE67	0072H	INT7				INT6			
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET01	0073H	INTT1 (Timer 1)				INTT0 (Timer 0)			
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET23	0074H	INTT3 (Timer 3)				INTT2 (Timer 2)			
		IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET45	0075H	INTTR5 (TREG5)				INTTR4 (TREG4)			
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET67	0076H	INTTR7 (TREG7)				INTTR6 (TREG6)			
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTES0	0077H	INTTX0				INTRX0			
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTES1	0078H	INTTX1				INTRX1			
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTETC01	0079H	INTTC1				INTTC0			
		ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTETC23	007AH	INTTC3				INTTC2			
		ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0

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

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

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

(2) External interrupt control

Interrupt Input Mode Control Register

IIMC (007BH)

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

Read-modify-write is prohibited.

INT0 input enable (Note)

0	INT0 disable (PB7 function only)
1	Input enable

$\overline{NMI}$ rising edge enable

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

INT0 level enable

0	Rising edge detect interrupt
1	High level interrupt

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

Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
$\overline{NMI}$	—	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	PB7	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 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) Micro DMA start vector

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

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

If the same vector is set in the micro DMA 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 micro DMA start vector registers of two channels, the interrupt generated in the channel with the smaller number is processed until the micro DMA transfer end. If the micro DMA start vector of this channel is not set again, the next micro DMA is started for the channel with the higher number. (micro DMA chaining)

Micro DMA0 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

Micro DMA1 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

Micro DMA2 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

Micro DMA3 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. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag between accepting and reading the interrupt vector. In this case, the CPU reads the default vector 0028H and reads the interrupt vector at address FFFF28H.

To avoid the above problem, place instructions that clear interrupt request flags after a DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.

In the case of changing the value of the interrupt mask register<IFF2 to 0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

In addition, take care as the following three circuits are exceptional and demand special attention.

INT0 level mode	INT0 in level mode is not an edge-detect interrupt, so the interrupt request flip-flop function is canceled. The peripheral interrupt request bypasses the S input of the flip-flop, and acts as the Q output. Changing modes from edge to level automatically clears the interrupt request flag.
	If the CPU enters the interrupt response sequence as a result of setting INT0 from 0 to 1, INT0 must be held at 1 until the interrupt response sequence is completed. If the INT0 level mode is used to release a halt, INT0 must be held at 1 from the time INT0 changes from 0 to 1, to the time when the halt is released. (Ensure that INT0 does not go back 0 due to noise before the halt is released.)
	When switching modes from level to edge, any interrupt request flag set in level mode is not cleared. Accordingly, clear the interrupt request flag using the following sequence.
	
	DI
	LD (IIMC), 00H ; Switches from level to edge.
	LD (INTE0AD), 00H ; Clears interrupt request flag.
	EI

INTAD	The interrupt request flip-flop can only be cleared by reset or by reading the A/D conversion result register, not by an instruction.
INTRX	The interrupt request flip-flop can only be cleared by reset or by reading the serial channel receive buffer, not by an instruction.

Note : The following instructions or pin changes are equivalent to instructions that clear the interrupt request flag.

INT0 : Instructions that switch to level mode after an interrupt request is generated in edge mode.
The pin input changes from high to low after an interrupt request is generated in level mode. ("H" → "L")

INTAD : Instructions that read the A/D conversion result register.

INTRX : Instructions that read the receive buffer.

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.

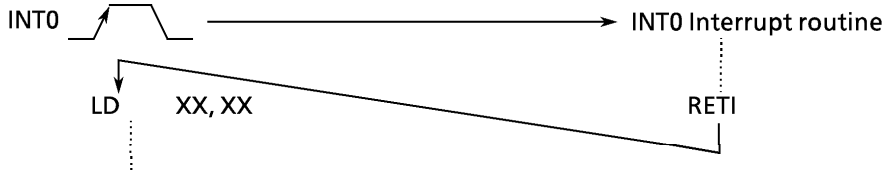
(2) IDLE : Only the built-in oscillator operates, while all other built-in circuits stop. The Power Consumption is reduced to 1/10 or less than that during NORMAL operation.

(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 micro DMA start except for INT0.)

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

```
Address
8203H     LD     (IIMC), 04H          ; sets input to enable for INT0
8206H     LD     (INTE0AD), 06H       ; sets interrupt level to '6' for INT0
8209H     EI     5                    ; sets interrupt level to '5' for CPU
820BH     LD     (WDMOD), 00H         ; sets HALT mode to 'RUN'
820EH     HALT                        ; halts CPU
```




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

(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 executed. Only the CPU stops executing the instruction. Until the HALT state is released, the CPU repeats dummy cycles. In the HALT state, an interrupt request is sampled with the falling edge of the "CLK" signal.

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

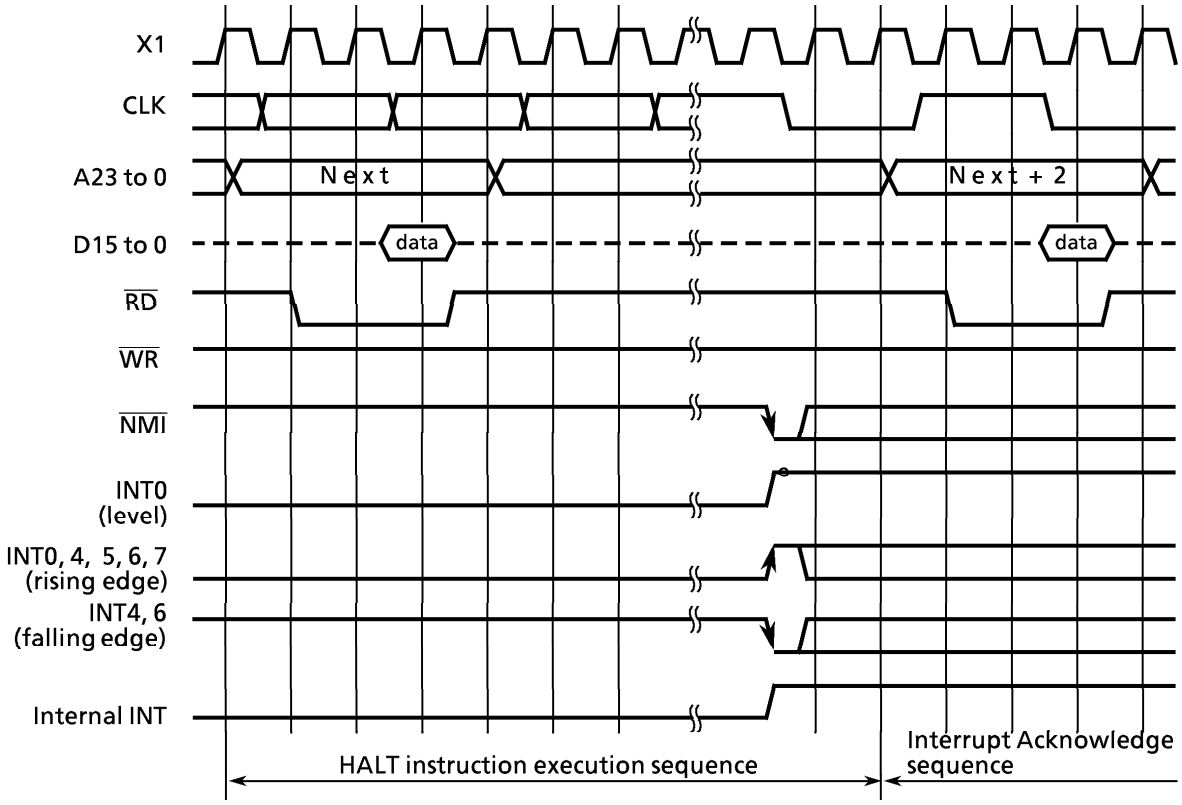


Figure 3.4.1 Timing Chart for Releasing the HALT State by Interrupt in RUN Modes

(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 the internal oscillator operates. The system clock in the MCU stops, and the CLK pin is fixed at the "1" level.

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

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

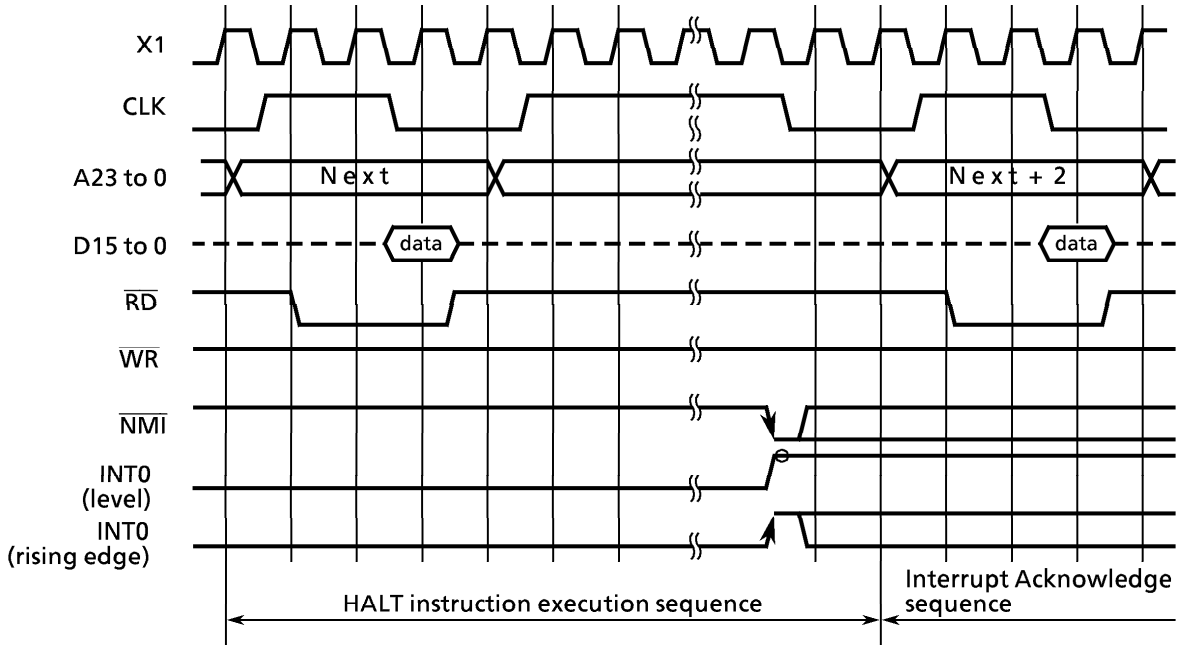


Figure 3.4.2 Timing Chart of HALT Released by Interrupts in IDLE Mode

(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.

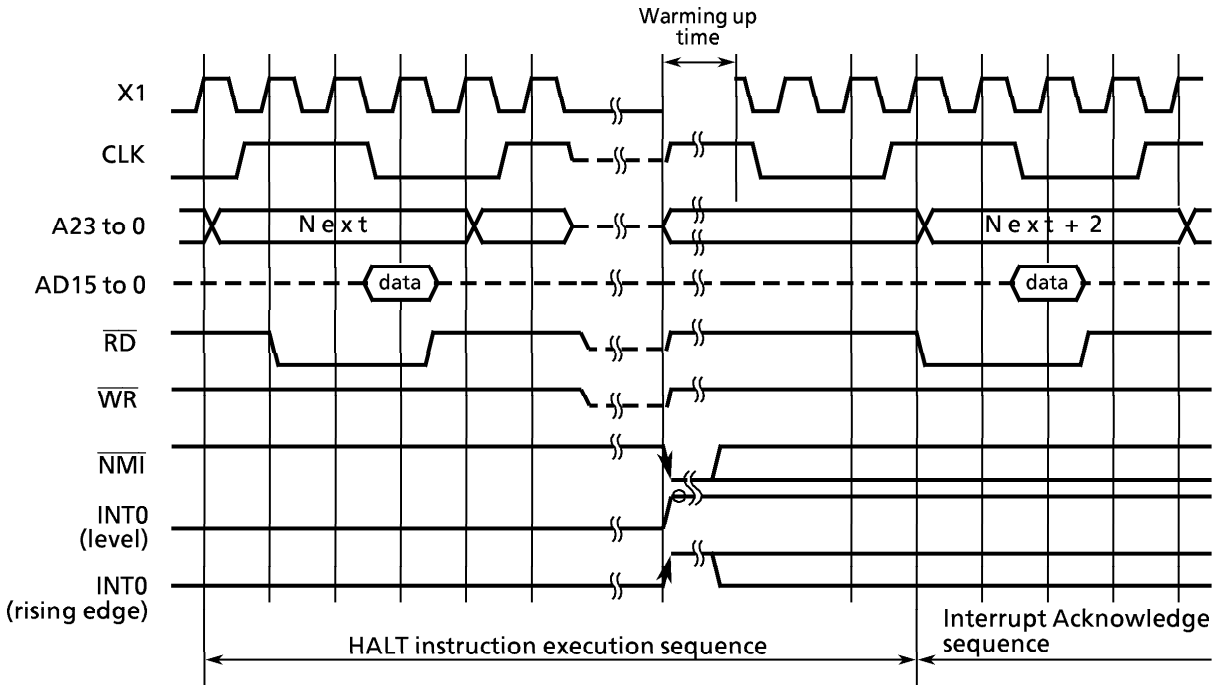


Figure 3.4.3 Timing Chart of HALT Released by Interrupt in STOP Mode

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

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

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

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.

Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

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
$\overline{RD}$, $\overline{WR}$	Output	HI-Z	"1"
P52 to P55 ($\overline{HWR}$, $\overline{BUSRQ}$, $\overline{BUSAK}$, R/$\overline{W}$)	Input mode Output mode	PU* PU*	PU△ Output
P60 to P65 ($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{REFOUT}$)	Output	HI-Z	Output
P70 to P77 (PG00 to PG13)	Input mode Output mode	PU* PU*	PU△ Output
P80 to P85 (TXD, RXD, SCLK, $\overline{CTS}$)	Input mode Output mode	PU* PU*	PU△ Output
P90 to P93 (AN0 to AN3)	Input (PORT) Input (AN0 to AN3)	invalid ◎	invalid ◎
PA0 ($\overline{WAIT}$)	Input mode Output mode	PU* PU*	PU△ Output
PA1 to PA3 (TI0, TO1, TO3)	Input mode Output mode	PU* PU*	PU△ Output
PB0 to PB6 (TI4 to 7, TO4 to 6, INT4 to 7)	Input mode Output mode	PU* PU*	PU△ Output
PB7 (INT0)	Input mode Output mode	PU△ PU△	PU△ Output
$\overline{NMI}$	Input	valid	valid
$\overline{WDTOUT}$	Output	Output	Output
CLK	Output	HI-Z	"1"
$\overline{RESET}$	Input	valid	valid
AM (8 / $\overline{16}$)	Input	◎	◎
$\overline{EA}$	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.
An instruction to access the port register (Ex. P8) should not be placed before the HALT instruction. There is possibility that the input gate is not disabled.
△ : 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	Operating		See Table 3.4 (1)
	8 bit Timer		Stopped	
	8 bit PWM Timer			
	16 bit Timer			
	Pattern Generator			
	Serial Interface			
	A/D Converter			
	Watch Dog Timer			
	DRAM Controller			
	Interrupt Controller			

Interrupt mask, request level			Interrupt request level ≧ Interrupt mask 〈 IFF2 to 0 〉			Interrupt request level*2 < Interrupt mask 〈 IFF2 to 0 〉		
Halt mode			RUN	IDLE	STOP	RUN	IDLE	STOP
Halt release sources	Inter-rupt	NMI	◎	◎	◎*1	◎	◎	◎*1
		INTWD	◎	×	×	◎	×	×
		INT0	◎	◎	◎*1	○	○	○*1
		INT4 to 7	◎	×	×	×	×	×
		INTT0 to 3	◎	×	×	×	×	×
		INTTR4 to 7	◎	×	×	×	×	×
		INTRXD0, 1	◎	×	×	×	×	×
		INTTXD0, 1	◎	×	×	×	×	×
		INTAD	◎	×	×	×	×	×
	RESET		◎	◎	◎	◎	◎	◎

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

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

× : 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 TMP95C061B has a total of 56 bits when the AM8/16 pin is set to high level; a total of 48 bits when the AM8 / 16 pin is set to low level.

These ports are also used for internal CPU and I/O. Table 3.5 (1) lists port pin functions. Table 3.5 (2) lists I/O port setting.

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

Table 3.5 (1) Functions of Ports

Port	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	P52	1	I/O	↑	Bit	$\overline{HWR}$
	P53	1	I/O	↑	Bit	$\overline{BUSRQ}$
	P54	1	I/O	↑	Bit	$\overline{BUSAK}$
	P55	1	I/O	↑	Bit	R/$\overline{W}$
Port6	P60	1	Output	–	(Fixed)	$\overline{CS0}$
	P61	1	Output	–	(Fixed)	$\overline{CS1}$
	P62	1	Output	–	(Fixed)	$\overline{CS2}$
	P63	1	Output	–	(Fixed)	$\overline{CS3}$ / $\overline{CAS}$
	P64	1	Output	–	(Fixed)	$\overline{RAS}$
	P65	1	Output	–	(Fixed)	$\overline{REFOUT}$
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	$\overline{CTS0}$ / SCLK0
	P83	1	I/O	↑	Bit	TXD1
	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	$\overline{WAIT}$
	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

Table 3.5 (2) I/O Port Setting

Port	Pin Name	Port (I/O) or Function	I/O Register		
			Pn	PnCR	PnFC
Port1	P1 (0 : 7) (Note 1)	Input Port	X	0	–
		Output Port	X	1	
		D (8 : 15)	X	X	
Port2	P2 (0 : 2)	Output Port	X	–	0
		A (16 : 23)	X		1
Port5	$\overline{RD}$	$\overline{RD}$ Output only for External Access	1	–	–
		Always $\overline{RD}$ Output	0		
	P5 (2 : 5)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	P52	$\overline{HER}$ Output	X	1	1
	P53	$\overline{BUSRQ}$ Input (no pull-up)	0	0	1
		$\overline{BUSRQ}$ Input (with pull-up)	1	0	1
	P54	$\overline{BUSAK}$ Output	X	1	1
	P55	R / $\overline{W}$ Output	X	1	1
Port6	P6 (0 : 5)	Output Port	X	–	0
	P60	$\overline{CS0}$ Output	X		1
	P61	$\overline{CS1}$ Output	X		1
	P62	$\overline{CS2}$ Output	X		1
	P63	$\overline{CS3}$ / $\overline{CAS}$ Output (Note 2)	X		1
	P64	$\overline{RAS}$ Output	X		1
	P65	$\overline{REFOUT}$ Output	X		1
Port7	P7 (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
		PGn Output	X	1	1
Port8	P8 (0 : 5)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	P80	TXD0 Output	X	1	1
	P83	TXD1 Output	X	1	1
	P81	RXD0 Input (no pull-up)	0	0	–
		RXD0 Input (with pull-up)	1	0	
	P84	RXD1 Input (no pull-up)	0	0	
		RXD1 Input (with pull-up)	1	0	
	P82	SCLK0 Output	X	1	1
		$\overline{CTS0}$ / SCLK0 Input (no pull-up)	0	0	0
		$\overline{CTS0}$ / SCLK0 Input (with pull-up)	1	0	0
	P85	SCLK1 Output	X	1	1
		$\overline{CTS0}$ / SCLK1 Input (no pull-up)	0	0	0
		$\overline{CTS0}$ / SCLK1 Input (with pull-up)	1	0	0

Note 1: Function is fixed according to input to AM8 / $\overline{16}$ pin.

Note 2: The function of P63 (CS3 / CAS) is selected using CS / WAIT control register B3CS <B3CAS>.

Port	Pin Name	Port (I/O) or Function	I/O Register		
			Pn	PnCR	PnFC
Port9	P9 (0 : 3)	Input Port	X	–	
		AN (0 : 3) Input (Note 3)	X		
PortA	PA (0 : 3)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	PA0	$\overline{\text{WAIT}}$ Input (no pull-up)	0	0	–
		$\overline{\text{WAIT}}$ Input (with pull-up)	1	0	
	PA1	TI0 Input (no pull-up)	0	0	
		TI0 Input (with pull-up)	1	0	
	PA2	TO1 Output	X	1	1
	PA3	TO3 Output	X	1	1
PortB	PB (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	PB0	TI4 / INT4 Input (no pull-up)	0	0	–
		TI4 / INT4 Input (pull-up)	1	0	
	PB1	TI5 / INT5 Input (no pull-up)	0	0	
		TI5 / INT5 Input (pull-up)	1	0	
	PB4	TI6 / INT6 Input (no pull-up)	0	0	
		TI6 / INT6 Input (pull-up)	1	0	
	PB5	TI7 / INT7 Input (no pull-up)	0	0	
		TI7 / INT7 Input (pull-up)	1	0	
	PB2	TO4 Output	X	1	1
	PB3	TO5 Output	X	1	1
	PB6	TO6 Output	X	1	1
	PB7 (Note 4)	INT0 Input (no pull-up)	0	0	–
		INT0 Input (with pull-up)	1	0	

Note 3: When P9 (0 : 3) are used as Input channels of the A / D converter, channels are selected using ADMOD＜ADCHn＞.

Note 4: When PB7 pin is used as INT0 pin, set IIMC＜IOIE＞to “1”. (enable interrupt Input.)

3.5.1 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR. 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 a data bus (D8 to 15).

TMP95C061B determines the port function and the data bus function according to the input state of AM8/$\overline{16}$ pin after reset. When AM8/$\overline{16}$ is set to low level, the data bus functions. When AM8/$\overline{16}$ is set to high level, the port functions. When using as the data bus (AM8/$\overline{16}$ = "0"), the bit of P1CR register should not be set to 1.

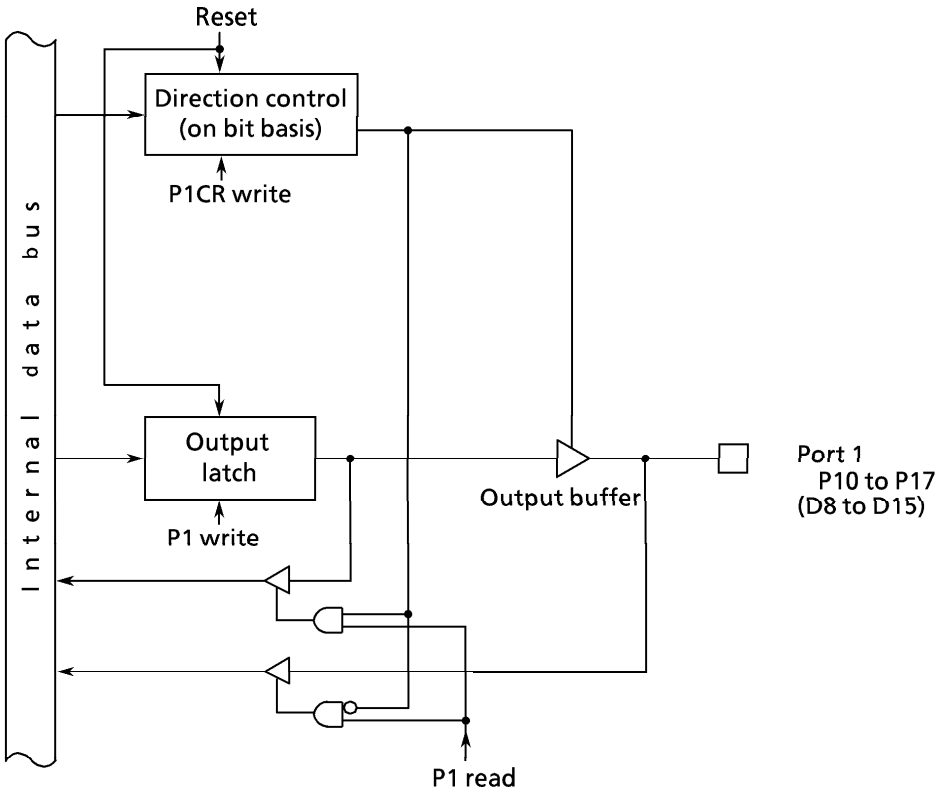


Figure 3.5 (1) Port 1

Port 1 Register

		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

		7	6	5	4	3	2	1	0
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 / $\overline{16}$ / P1CR <P1XC>	0	1
0	Data bus (D15 to 8)	Input port
1	Don't set	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 output-only port. A reset sets all bits of the output latches in the port 2 register (P2) to "1" and all port pins output "1".

In addition to functioning as a general-purpose output port, port 2 can also function as address bus (A16 to 23). The port function is specified by function register P2FC. Port pins can be selected individually as either output ports or address bus pins.

In TMP95C061B with external ROM, a reset sets all bits of the function register to "1", and sets the pins as address bus pins (A16 to A23).

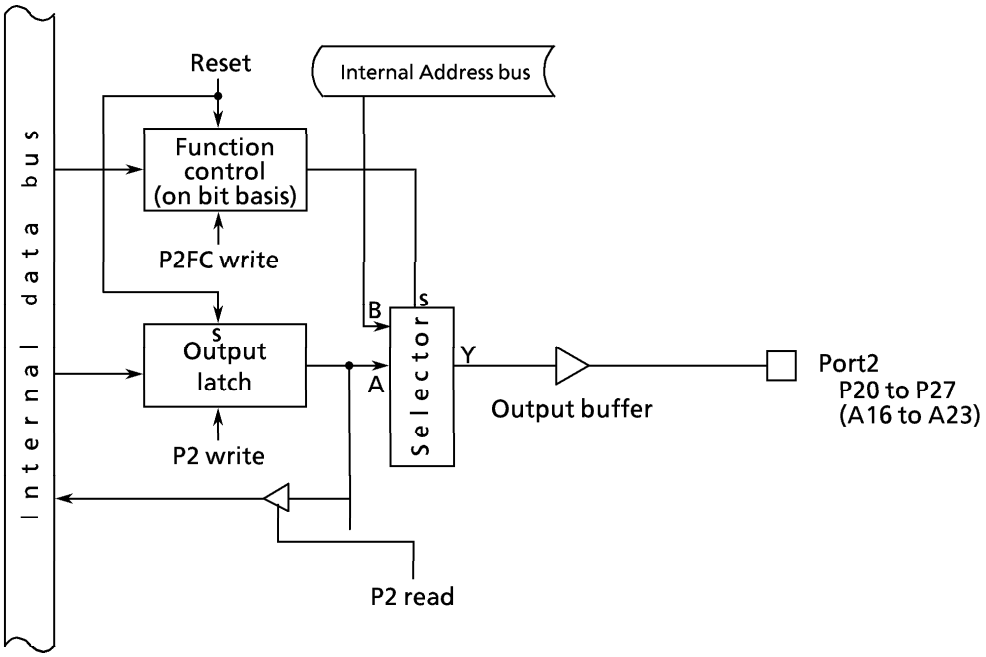


Figure 3.5 (3) Port2

Port 2 Register

	7	6	5	4	3	2	1	0
bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R / W							
After reset	(Output latch register is set to "1")							

P2 (0006H)

Port 2 Function Register

	7	6	5	4	3	2	1	0
bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	1	1	1	1	1	1	1	1
Function	0 : Port, 1 : Address bus (A23 to A16)							

P2FC (0009H)

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

Figure 3.5 (4) Registers for Port 2

3.5.3 Port5 (P52 to P55)

Port 5 is a 4-bit general-purpose I/O port. I/O can be set on a bit basis using control register P5CR and function register P5FC. Resetting does the following :
Resets all bits of the Port 5 output latch, the control register P5CR and the function register P5FC to "0" and sets each port input mode with pull-up resisters.

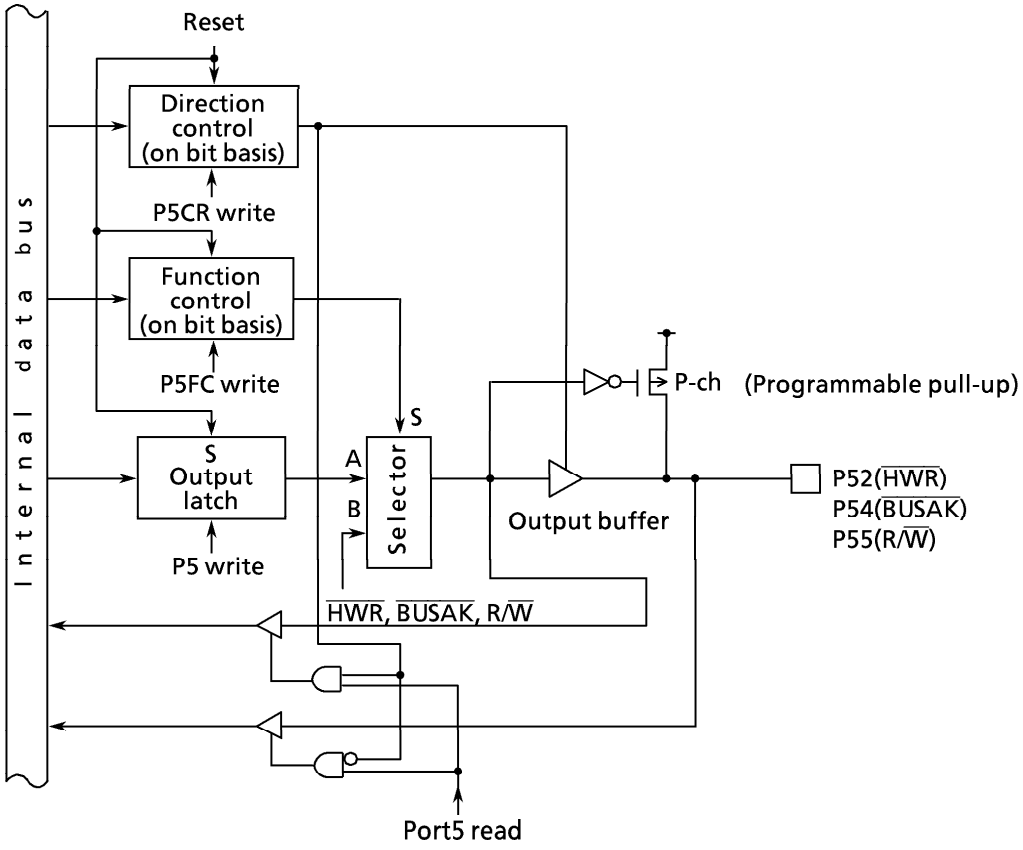


Figure 3.5 (5) Port5 (P50, P51, P52, P54, P55)

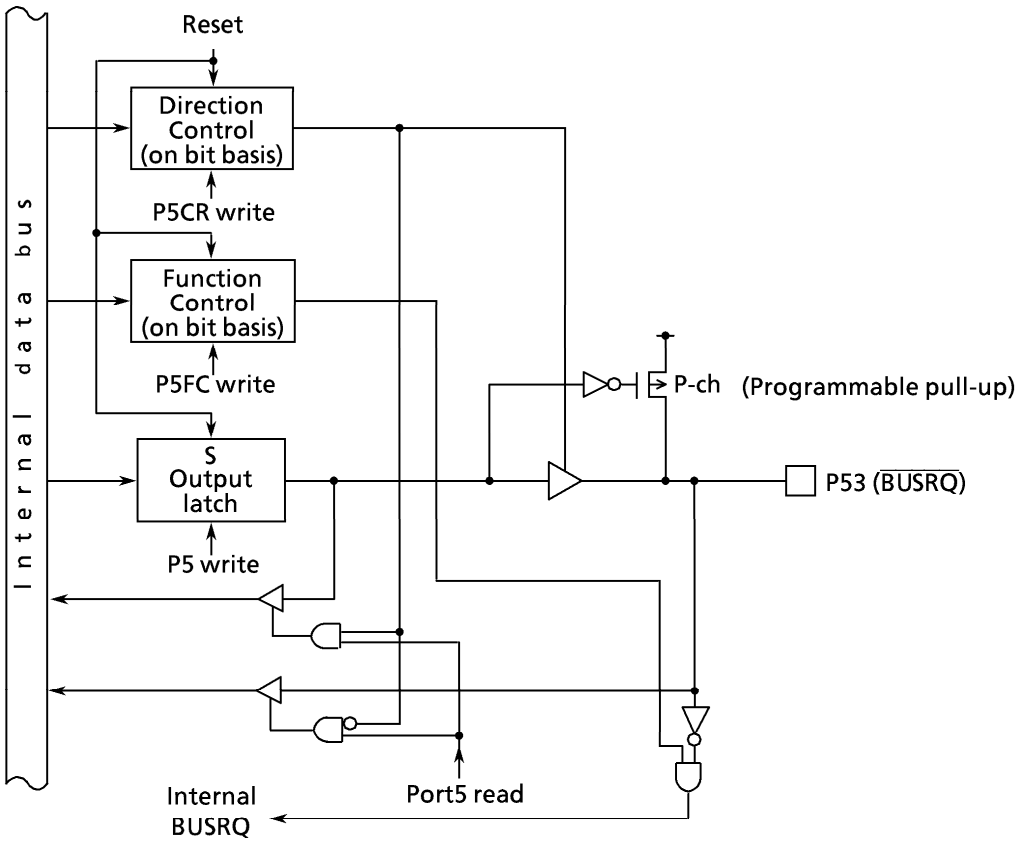


Figure 3.5 (6) Port5 (P53)

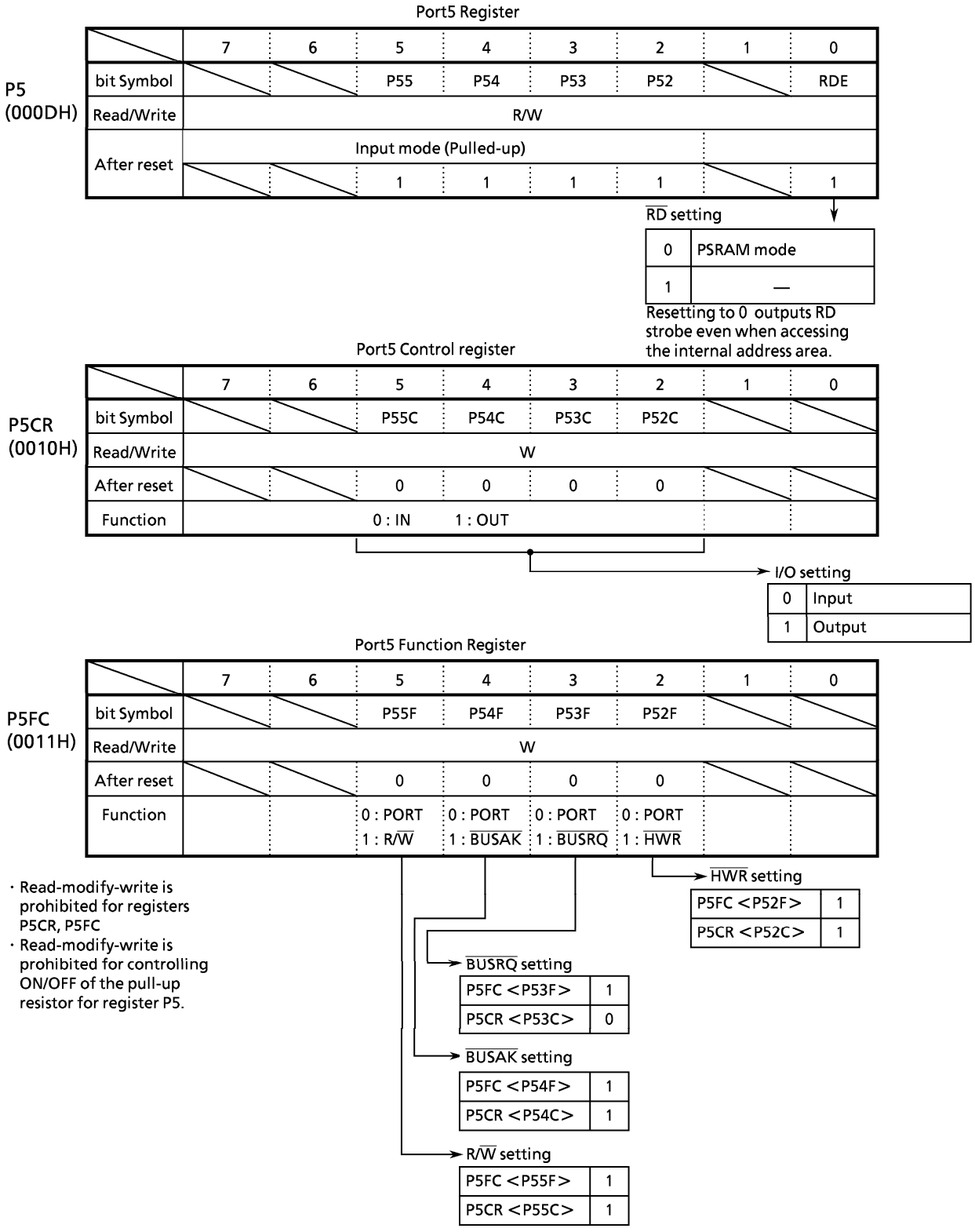


Figure 3.5 (7) Registers for Port5

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 ($\overline{CS0}$ to $\overline{3}$, $\overline{CAS}$, $\overline{RAS}$ and $\overline{REFOUT}$). After resetting, each port operates as output port.

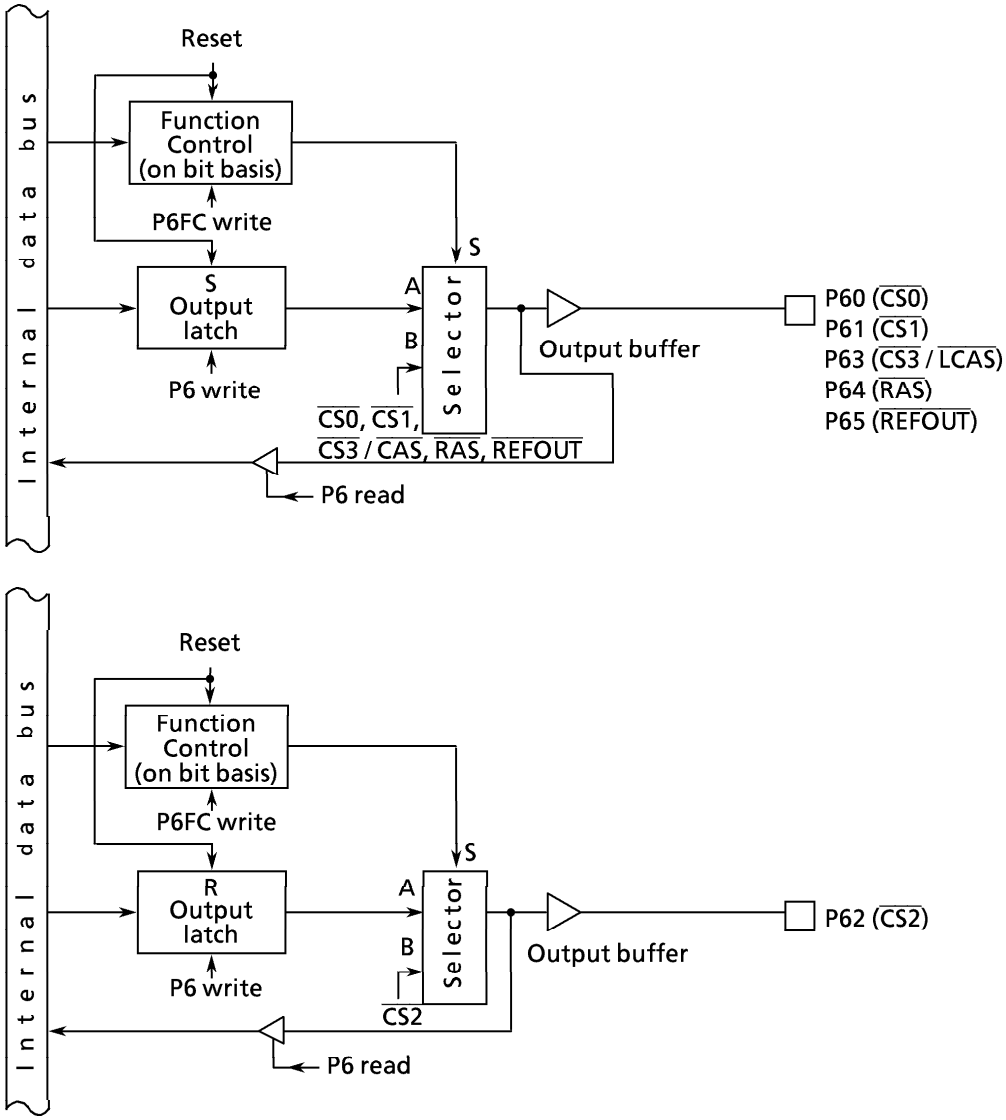


Figure 3.5 (8) Port6

Port 6 Register

P6 (0012H)

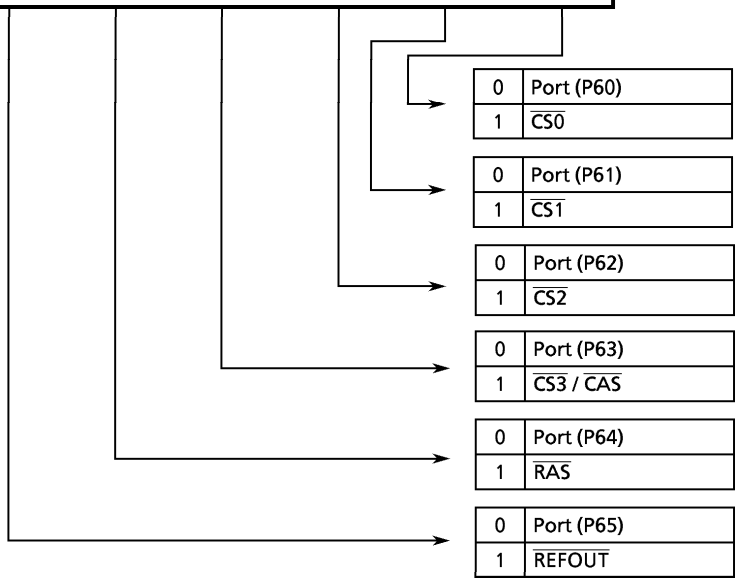
	7	6	5	4	3	2	1	0
bit Symbol			P65	P64	P63	P62	P61	P60
Read/Write			R/W					
After reset			1	1	1	0	1	1

Port 6 Function Register

P6FC (0015H)

	7	6	5	4	3	2	1	0
bit Symbol			P65F	P64F	P63F	P62F	P61F	P60F
Read/Write			W					
After reset			0	0	0	0	0	0
Function			0 : PORT		1 : $\overline{CS}$ / $\overline{CAS}$, $\overline{RAS}$, $\overline{REFOUT}$			

Read-modify-write is prohibited for registers P6FC



Note: The function of P63 ($\overline{CS3}$ / $\overline{CAS}$) 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 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 1 in the corresponding bit of the 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.

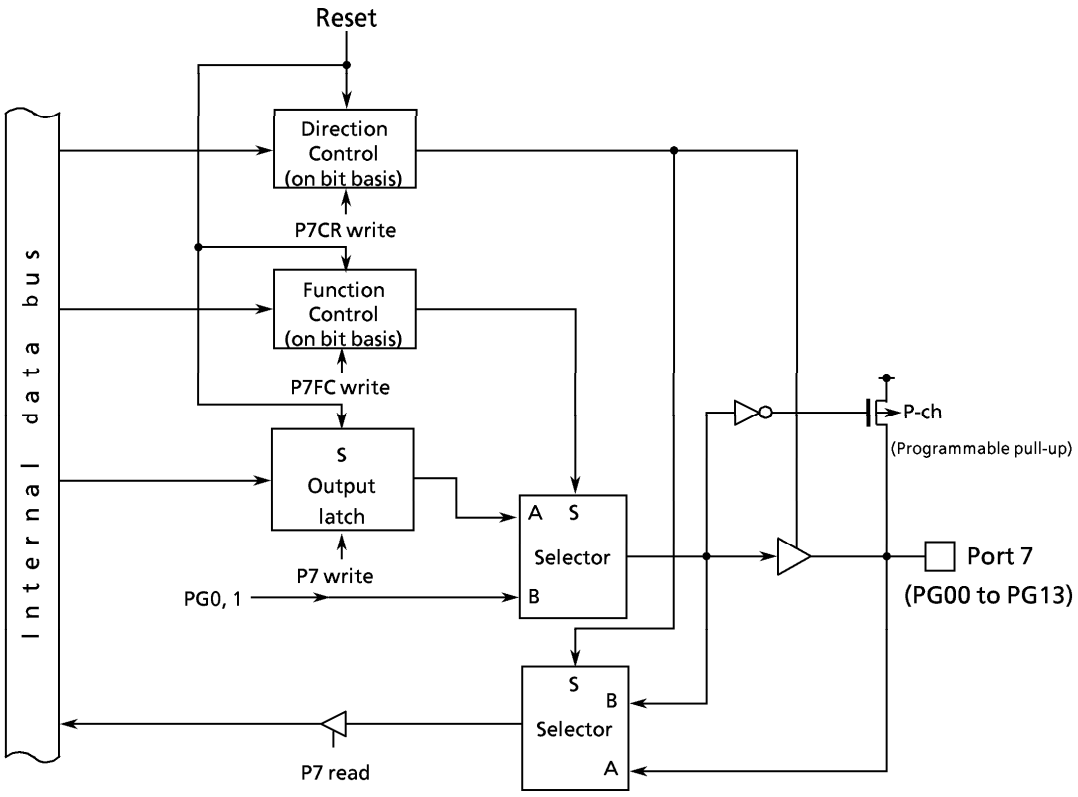


Figure 3.5 (10) Port 7

Port 7 Register

P7 (0013H)

	7	6	5	4	3	2	1	0
bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W							
After reset	Input mode (with pull-up resistor)							
	1	1	1	1	1	1	1	1

Port 7 Control Register

P7CR (0016H)

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

Port 7 I/O setting

0	Input
1	Output

Port 7 Function Register

P7FC (0017H)

	7	6	5	4	3	2	1	0
bit Symbol	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : PORT 1 : PG1-OUT				0 : PORT 1 : PG0-OUT			

- Read-modify-write is prohibited for registers P7CR and P7FC
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P7.

Port 7 Function setting

0	General-purpose port
1	Stepping motor control / Pattern generation port

Figure 3.5 (11) Refister for Port 7

3.5.6 Port 8 (P80 to P85)

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

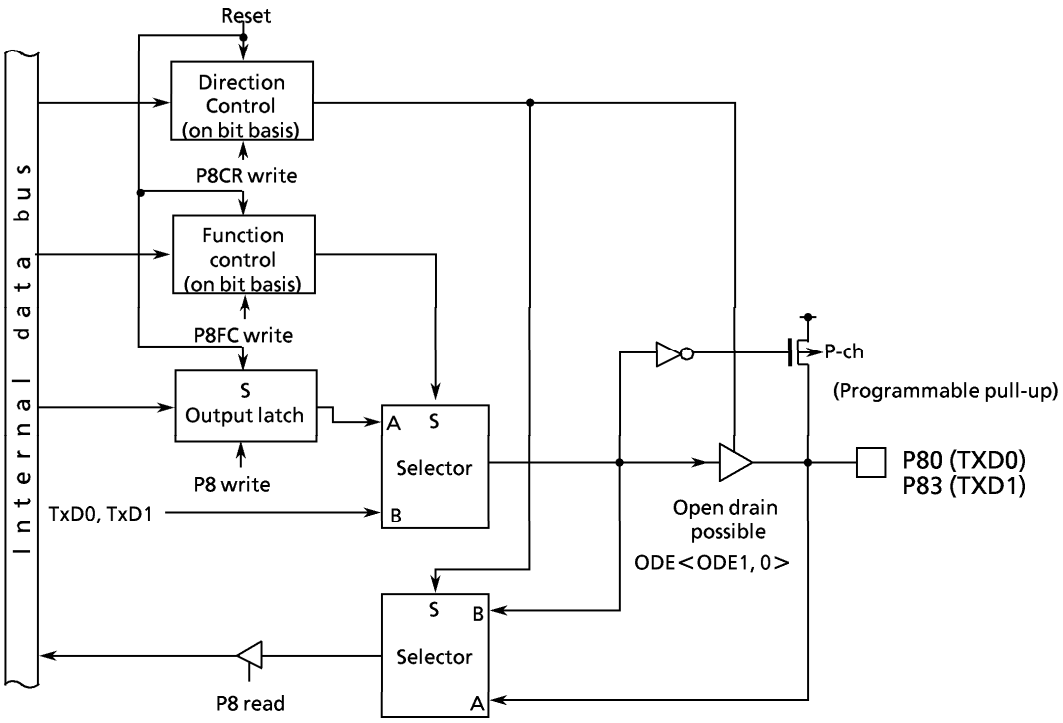


Figure 3.5 (12) Port 80, 83

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

P81 and P84 are I/O ports, and also used as RXD input pins for serial channels.

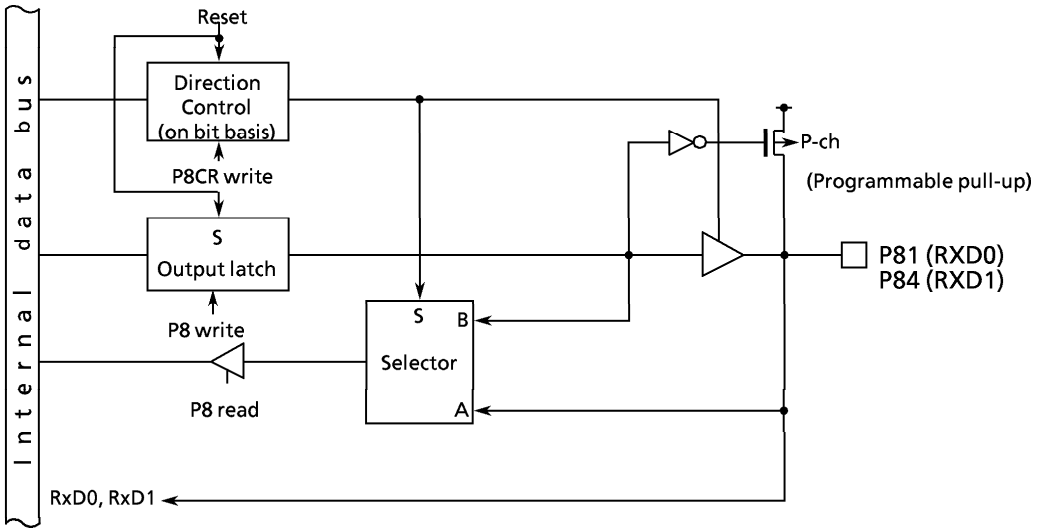


Figure 3.5 (13) Port 81, 84

(3) Port 82 ($\overline{\text{CTS0}}$ / SCLK0)

P92 is an I/O port, and also used as a $\overline{\text{CTS}}$ input pin or as a SCLK0 I/O pin for serial channels.

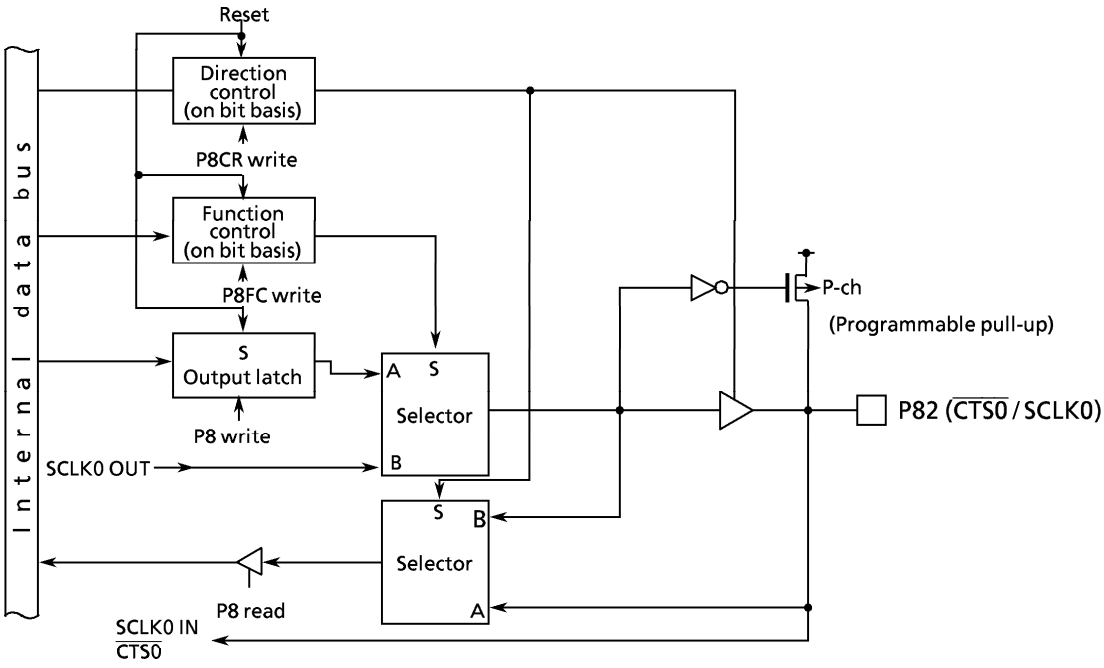


Figure 3.5 (14) Port 82

(4) Port 85 (SCLK1)

P85 is general-purpose I/O port. It is also used as a SCLK1 I/O pin for serial channel 1.

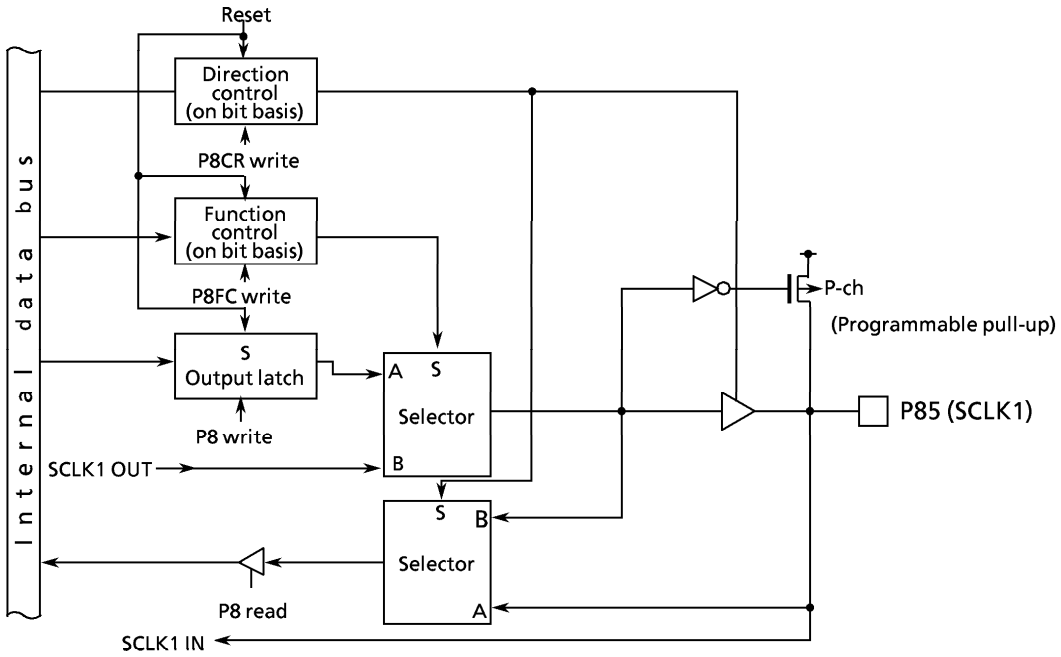


Figure 3.5 (15) Port 85

Port 8 Register

P8 (0018H)

	7	6	5	4	3	2	1	0
bit Symbol			P85	P84	P83	P82	P81	P80
Read/Write			R/W					
After reset			Input mode (Pulled-up)					
			1	1	1	1	1	1

Port 8 Control Register

P8CR (001AH)

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

Port 8 I/O setting

0	Input
1	Output

Port 8 Function Register

P8FC (001BH)

	7	6	5	4	3	2	1	0
bit Symbol			P85F		P83F	P82F		P80F
Read/Write			W		W	W		W
After reset			0		0	0		0
Function			0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0		0 : PORT 1 : TxD0

- *Read-modify-write is* prohibited for registers P8CR and P8FC
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P8.

P80 TxD0 output setting (Note)

P8FC <P80F>	1
P8CR <P80C>	1

P82 SCLK0 output setting

P8FC <P82F>	1
P8CR <P82C>	1

P85 SCLK1 output setting

P8FC <P85F>	1
P8CR <P85C>	1

P83 TxD1 output setting (Note)

P8FC <P83F>	1
P8CR <P83C>	1

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.
For example, even when the pins are used as input port pins (P81/P84), the input data for P81/P84 are input to SIO as a serial receive data (RxD0/RxD1).

Figure 3.5 (16) Register for Port 8

3.5.7 Port 9 (P90 to P93)

Port 9 is a 4-bit Input port, also used as analog input pins for the internal A/D Converter.

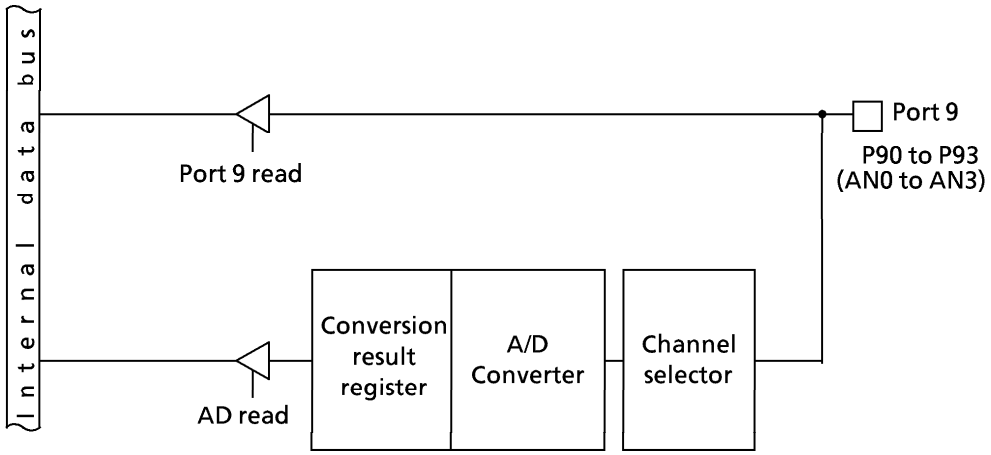


Figure 3.5 (17) Port 9

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			

Note : Select the input channels for the A/D converter in A/D converter mode register ADMOD.

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 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 an wait input pin $\overline{\text{WAIT}}$; Port A1 as an 8-bit timer input (TI0), Port A2 as an 8-bit timer output (TO1), and Port A3 as an 8-bit timer 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.

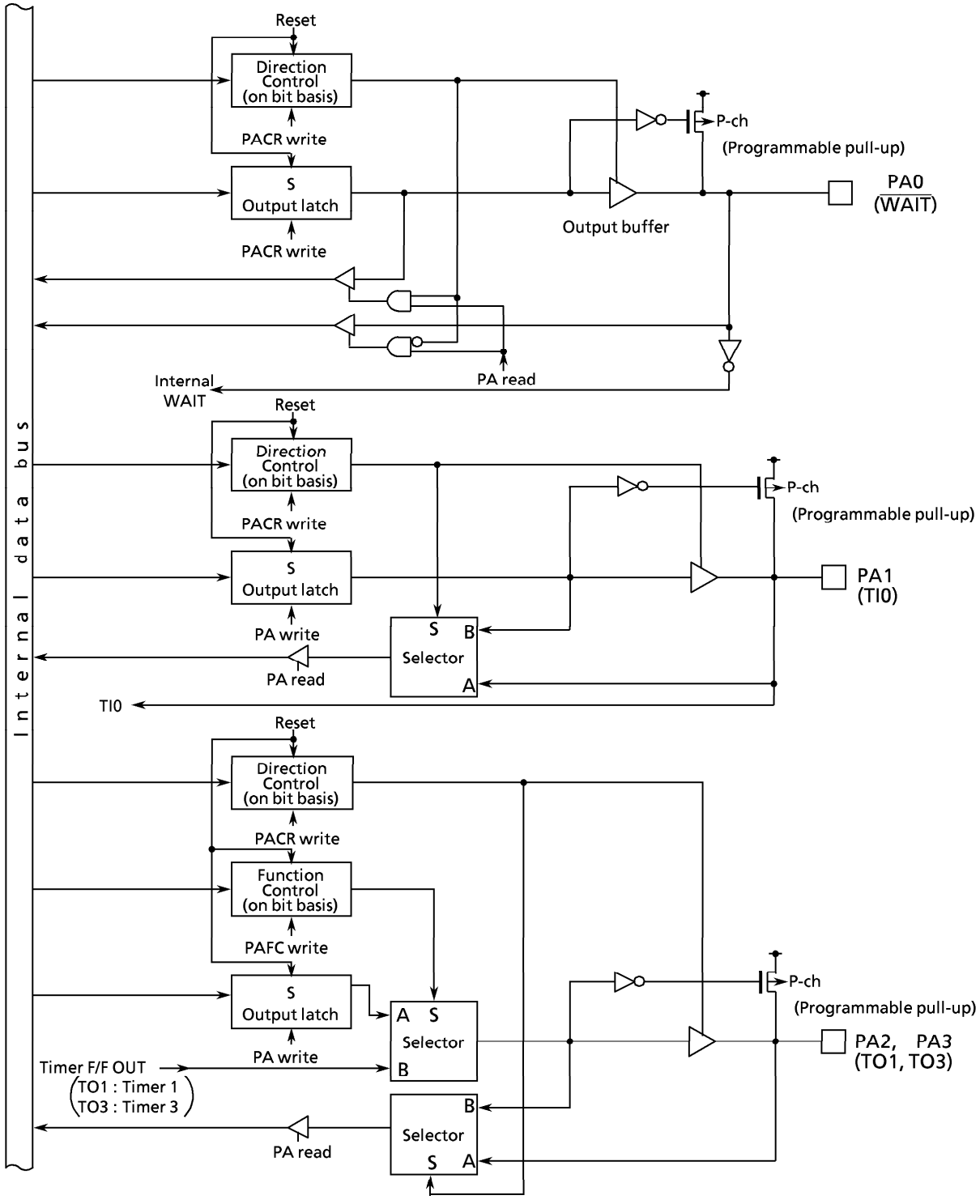


Figure 3.5 (19) Port A

Port A Register

PA (001EH)

	7	6	5	4	3	2	1	0
bit Symbol					PA3	PA2	PA1	PA0
Read/Write					R/W			
After reset					Input mode (Pulled-up)			
					1	1	1	1

Port A Control Register

PACR (002CH)

	7	6	5	4	3	2	1	0
bit Symbol					PA3C	PA2C	PA1C	PA0C
Read/Write					W			
After reset					0	0	0	0
Function					0 : IN		1 : OUT	

Port A I/O setting

0	Input
1	Output

Port A Function Register

PAFC (002DH)

	7	6	5	4	3	2	1	0
bit Symbol					PA3F	PA2F		
Read/Write					W			
After reset					0	0		
Function					0 : PORT 1 : TO3	0 : PORT 1 : TO1		

- Read-modify-write is prohibited for registers PACR and PAFC
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register PA.

Timeout 1 output setting

PAFC <PA2F>	1
PACR <PA2C>	1

Timeout 3 output setting

PAFC <PA3F>	1
PACR <PA3C>	1

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 (TI1).

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

(1) PB0 to PB6

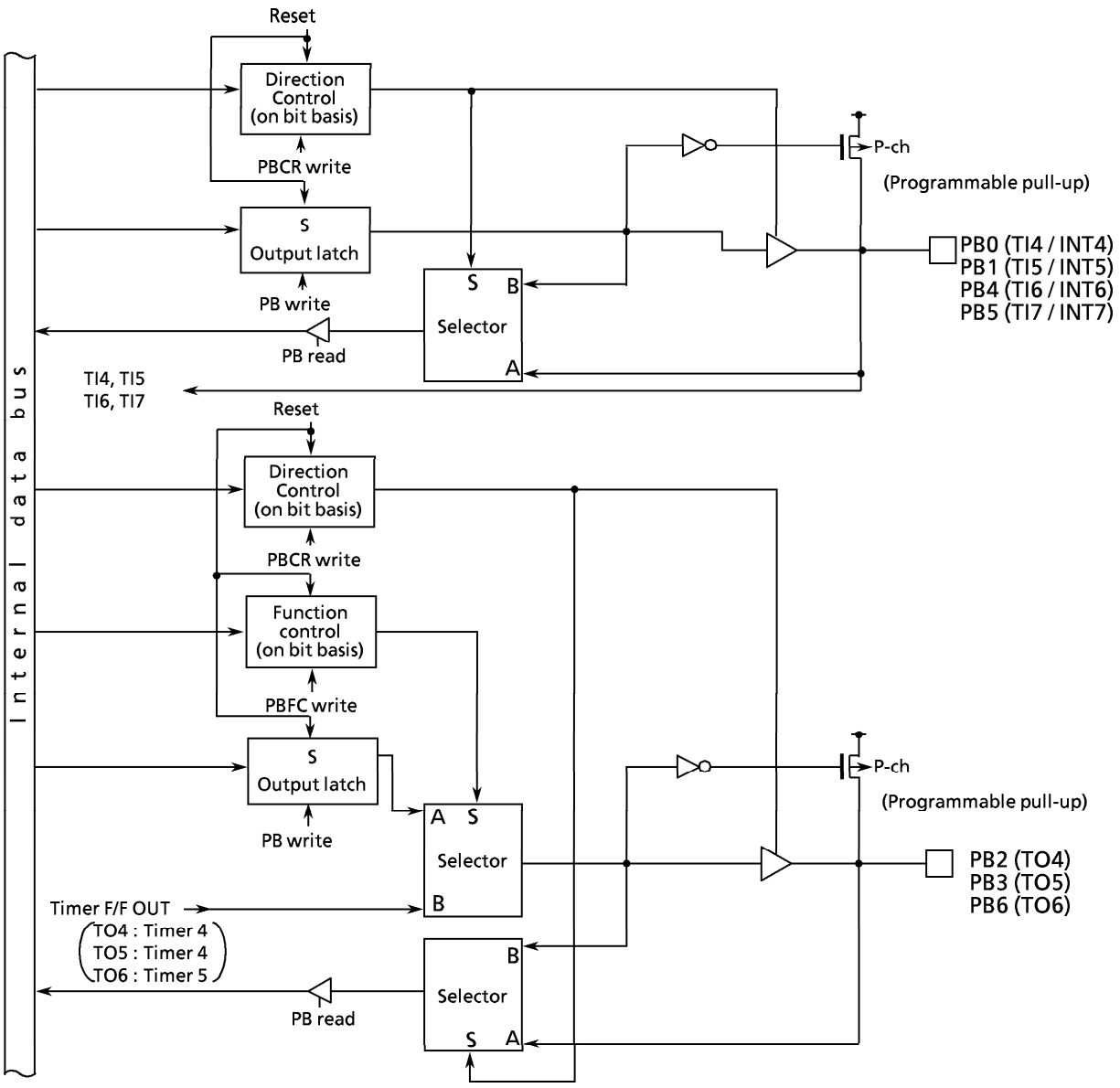


Figure 3.5 (21) Port B (PB0 to PB6)

(2) PB7 (INT0)

Port B7 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

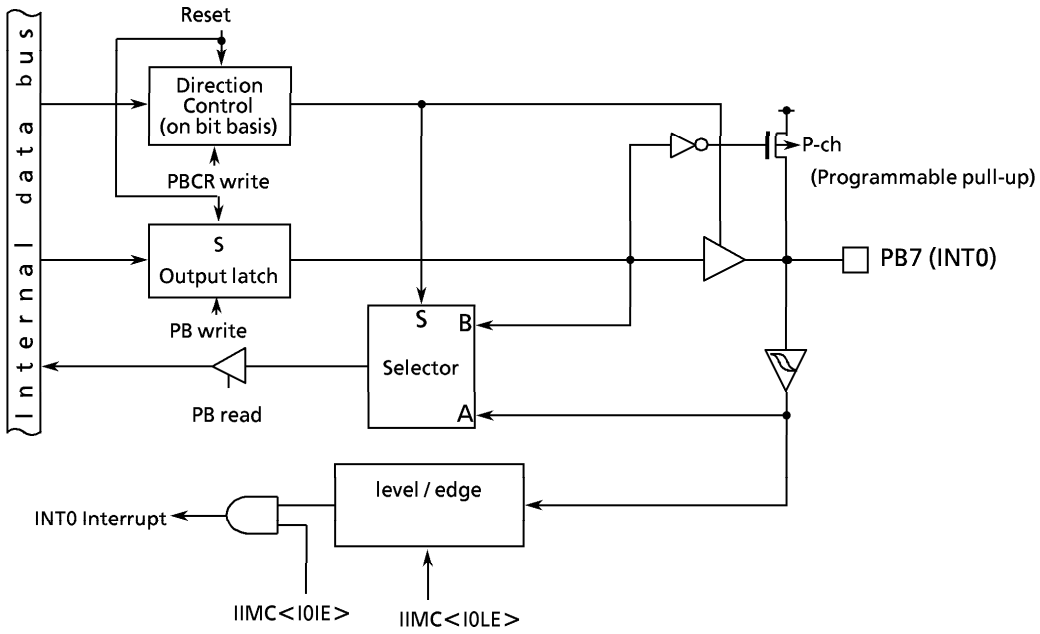


Figure 3.5 (22) Port B7

Port B Register

PB (001FH)

	7	6	5	4	3	2	1	0
bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	R/W							
After reset	Input mode (Pulled-up)							
	1	1	1	1	1	1	1	1

Port BControl Register

PBCR (002EH)

	7	6	5	4	3	2	1	0
bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN				1 : OUT			

Port B I/O setting

0	Input
1	Output

Port B Function Register

PBFC (002FH)

	7	6	5	4	3	2	1	0
bit Symbol		PB6F			PB3F	PB2F		
Read/Write		W			W	W		
After reset		0			0	0		
Function		0 : PORT 1 : TO6			0 : PORT 1 : TO5	0 : PORT 1 : TO4		

- Read-modify-write is prohibited for registers PBCR and PBFC
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register PB.

PB2 Timerout 4 output setting

PBFC <PB2F>	1
PBCR <PB2C>	1

PB3 Timerout 5 output setting

PBFC <PB3F>	1
PBCR <PB3C>	1

PB6 Timerout 6 output setting

PBFC <PB6F>	1
PBCR <PB6C>	1

Note: PB0 / TI4, PB1 / TI5, PB4 / TI6 and PB5 / TI7 pins do not have a register changing Port / Function. For example, even when the pins are used as input port pins, data are input to the 16-bit timer. When PB7 / INT0 pin is used as an INT0 pin, set PBCR<PB7C> to "0" and IIMC <I0IE> to "1".

Figure 3.5 (23) Register for Port B

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

TMP95C061B has a built-in chip select / wait controller used to control chip select ($\overline{CS0}$ to $\overline{CS3}$ pins), wait ($\overline{WAIT}$ pin), and data bus size (8 or 16 bits) for any of the four block address areas.

And there is an AM8 / $\overline{16}$ pin which selects external data bus width for TMP95C061B.

3.6.1 Control Register

Table 3.6 (1) shows control registers.

Each block address area is controlled using CS / WAIT control register. Start address register (MSAR0 to MSAR3) and address mask register (MAMR0 to 3).

Table 3.6 (1) Chipselect / wait control register

		7	6	5	4	3	2	1	0
B0CS (0068H) (Prohibit RMW)	bit Symbol				B0E	—	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	
B1CS (0069H) (Prohibit RMW)	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	
B2CS (006AH) (Prohibit RMW)	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	
B3CS (006BH) (Prohibit RMW)	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: $\overline{CS3}$ output 1: $\overline{CAS}$ output	0: 16 BIT 1: 8 BIT	00: 2 WAIT 01: 1 WAIT 10: 1 WAIT + n 11: 0 WAIT	
BEXCS (006CH) (Prohibit RMW)	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	

Note : Read-modify-write is prohibited for registers B0CS, B1CS, B2CS, B3CS and BEXCS.

(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).

(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 / $\overline{16}$ = 0). In 8 bit bus mode (AM8 / $\overline{16}$ = 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 / $\overline{16}$).

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.

(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 $\overline{WAIT}$ pin status. Setting them to 01 inserts a 1-state wait regardless of the $\overline{WAIT}$ status. Setting them to 10 inserts a 1-state wait and samples the $\overline{WAIT}$ pin status. If the pin is low, inserting the wait maintains the bus cycle until the pin goes high. Setting them to 11 completes the bus cycle without a wait regardless of the $\overline{WAIT}$ pin status.

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

Note: In case of competition of accessing and refreshing to DRAM, TMP95C061B automatically inserts refresh cycle in addition to settled wait cycle.

(4) CS / CAS Waveform select

Bit3 of Control register B3CS is used to specify waveform mode output from the chip select pin ($\overline{CS3}$ / $\overline{CAS}$). Setting this bit to 0 specifies $\overline{CS3}$ waveforms; setting it to 1 specifies $\overline{CAS}$ 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	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

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

(5) Extra CS area bus size / 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 registers. This register has no master enable bit, so always enable to unspecified area. Each bit has same meaning as BxCS.

(6) Accessing 16M-byte Area / Address Setting Area

Setting B2CS<B2M> = 0 selects CS2 in the 16M-byte area (000080H to FFFFFFH). Setting B2CS<B2M> = 1 selects CS2 according to the setting area for start address register MSAR2 and address mask 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) and address mask 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 outputted from the corresponding chip select pin ($\overline{CS0}$ to $\overline{CS3}$) if it is enabled (BxE = "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.

When the set address area overlaps with the internal I/O area, the functions as the internal I/O area take priority of the set address area.

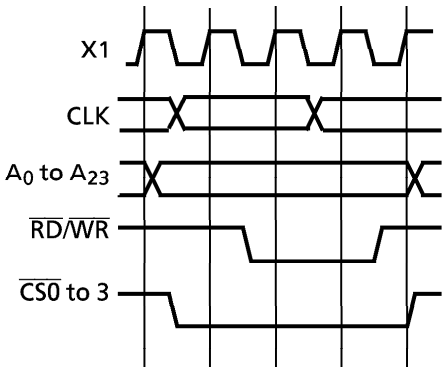


Figure 3.6 (1) Chip Select ($\overline{CS0}$ to $\overline{CS3}$) Operation Timing

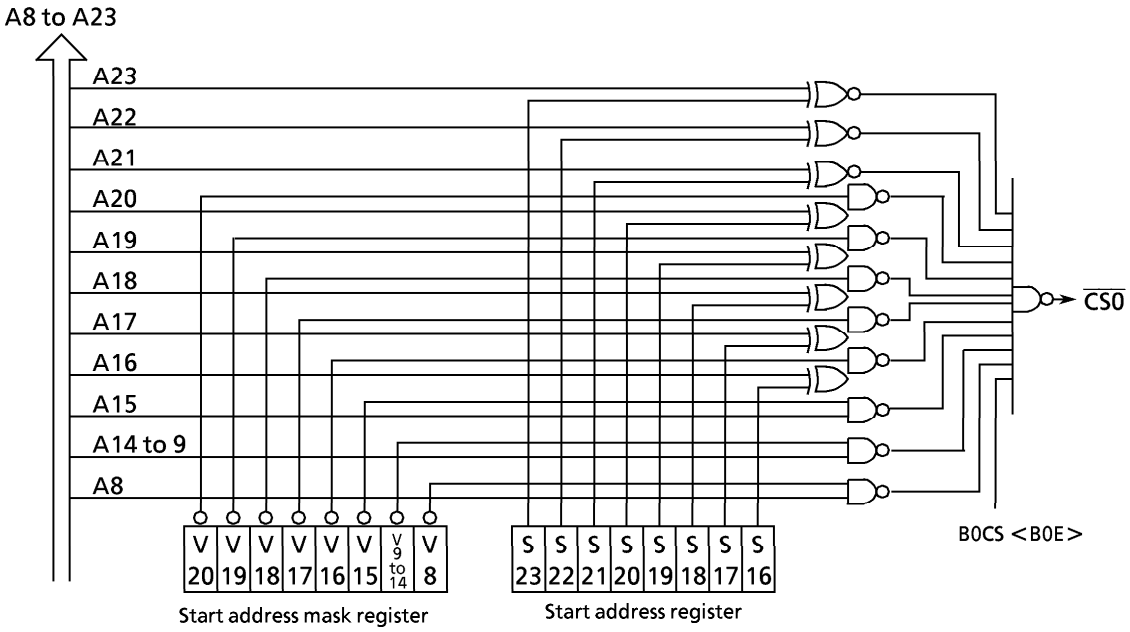


Figure 3.6 (2) $\overline{CS0}$ Address Decode Block Diagram

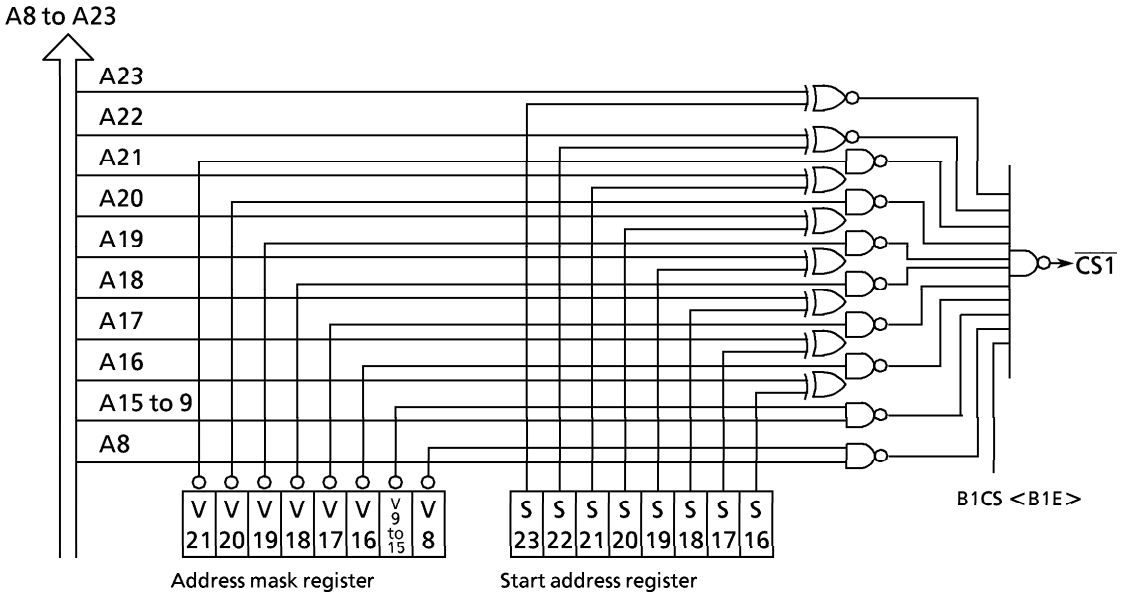


Figure 3.6 (3) $\overline{CS1}$ Address Decode Block Diagram

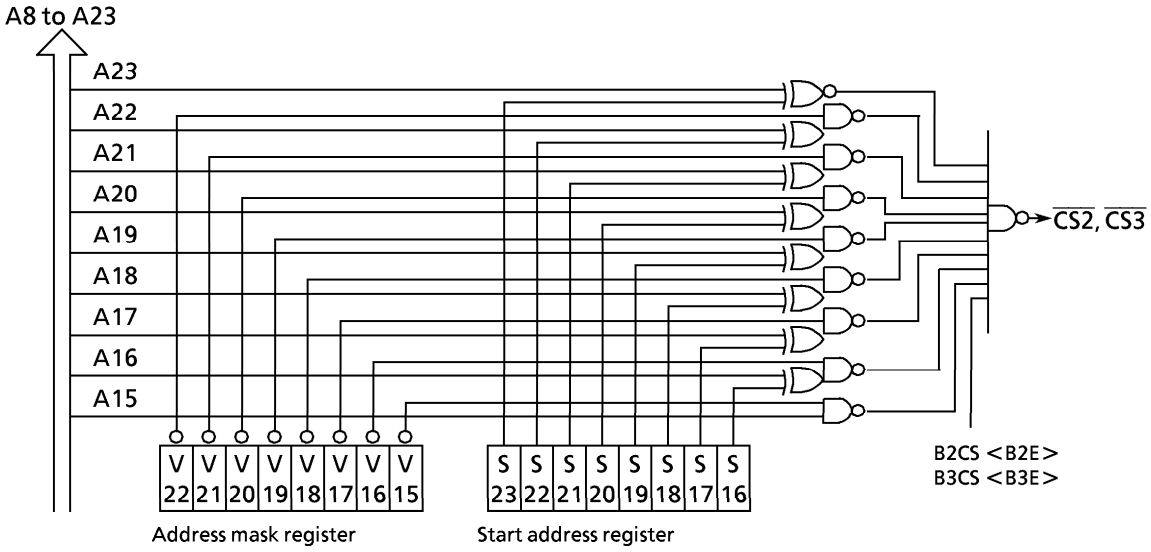


Figure 3.6 (4) $\overline{CS2} \cdot \overline{CS3}$ Address Decode Block Diagram

(1) Memory start address register / Memory address mask 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	1
	Function	Set start address A23 to A16							

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

Table 3.6 (3) Memory Start Address Register

Memory address mask register ($\overline{CS0}$)

		7	6	5	4	3	2	1	0
MAMR0 (003DH)	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 $\overline{CS0}$ address A8 to A20

Memory address mask register ($\overline{CS1}$)

		7	6	5	4	3	2	1	0
MAMR1 (003FH)	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 $\overline{CS1}$ address A8 to A21

Memory address mask register ($\overline{CS2}$, $\overline{CS3}$)

		7	6	5	4	3	2	1	0
MAMR2 (005DH) / MAMR3 (005FH)	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 $\overline{CS2}$ to $\overline{CS3}$ address A15 to A22

Table 3.6 (4) Memory Address Mask Register

MMSAR0 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 MASR0 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 <S16>=0, the comparison of the value of <S16> and address A16 is enabled and CS0 is enabled only when 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 "0FFH", and sets the control register bits B0E, B1E, to "0". So chip select $\overline{CS0}$, $\overline{CS1}$ and $\overline{CS3}$ are disable after resetting, while Bit B2E=1, B2M=0 and $\overline{CS2}$ is enable for memory area 000080H to 0FFFFFFH (16Mbyte).

(2) How to set the start address

The address decoder is output by specifying the start address for $\overline{CS}$ 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 cue to the value of the MAMR.

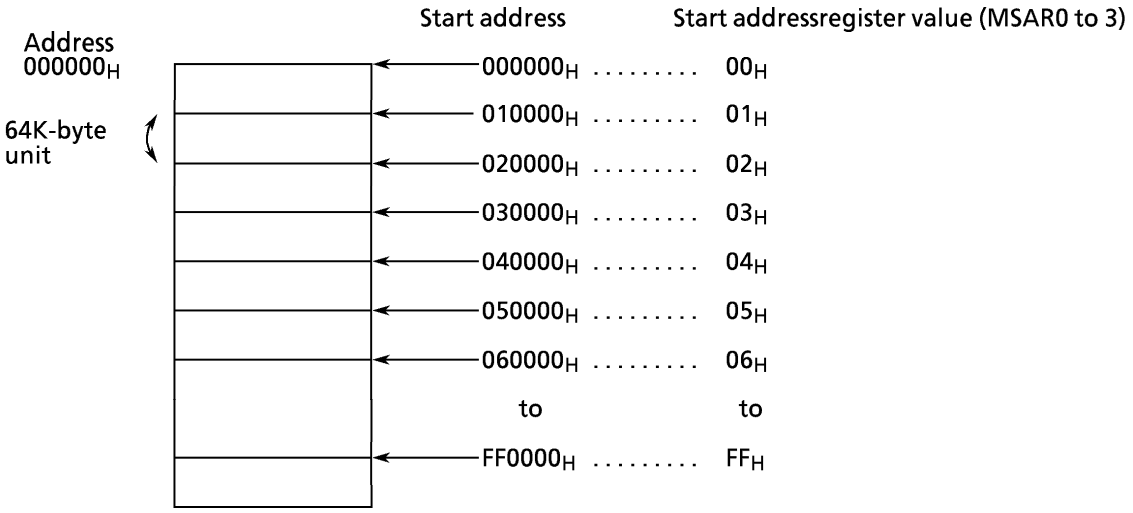


Figure 3.6 (5) Where to Set Start Address

(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 (Figures 3.6 (2) to (4)), $\overline{CS0}$, $\overline{CS1}$, or $\overline{CS2}$ / $\overline{CS3}$ can specify the address area for which the chip select signal can be output depending on whether to compare the addresses A8 to A20, A8 to A21, or A15 to A22 respectively.

CS \ Size	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	○	○	○	○	○		
CS1	○	○		○	○	○	○	○	○	○	
CS2			○	○	○	○	○	○	○	○	○
CS3			○	○	○	○	○	○	○	○	○

Figure 3.6 (6) Chip Select and Space Size

(4) Start address / address space setting procedure

① Set memory start address register (MSARx)
(Set start address)

② Set memory start address mask register (MAMRx)
(Set address area)

③ Set control register (BxCS)
data bus width, number of waits, enable / disable of the area

(Example)

When setting the $\overline{CS0}$ area to 64Kbyte (010000 to 01FFFFH), 16 bit data width and non-wait,

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

3.7 Dynamic RAM (DRAM) Controller

TMP95C061B incorporates a DRAM controller for interface with×8 / 16 bit DRAM. The DRAM controller consists of a control circuit to refresh the DRAM, an access circuit for reading and writing, and a row / column address multiplexer.

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

Control Register

DREFCR (005AH)

	7	6	5	4	3	2	1	0
bit Symbol	DMI	RS2	RS1	RS0	RW2	RW1	RW0	RC
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	Dummy cycle 0: Prohibit 1: Execute	Refresh cycle insertion interval 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0: Not inserted 1: Inserted

Refresh cycle control

0	not insert refresh cycle
1	insert refresh cycle

Refresh cycle width control

RW2	RW1	RW0	Refresh cycle width
0	0	0	2 states
0	0	1	3 states
0	1	0	4 states
0	1	1	5 states
1	0	0	6 states
1	0	1	7 states
1	1	0	8 states
1	1	1	9 states

Refresh cycle insertion interval See Table 3.7 (2).

Dummy cycle control

0	Prohibit dummy cycle
1	Execute dummy cycle

Figure 3.7 (1) Refresh Control Register

Read-modify-write is prohibited for registers DMEMCR

DMEMCR (005BH)

	7	6	5	4	3	2	1	0
bit Symbol	SRFC	–	BRM	MACM	MUXE	MUXW1	MUXW0	MAC
Read/Write	W	–	R / W					
After reset	1	–	0	0	0	0	0	0
Function	Self-Refresh 0: Execute 1: Release	–	bus release mode control 0 : release 1 : not release	memory access control 0 : nomal 1 : slow	address multiple 0: disable 1: Enable	multiplexed address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		memory access control 0: disable 1: Enable

memory access control

0	disable
1	enable

multiplexed address length control
(enable only when MUXE = "1")

MUXW1	MUXW0	multiplexed address length
0	0	8 bit
0	1	9 bit
1	0	10 bit
1	1	11 bit

address multiplex control

0	disable
1	enable

memory access control

0	Nomal access mode
1	Slow access mode

bus release mode Control

0	DRAM control signals are also released
1	DRAM control signals are not released

Self-refresh control

0	Execute self-refresh
1	Release self-refresh

Figure 3.7 (2) DRAM memory access control register

Operation description

(1) Memory access control

Access control block is enable when DMEMCR<MAC> = 1. And then DRAM control signals ($\overline{RAS}$, $\overline{CAS}$ and $\overline{REFOUT}$) are output during the time CPU accesses $\overline{CS3}$ 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 insertion 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.

Figure 3.7 (3), (4) shows 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)

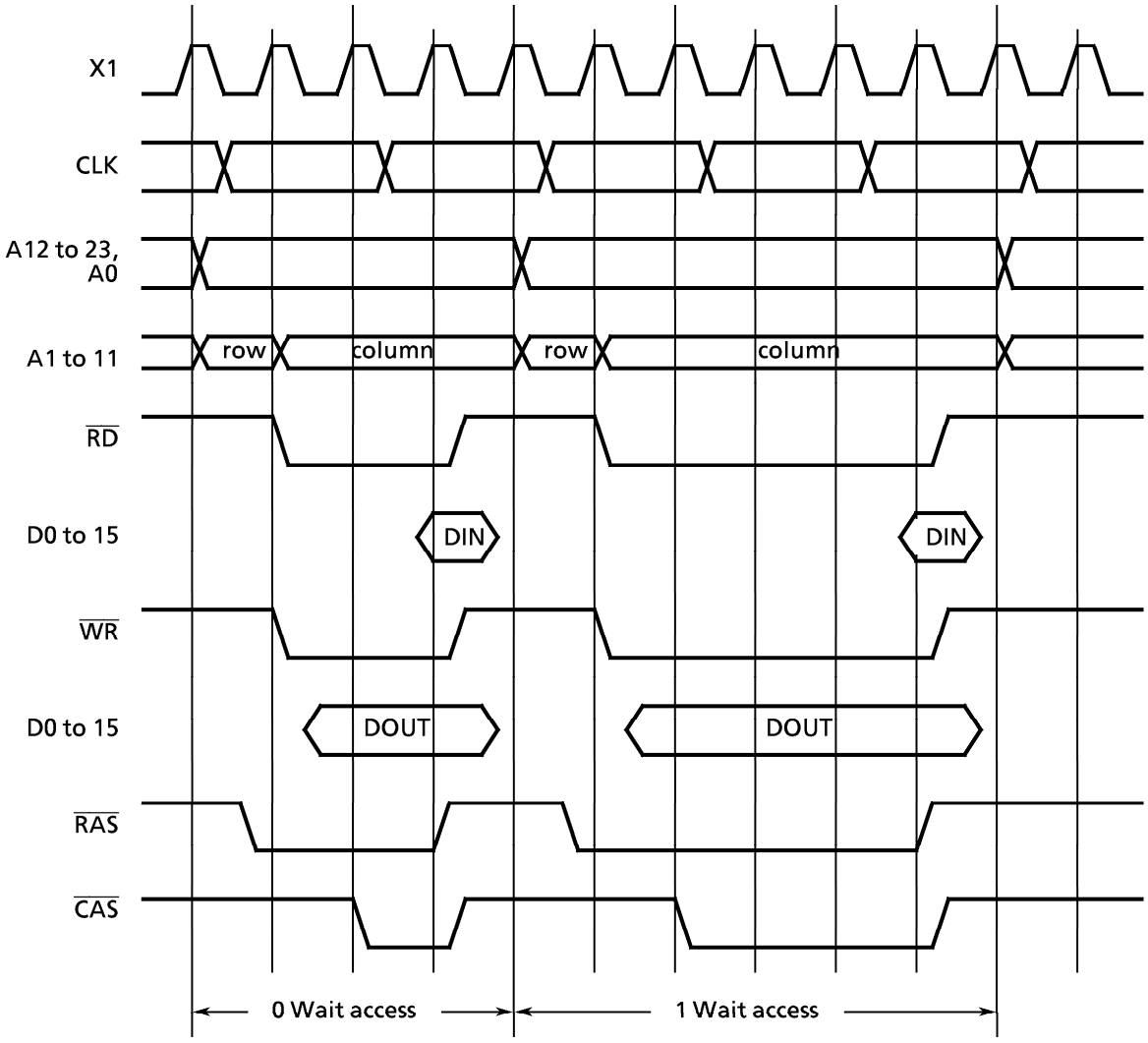


Figure 3.7 (3) DRAM Access Timing (Nomal Access Mode)

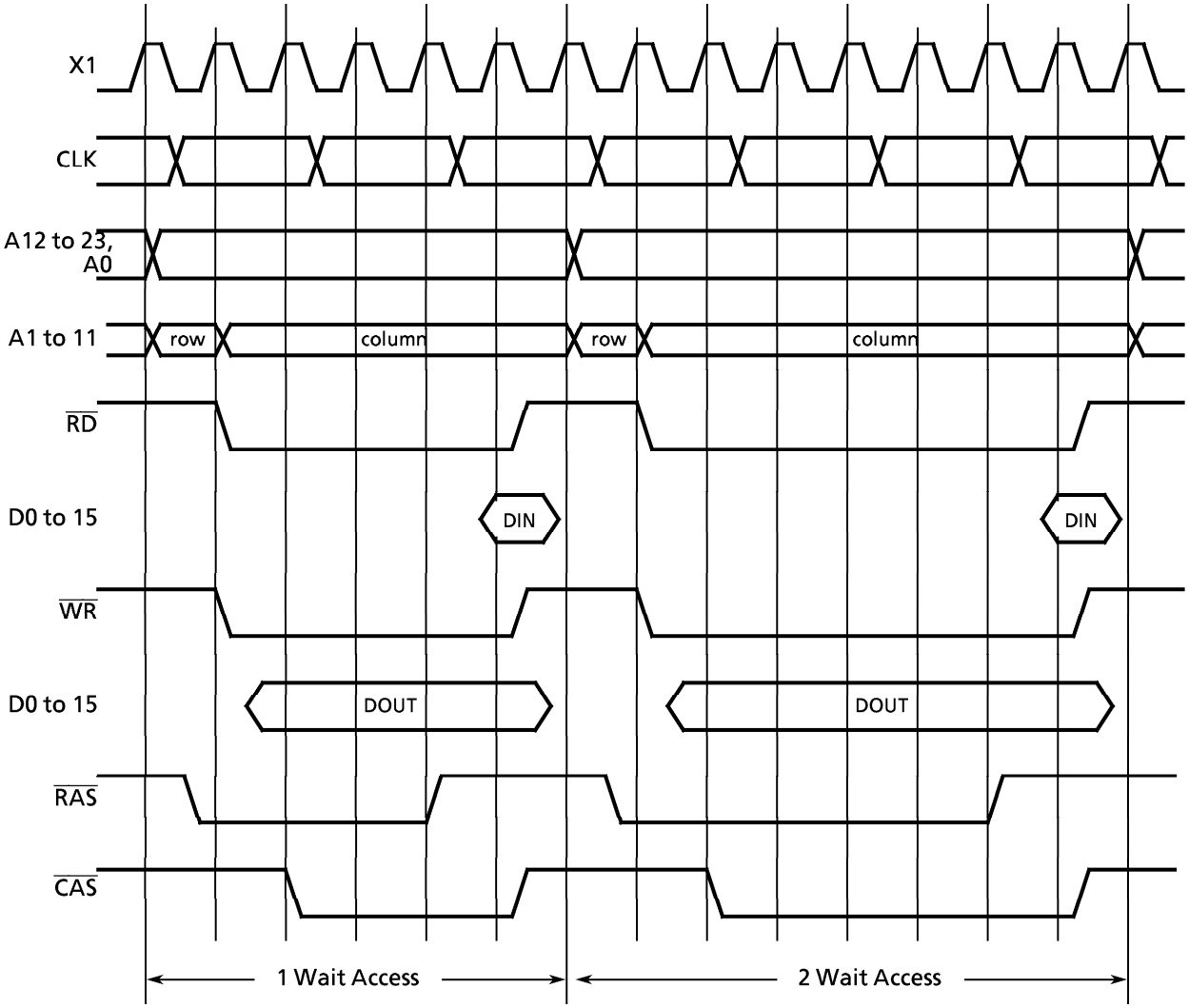


Figure 3.7 (4) DRAM Access Timing (Slow Access Mode)

(2) Refresh controller

The TMP95C061B can output $\overline{RAS}$ / $\overline{CAS}$ used to refresh the DRAM. At the same time the state signal $\overline{REFOUT}$ which indicates a refresh cycle is output. (Only for interval refresh mode.)

DRAM can be refreshed easily because $\overline{RAS}$ / $\overline{CAS}$ / $\overline{REFOUT}$ output frequency and pulse width are programmable.

The refresh controller has the following features.

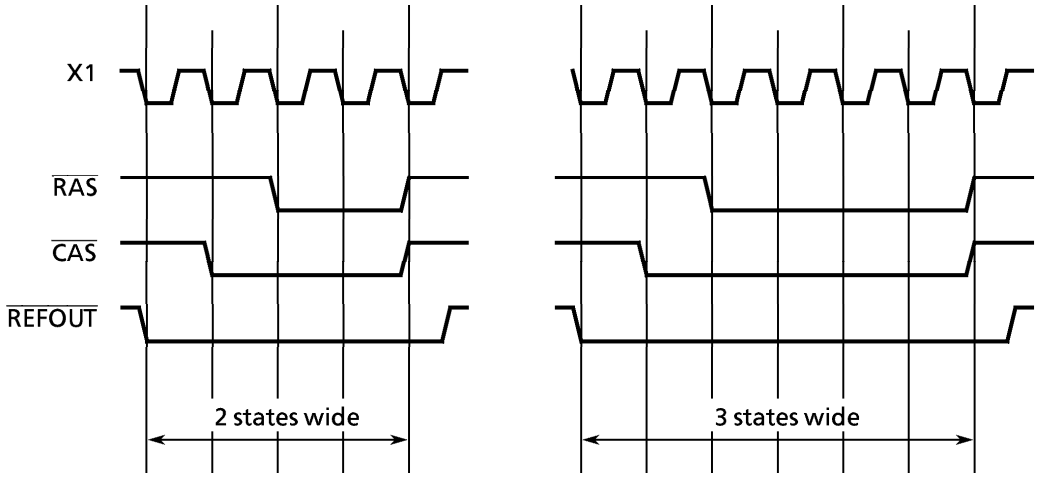
- Refresh mode : $\overline{CAS}$ before $\overline{RAS}$ interval refresh mode
  $\overline{CAS}$ before $\overline{RAS}$ self refresh mode
- Refresh interval : 31 to 195 states (programmable)
- Refresh cycle width : 2 to 9 states (programmable)
- Dummy cycle can be generated
- Refresh cycle is asynchronous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

i) $\overline{CAS}$ before $\overline{RAS}$ interval refresh mode

The refresh interval and refresh width for $\overline{CAS}$ before $\overline{RAS}$ interval refresh mode depends on the DRAM being used.

Therefore, TMP95C061B 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 $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.



a) When the refresh cycle width is set to 2 states

b) When the refresh cycle width is set to 3 states

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<RS2 to 0> according to the system clock being used.

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

Table 3.13 (2) Refresh Cycle Insertion Interval

Refresh Cycle			Insertion Interval (states)	Frequency (f_{OSC})						
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	48.75	39.0	31.2	27.86	24.4	19.5	(15.6)

(Unit : μs)

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

③ Refresh cycle control

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

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

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

However, $\overline{REFOUT}$ is not output. ("1" is output.)

Figure 3.7 (6) shows the self refresh mode timing diagram.

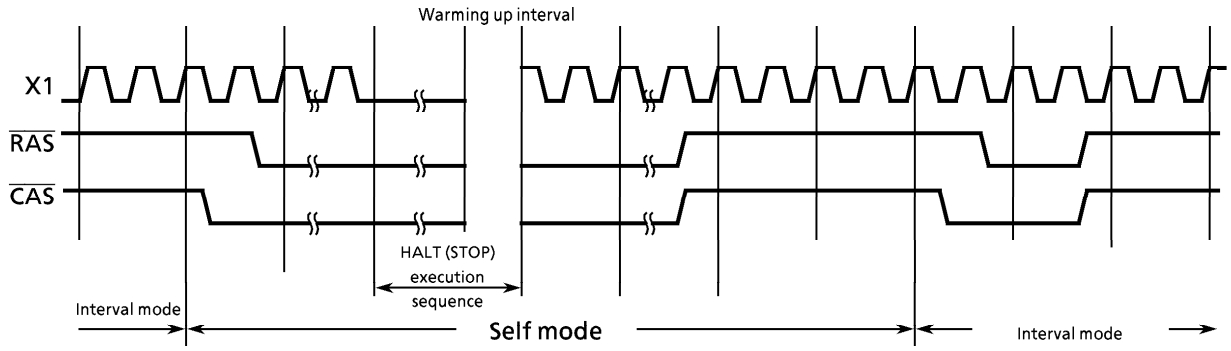


Figure 3.7 (6) Self Refresh Cycle Timing

This mode is executed as follows. First, the settings are made for interval mode. Then B3CS<SRFC> is set to "0" before a HALT instruction to perform one refresh. Then the $\overline{CAS}$ pin and $\overline{RAS}$ 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 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 performed.)

After setting DMEMCR<SRFC> to "0", make sure that the HALT instruction comes after NOP or some instructions.

(3) DRAM initialize

The DRAM controller can generate consecutive $\overline{CAS}$ before $\overline{RAS}$ dummy cycles necessary when using DRAM. This is executed by setting DREFCR<DMI> bit to "1".

Writing 0 to <DMI> (including reset), enabling refresh cycle insert (DREFCR <RC> = 1), or enabling access control (DMEMCE<MAC> = 1) cancels a dummy cycle.

If a dummy cycle is canceled by enabling refresh cycle insert or access control, the <DMI> bit is not zero-cleared.

The dummy cycle width is fixed to 4 states and the interval is fixed to 6 states.

Figure 3.7 (7) shows the $\overline{CAS}$ before $\overline{RAS}$ dummy cycle timing.

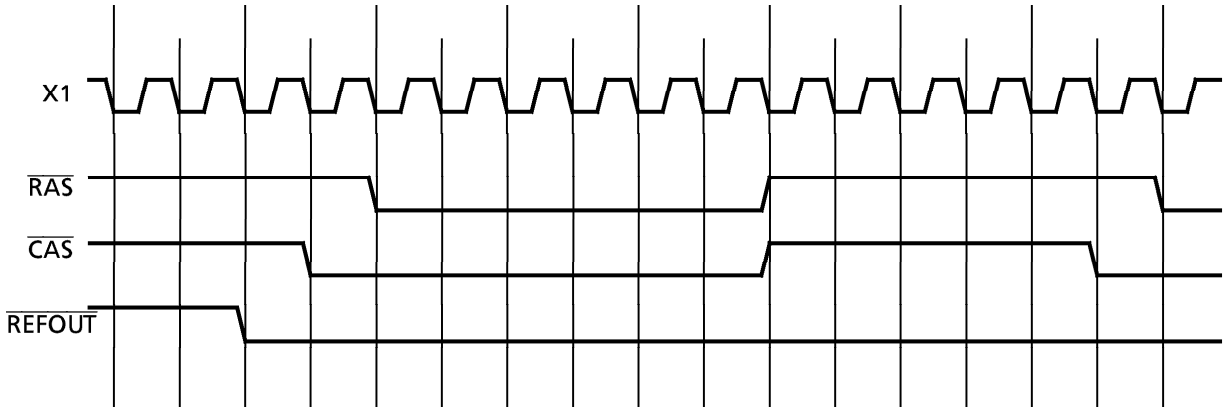


Figure 3.7 (7) $\overline{CAS}$ Before $\overline{RAS}$ Dummy Cycle Timing

(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 TMP95C061B 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 only 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 (DMEMCR<BRM> = 0)

When the bus release request (BUSRQ) pin is set to active (low level), the TMP95C061B acknowledges the bus release request. After the current bus cycle (including DRAM access cycle) ends, the TMP95C061B 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 TMP95C061B 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 used not to release DRAM control dedicated pin (DMEMCR<BRM> = 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 other pins release the bus. Unlike (i), bus release timing is not influenced by a refresh request.

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

(6) Notes

When refresh and access contend, the $\overline{WR}$ and $\overline{HWR}$ pins are set to active and output refresh signals. (Figure 3.7 (8))

① When DRAM is used for executing $\overline{WRITE}$ or $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle test mode at timing (a) in Figure 3.7 (8), a circuit, such as the one shown in 3.7 (7) Connection Example (v), external circuit example (a), is required to avoid test mode. Direct connection is not allowed.

② When DRAM is used to set write-per-bit mode at timing (b) in Figure 3.7 (8), a circuit, such as the one shown in 3.7 (7) Connection Example (vi), external circuit example (b), to avoid this mode is required. Direct connection is not allowed. (TMP95C061B does not support write-per-bit mode.)
For DRAM supporting both test mode in ①. above and write-per-bit mode, use a bypass circuit, such as the one shown in 3.7 (7) Connection Example (vi), external circuit example (b).

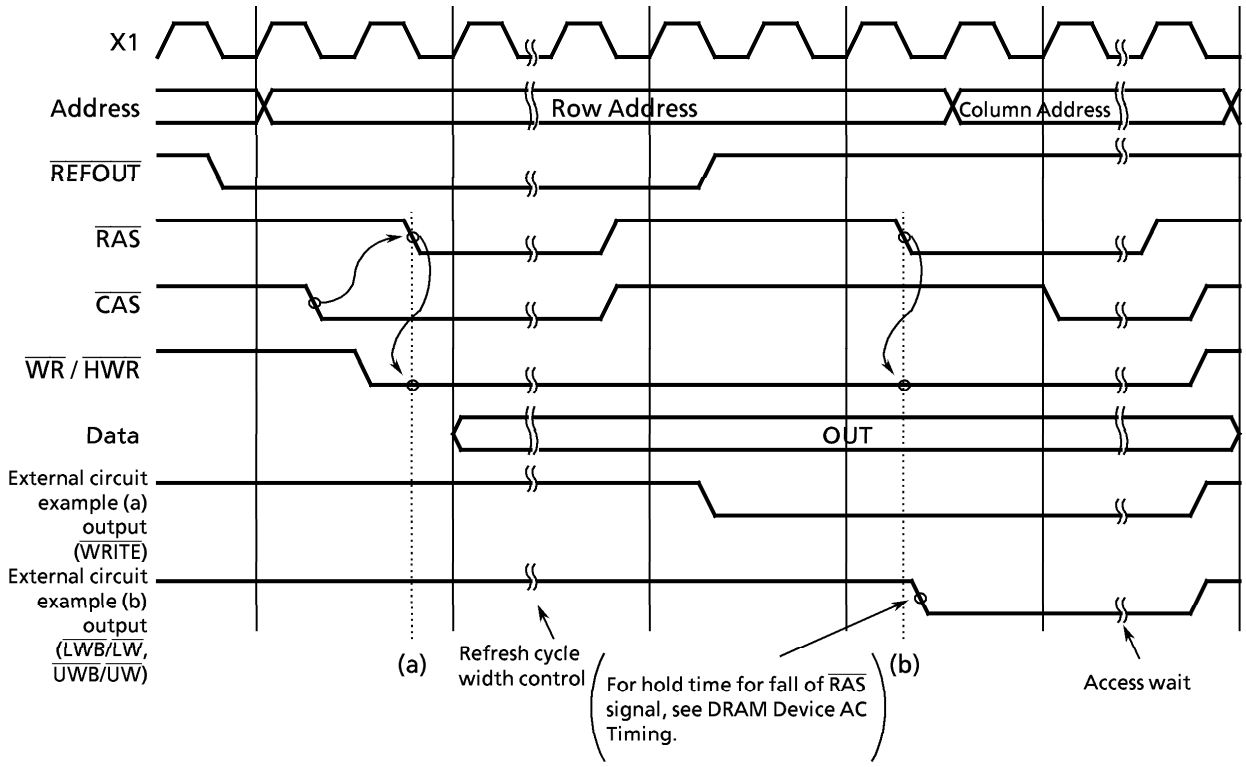
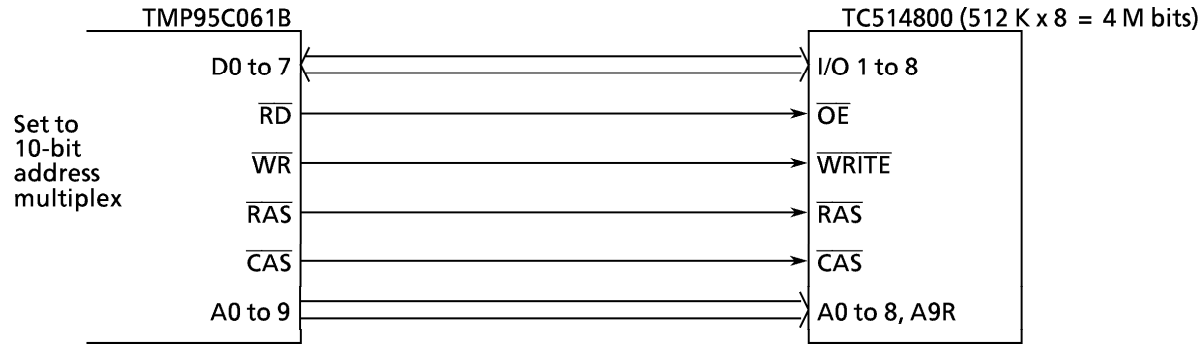


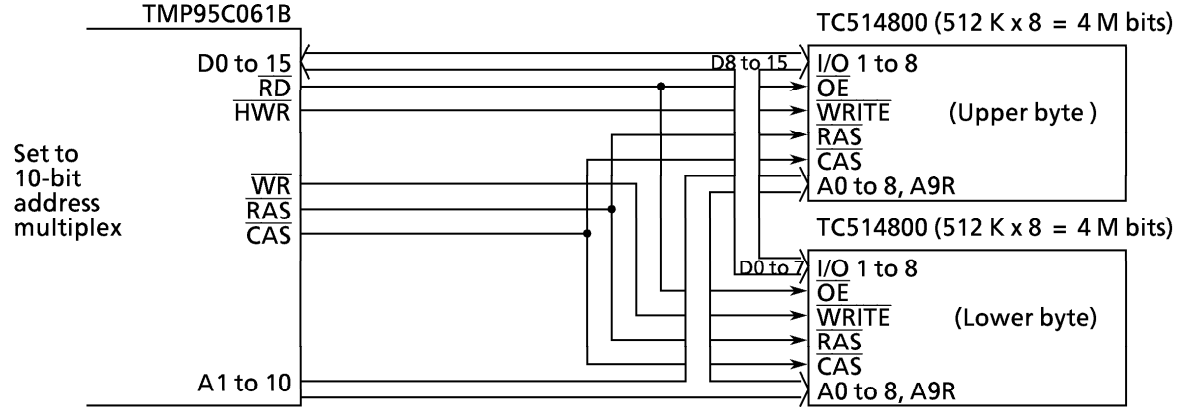
Figure 3.7 (8) Timings for Refresh and Access Contention

(7) Connection Example

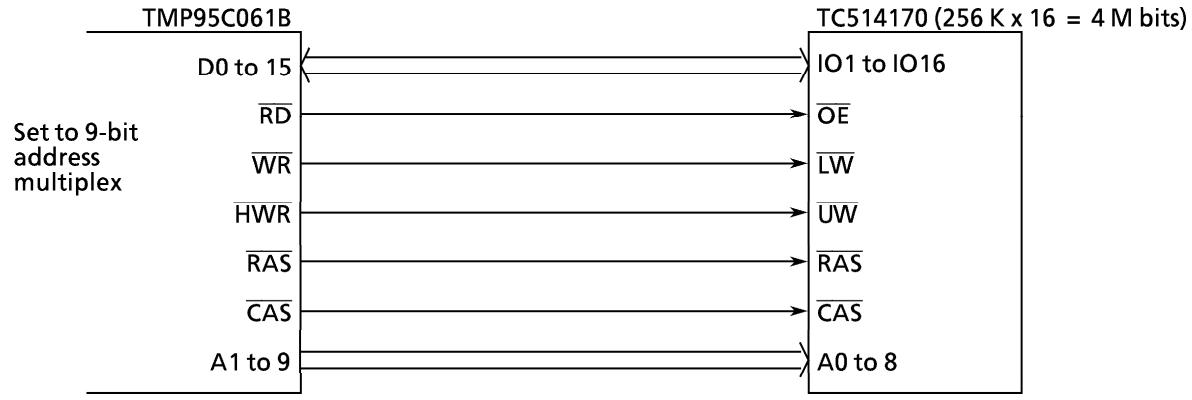
Connection Example (i) 8-bit bus configuration



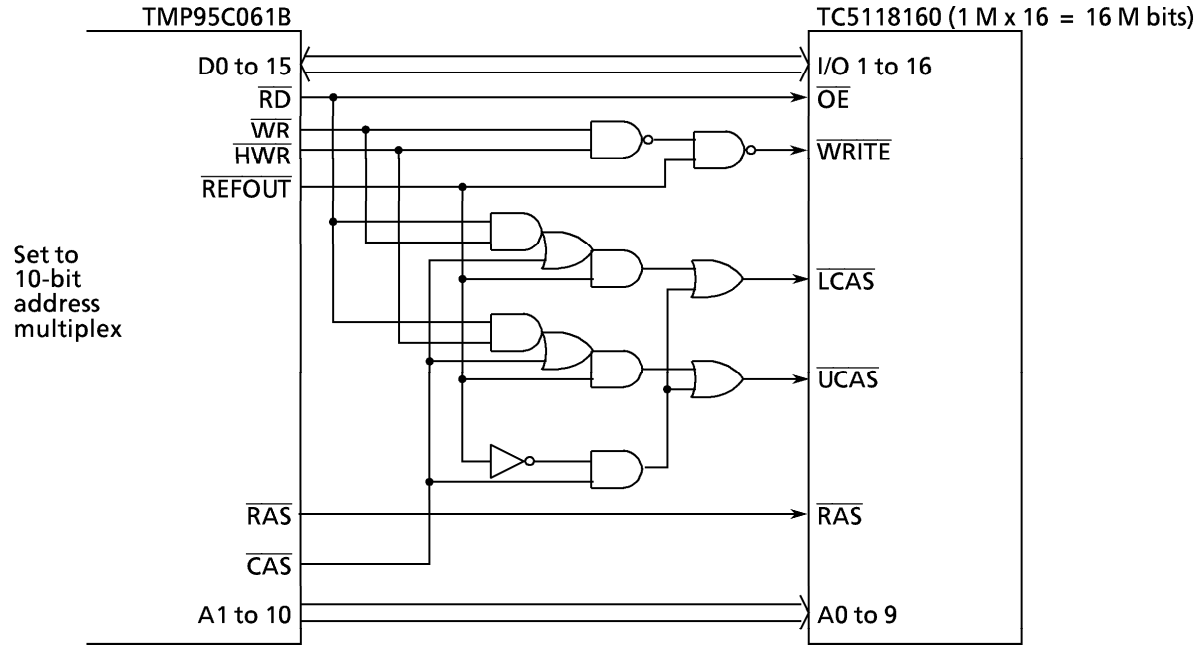
Connection Example (ii) 16-bit bus configuration (8-bit DRAM x 2)



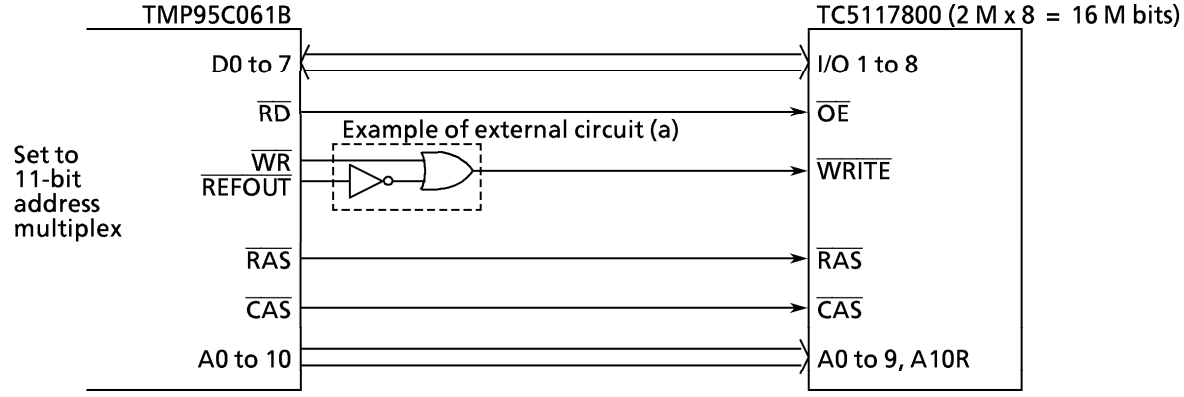
Connection Example (iii) 16-bit bus configuration (2WE mode)



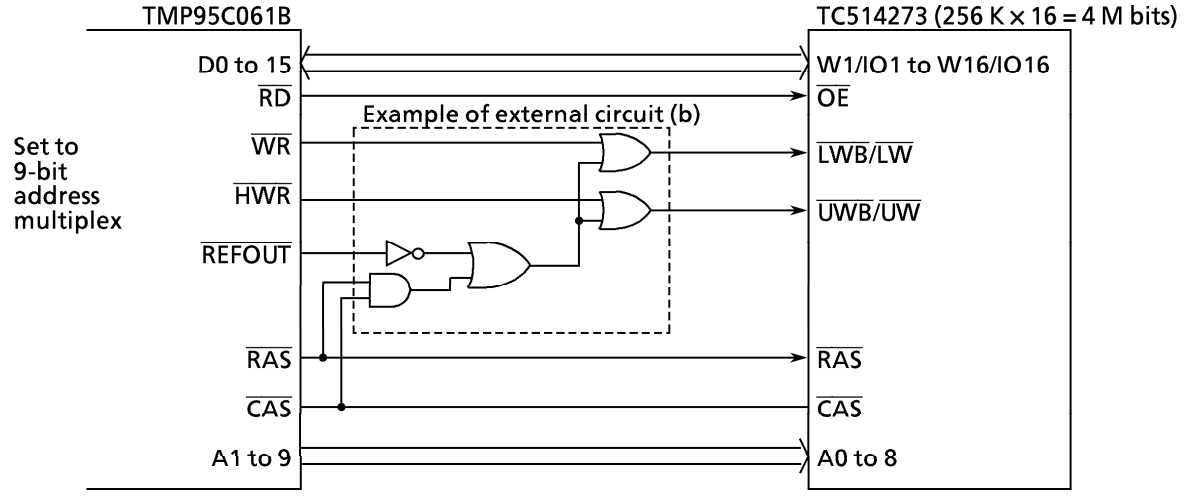
Connection Example (iv) 16-bit bus configuration (2CAS mode)



Connection Example (v) Connection to DRAM supporting $\overline{\text{WRITE}}$ or $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh test mode



Connection Example (vi) Connection to DRAM supporting write-per-bit mode or DRAM supporting both write-per-bit mode and $\overline{\text{WRITE}}$ or $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh test mode.



3.8 8-bit Timers

TMP95C061B 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 timer. The following four operating modes are provided for the 8-bit timers.

- 8-bit interval timer mode (4 timers)
- 16-bit interval timer mode (2 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 with constant cycle) output mode (2 timers)

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

Timer 2 / 3 have the same configuration of circuit as Timer 0 / 1. The difference between Timer 0 and Timer 2 is that Timer 0 has external clock input pin (TI0), while Timer 2 has none.

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

Among the input clock sources for the interval timers, the internal clocks of ϕT1, ϕT4, ϕT16, and ϕT256 are obtained from the 9-bit prescaler shown in Figure 3.8 (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.

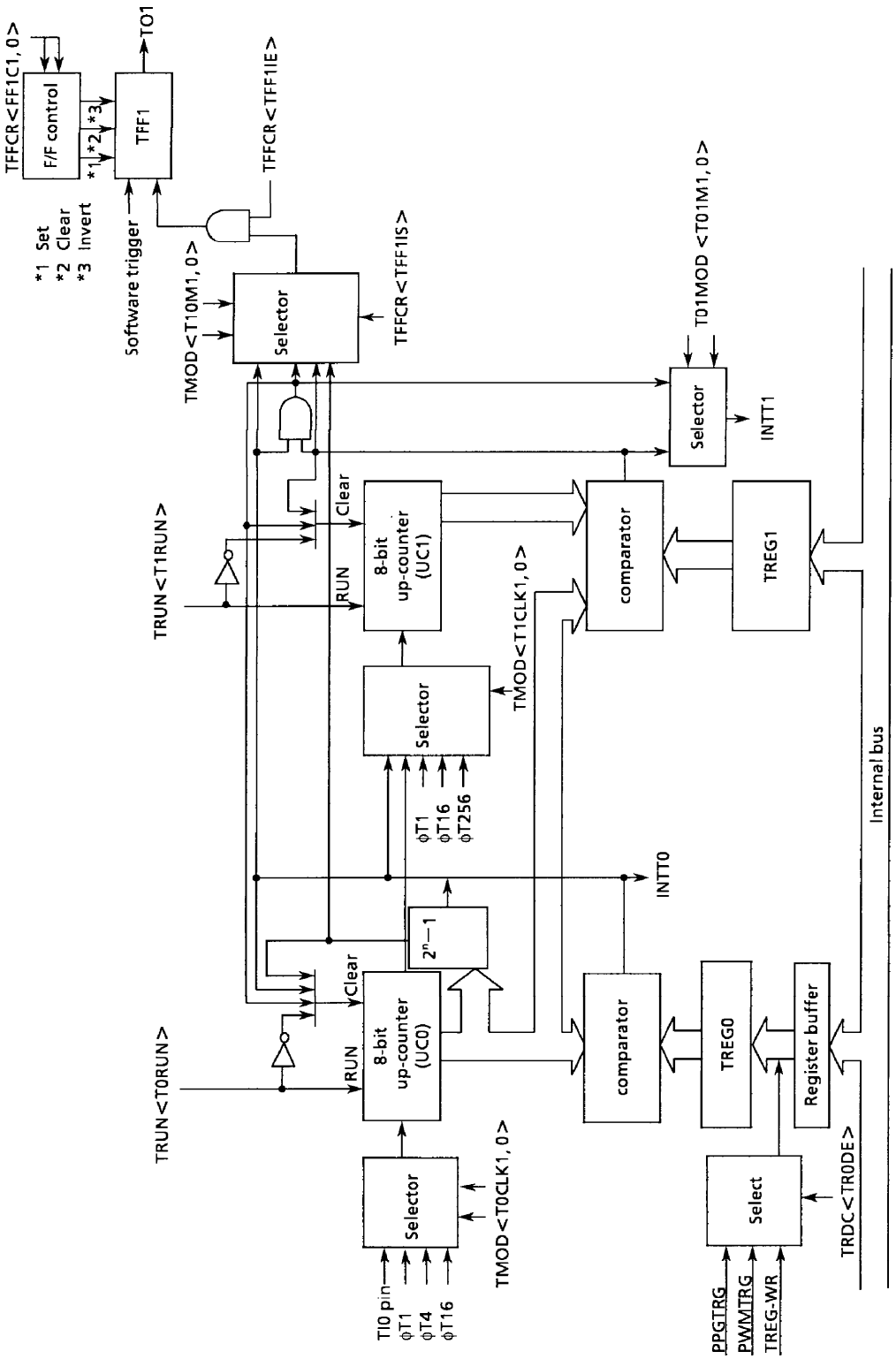


Figure 3.8 (1) Block Diagram of 8-bit Timers (Timers 0 and 1)

① Prescaler

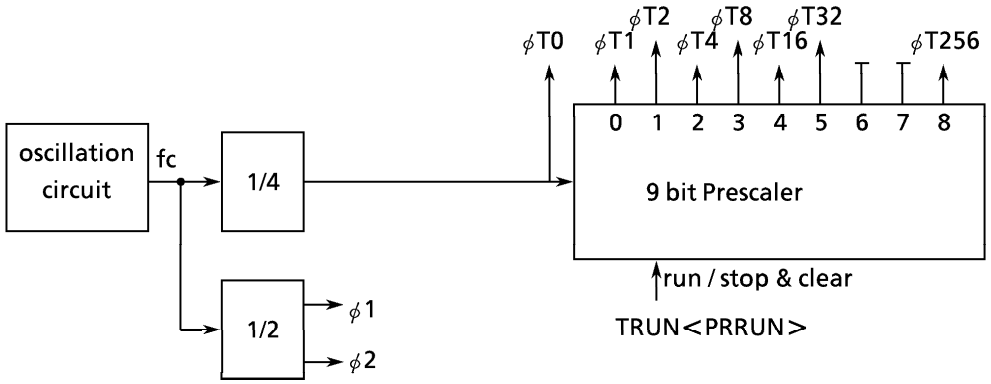
These are 9 bit prescaler and prescaler clock selection register to generate input clock for 8 bit Timer 0 / 1, Timer 4 / 5 and Serial Interface 0 / 1.

The 8 bit Timer 0, 1uses 4 types of clock : ϕT1, ϕT4, ϕT16, and ϕT256 among the prescaler output.

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

Cycle

Clock \ fc	25 MHz
ϕT1 (8/fc)	320 ns
ϕT4 (32/fc)	1.28 μs
ϕT16 (128/fc)	5.12 μs
ϕT256 (2048/fc)	81.92 μs



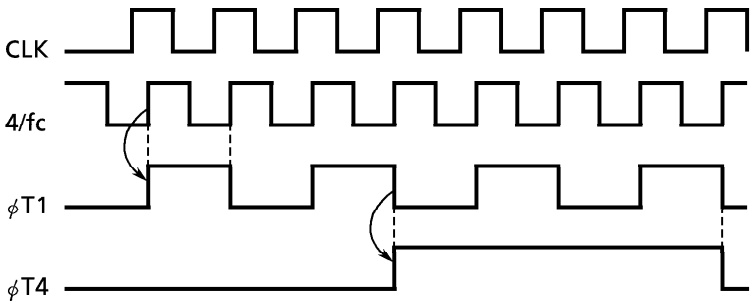


Figure 3.8 (2) Prescaler

② Up-counter

These are an 8 bit binary counter which counts up by the input clock pulse specified by 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 ϕT1, ϕT4, and ϕT16 and the external clock input (TI0 : 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 16 bit mode, the input clock is selected from the internal clocks ϕT1, ϕT16 and ϕT256, and the output of 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, T01MOD3, 2=01, ϕT1 becomes the input of timer 1 (8 bit timer mode).

Operation mode is also set by T01MOD register and T23MOD 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 & 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 controll 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 enable state, the data is transfered from the register buffer to the timer register when the 2^n-1 overflow occurs in PWM mode, or at the PPG cycle in PPG mode.

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 <TRDDE>/<TR2DE> to 1, and write the following data in the register buffer.

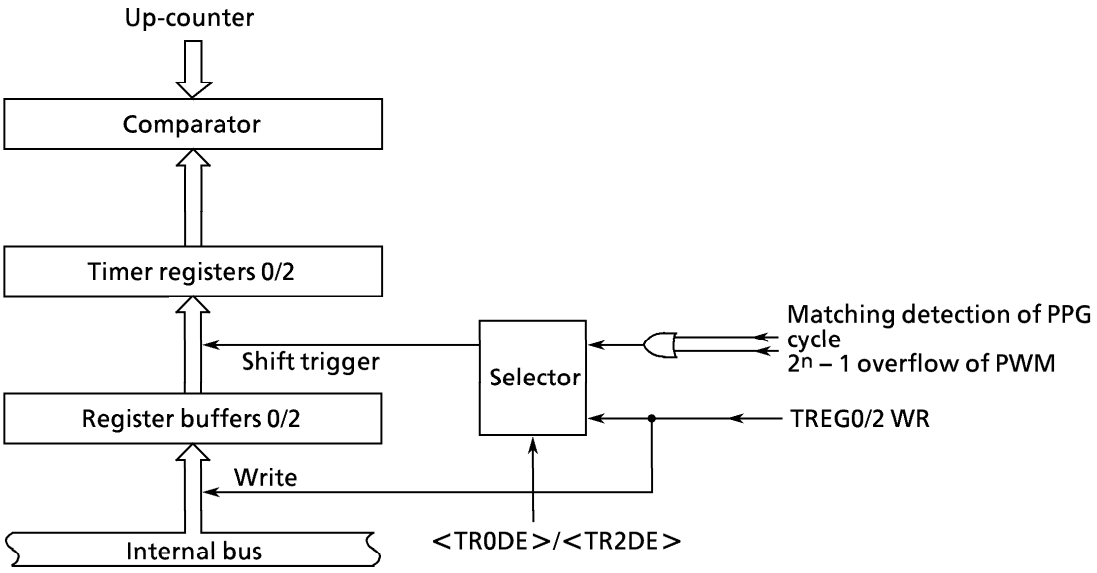


Figure 3.8 (3) Configuration of Timer Register 0/2

Note : Timer register and the register buffer are allocated to the same memory address. When < TR0DE>/<TR2DE>=0, the same value is written in the register buffer as well as the timer register, while when < TR0DE>/<TR2DE>=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.

T01MOD (0024H)

	7	6	5	4	3	2	1	0
bit Symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: 2^6-1 10: 2^7-1 11: 2^8-1	PWM cycle	00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 0

00	External clock Input TI0
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Input clock of timer 1

	T01MOD7, 6 ≠ 01	T0/MOD7, 6 = 01 <T10M1,0> = 01
00	Comparator output of timer 0	Overflow output of timer 0 (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

Select PWM cycle
(Don't care except PWM mode)

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

Set the operation mode of timer 0 and 1.

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

Figure 3.8 (4) Timer0/1 Mode control Register (T01MOD)

T23MOD (0028H)

	7	6	5	4	3	2	1	0
bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: $2^6 - 1$ PWM 10: $2^7 - 1$ Cycle 11: $2^8 - 1$		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: – 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 2

00	——
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Input clock of timer 3

	T23MOD7, 6 ≠ 01	T23MOD7, 6 = 01
00	Comparator output of timer 2	Overflow output of Timer 2 (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

Select PWM cycle
(Don't Care except PWM mode)

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

Set operation mode of timer 2, 3

00	Two 8-bit timers (timer 2, 3)
01	16-bit timer
10	8-bit PPG output
11	8-bit PWM output (timer 2) + 8-bit timer (timer 3)

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

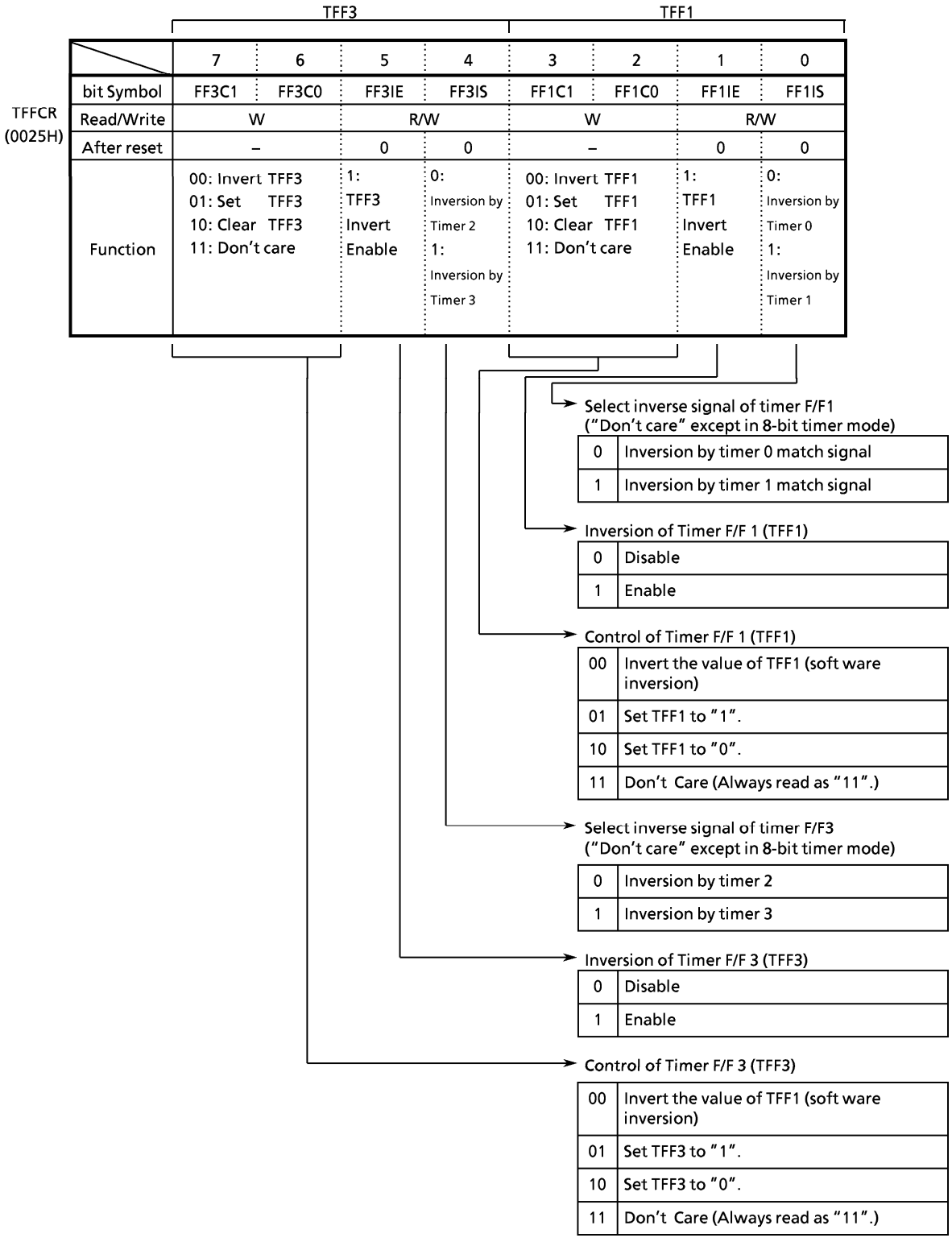


Figure 3.8 (6) 8-bit Timer Flip-flop Control Register (TFFCR)

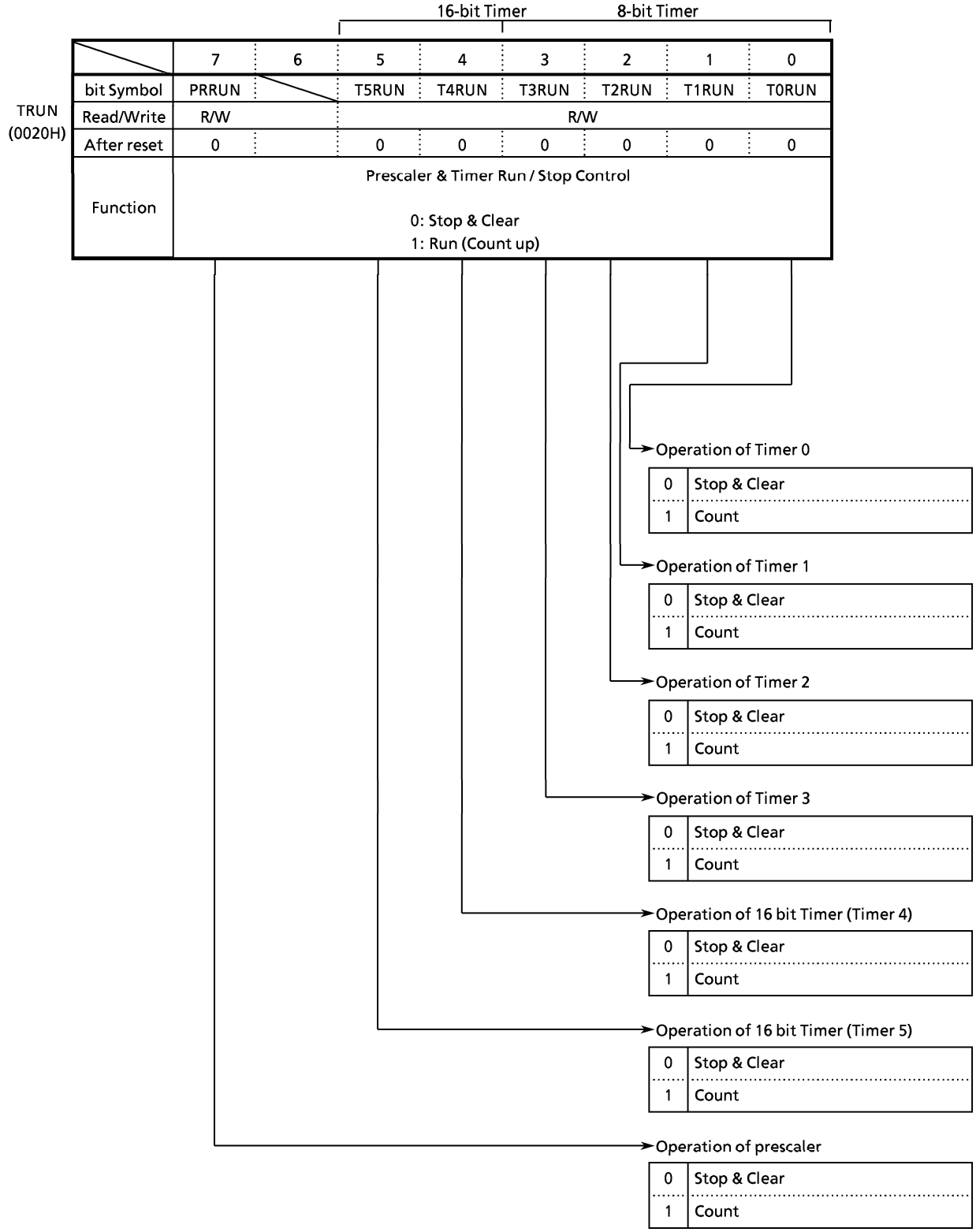


Figure 3.8 (7) Timer Operation Control Register (TRUN)

TRDC (0029H)

	7	6	5	4	3	2	1	0
bit Symbol							TR2DE	TR0DE
Read/Write							R/W	
After reset							0	0
Function							0: Double Buffer Disable 1: Double Buffer Enable	

Operation of Timer Register 0 Double buffer

0	Disable
1	Enable

Operation of Timer Register 2 Double buffer

0	Disable
1	Enable

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

① Generating interrupts in a fixed cycle

To generate timer 1 interrupt at constant intervals using timer 1 (INTT1), first stop timer 1 then set the operation mode, input clock, and a cycle to T01MOD and TREG1 register, respectively. Then, enable interrupt INTT1 and start the counting of timer 1.

Example : To generate timer 1 interrupt every 32 μs at fc=25 MHz, 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 φT1 (0.32 μs @ fc = 25 MHz)
                                as the input clock.
TREG1     ← 0 1 1 0 0 1 0 0     Set the timer register 32 μs ÷ φT1 = 100 = 64H
INTET01   ← 1 1 0 1 - - - -     Enable INTT1, and set it to "Level 5".
TRUN      ← 1 X - - - - 1 -     Start timer 1 counting.
```

Note : X; Don't care –; No change

Use the table 3.8 (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 fc = 25 MHz)	resolution
ϕT1 (8/fc)	0.32 μs to 81.92 μs	0.32 μs
ϕT4 (32/fc)	1.28 μs to 327.7 μs	1.28 μs
ϕT16 (128/fc)	5.12 μs to 1.311 ms	5.12 μs
ϕT256 (2048/fc)	81.92 μs to 20.97 ms	81.92 μs

② Generating a 50 % duty square wave pulse

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

Example : To output a 1.92 μs square wave pulse through TO1 pin at fc=25 MHz, set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

```
          7 6 5 4 3 2 1 0
TRUN   ← - X - - - - 0 -      Stop timer 1, and clear it to "0".
TO1MOD← 0 0 X X 0 1 - -       Set the 8-bit timer mode, and select φT1 as the input clock.
                              Set the timer register at 1.92 μs ÷ φT1 ÷ 2 = 3.
TREG1  ← 0 0 0 0 0 0 1 1      Clear TFF1 to "0", and set to invert by the match detect signal
TFFCR  ← - - - - 1 0 1 1      from timer 1.

PACR   ← X X X X - 1 - -      Select PA2 as TO1 pin.
PAFC   ← X X X X - 1 X X      Start timer 1 counting.
TRUN   ← 1 X - - - - 1 -
```

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

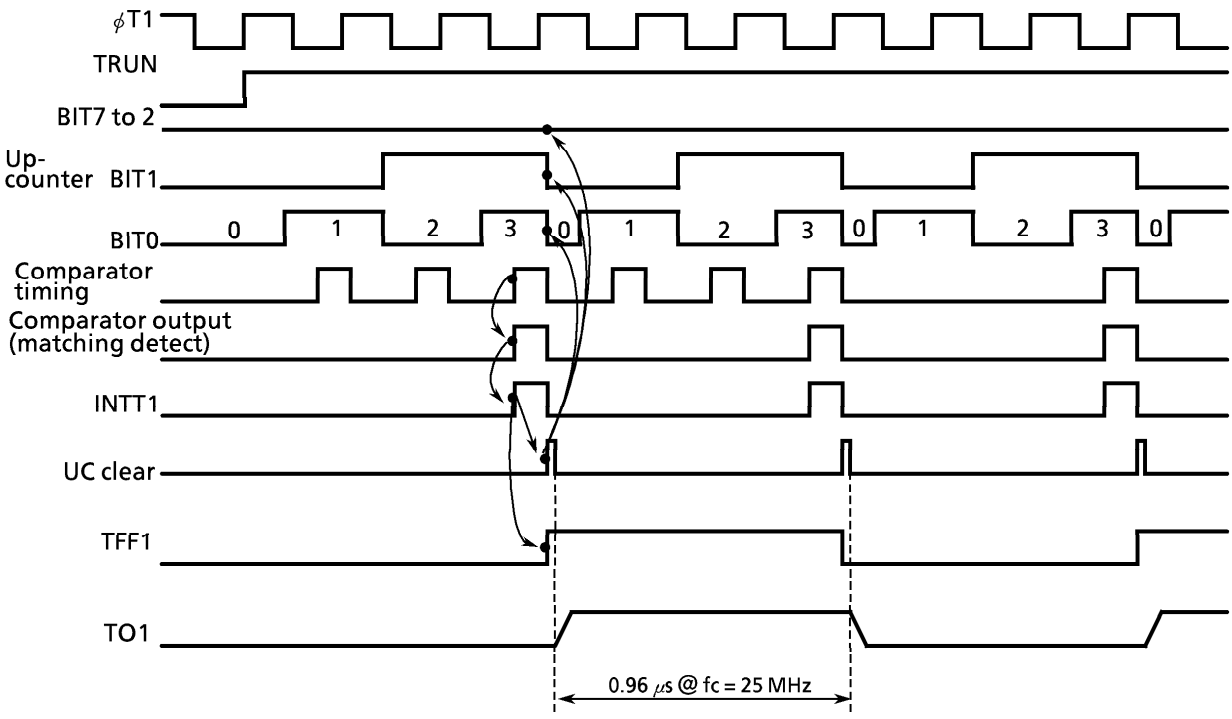


Figure 3.8 (9) Square Wave (50 % Duty) Output Timing Chart

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.

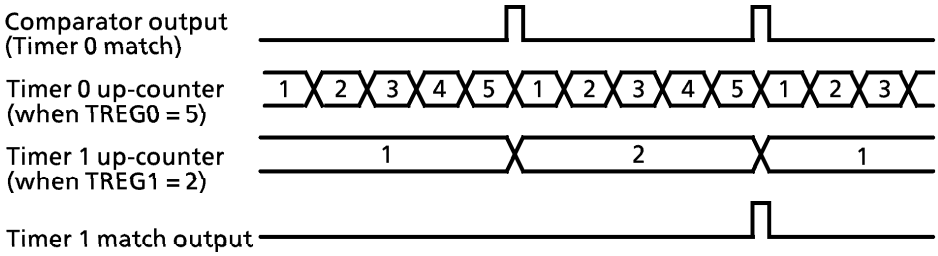


Figure 3.8 (10) Timer 1 count up by timer 0

④ Output inversion with software

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

Writing 00 to TFFCR<FF1C1, 0> inverts the value of TFF1. Writing 00 to 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<TFF1C1,0> to set TFF1 to "1".

Note: The value of timer register and tiner Flip-flop cannot be read.

(2) 16-bit timer mode

A 16-bit interval timer is configured 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 connecting 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, and the comparator is restarted by writing data into TREG1.)

Table 3.8 (2) The interrupt period and input clock in 16 bit timer mode

Input clock	Interrupt period (fc = 25 MHz)	resolution
ϕT1 (8/fc)	0.32 μs to 20.971 ms	0.32 μs
ϕT4 (32/fc)	1.28 μs to 83.885 ms	1.28 μs
ϕT16 (128/fc)	5.12 μs to 335.539 ms	5.12 μs

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

When counting with input clock of ϕT16 (5.12 μs @ 25 MHz)

$0.32\ \text{s} \div 5.12\ \mu\text{s} = 62500 = \text{F424H}$

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 cleared. And then the interrupt INTT0 is not generated.

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 (Input overflow of Timer 0 to Timer 1)	not generate the interrupt	output disable	TREG0 (Continued to count up after match)	generate the interrupt	output enable	TREG1$*2^8$ + TREG0
8 bit Timer mode (input match of Timer 0 to Timer 1)	generate the interrupt	output enable (Timer 0 or Timer 1)	TREG0 (Cleared after match)	generate the interrupt	output enable (Timer 0 or Timer 1)	TREG1$*$ TREG0

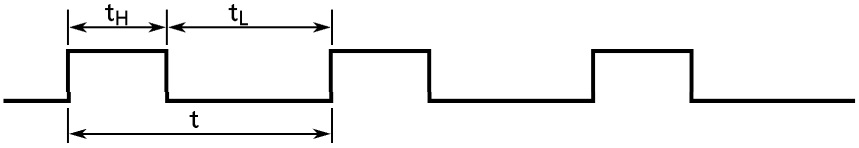
Example : When TREG1 = 04H and TREG0 = 80H

Figure 3.8 (11) Timer output by 16-bit timer mode

(3) 8-bit PPG (Programmable Pulse Generation) Output 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 TO3 (also used as PA3).



Here shows the operation of Timer 0. Timer 2 provides the same operation as Timer 1.

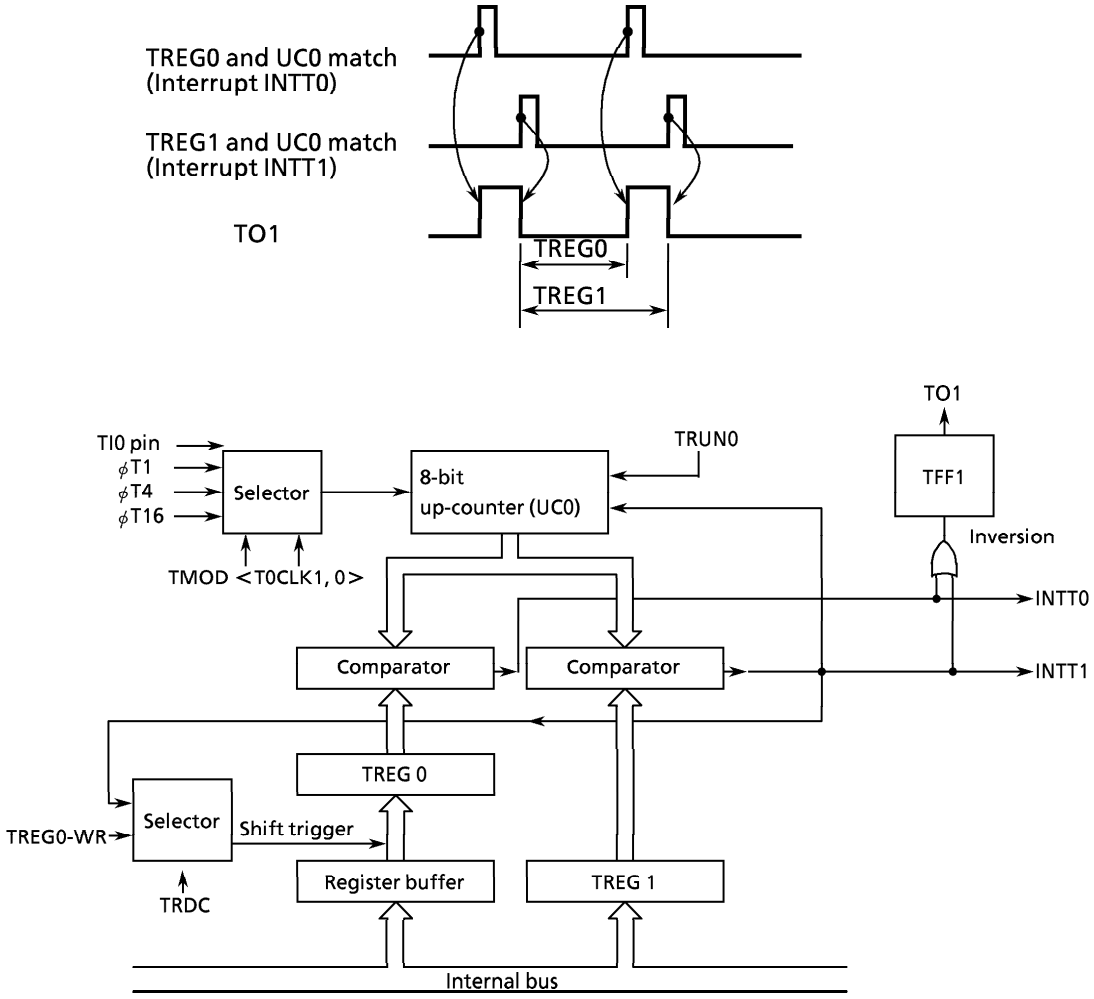
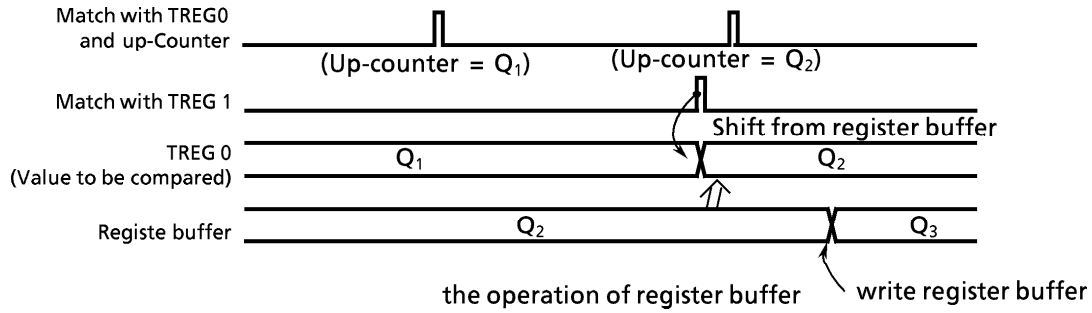


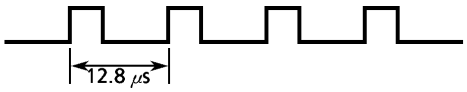
Figure 3.8 (12) Block Diagram of 8-Bit PPG Output Mode

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

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



Example : Generating 1/4 duty 78.125 kHz pulse (at fc = 25 MHz)



- Calculate the value to be set for timer register.
  To obtain the frequency 78.125 kHz, the pulse cycle t should be : t = 1/78.125 kHz = 12.8 μs.
  Given φT1 = 0.32 μs (at 25 MHz),
  12.8 μs ÷ 0.32 μs = 40
  Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H
  and then duty to 1/4, t × 1/4 = 12.8 μs × 1/4 = 3.2 μs
  3.2 μs ÷ 0.32 μs = 10
  Therefore, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

	MSB 7	6	5	4	3	2	1	LSB 0	
←									
TRUN ←	0	X	-	-	-	-	0	0	Stop timer 0 and timer 1 and clear it to "0".
T01MOD ←	1	0	X	X	0	1	0	1	Set the 8-bit PPG mode, and select φT1 as input clock.
TFFCR ←	-	-	-	-	0	1	1	x	Sets TFF 1 and enable the inversion and double buffer enable.
									Writting "10" provides negative logic pulse.
TREG0 ←	0	0	0	0	1	0	1	0	Write "0AH".
TREG1 ←	0	0	1	0	1	0	0	0	Write "28H".
PACR ←	X	X	X	X	-	1	-	-	Set PA2 as the T01 pin.
PAFC ←	X	X	X	X	-	1	X	X	
TRUN ←	1	X	-	-	-	-	1	1	Start timer 0 and Timer 1 counting.

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

(4) 8-bit PWM Output mode (Pulse Width Modulation)

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.

Here shows the PWM mode operation of Timer 0. Timer 2 provide the same operation as Timer 0.

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

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

(Set value of timer register) < (Set value of 2^n-1 counter overflow)
(Set value of timer register) ≠ 0

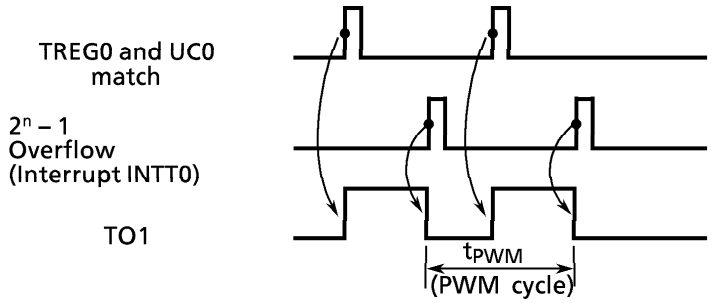


Figure 3.8 (13) shows the block diagram of this mode.

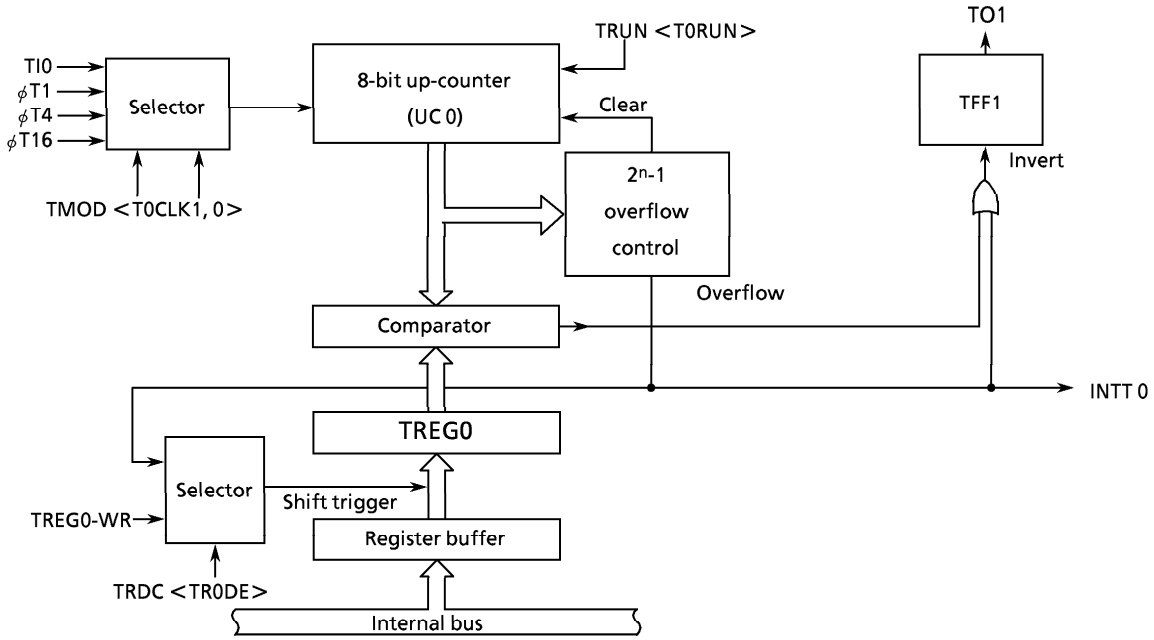
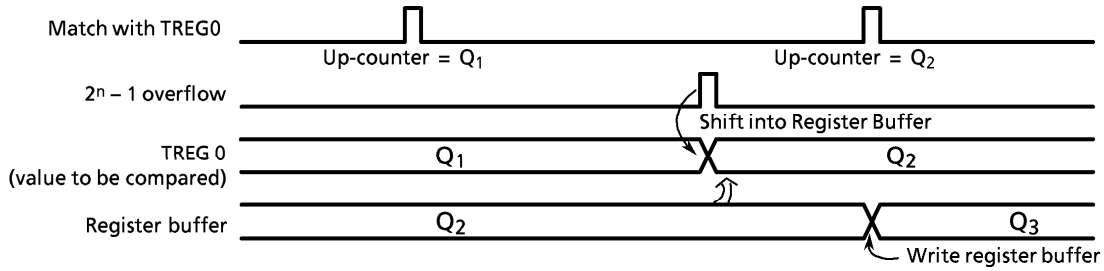


Figure 3.8 (13) Block Diagram of 8-Bit PWM Mode

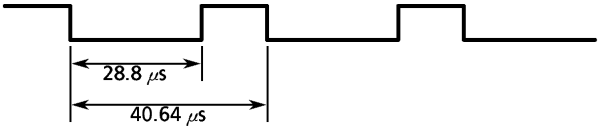
In this mode, the value of register buffer will be shifted in TREG0 if 2^n-1 overflow is detected when the double buffer of TREG0 is enabled.

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



The operation of register buffer

Example : To output the following PWM waves to TO1 pin at fc=25 MHz.



To realize 40.64 μs of PWM cycle by ϕT1=0.32 μs (at fc=25 MHz),

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

Consequently, n should be set to 7.

As the period of low level is 28.8 μs, for ϕT1=0.32 μs,

set the following value for TREG0.

$28.8\ \mu s \div 0.32\ \mu s = 90 = 5AH$

	MSB 7 6 5 4 3 2 1 LSB 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 ϕT1 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 setting of $2^n - 1$ counter

	PWM cycle (@ fc = 25 MHz)		
	ϕT1	ϕT4	ϕT16
$2^6 - 1$	20.2 μs (49.6 kHz)	80.6 μs (12.4 kHz)	322.6 μs (3.1 kHz)
$2^7 - 1$	40.6 μs (24.6 kHz)	162.6 μs (6.2 kHz)	650.2 μs (1.5 kHz)
$2^8 - 1$	81.6 μs (12.3 kHz)	326.4 μs (3.1 kHz)	1.31 ms (0.8 kHz)

(5) Table 3.8 (4) shows the settings for all 8-bit timer modes.

Table 3.8 (4) Selection of 8 bit timer mode and control register

Timer Mode (8 bit timer × 2ch)	Mode T01M (T23M)	PWM0 (PWM2)	Upper clock input T1CLK (T3CLK)	Lower clock input T0CLK (T2CLK)	Selection of Inversion FF1IS (FF3IS)
16 bit timer (16 bit) × 1ch	01	–	–	(External clock, ϕT1, 4, 16)	–
8 bit timer (Input of upper timer is output of power one)	00	–	00	(External clock, ϕT1, 4, 16)	0: Lower timer 1: Upper timer
8 bit timer × 2ch	00	–	(ϕT1, 16, 256)	(External clock, ϕT1, 4, 16)	0: Lower timer 1: Upper timer
8 bit PPG × 1ch	10	–	–	(External clock, ϕT1, 4, 16)	–
8 bit PWM × 1ch (Lower) 8 bit timer × 1ch (Upper)	11	PWM cycle	(ϕT1, 16, 256)	(External clock, ϕT1, 4, 16)	–

3.9 16-bit Timer

TMP95C061B contains two (timer 4 and timer 5) multifunctional 16-bit timer / event counter with the following operation modes.

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

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

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

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

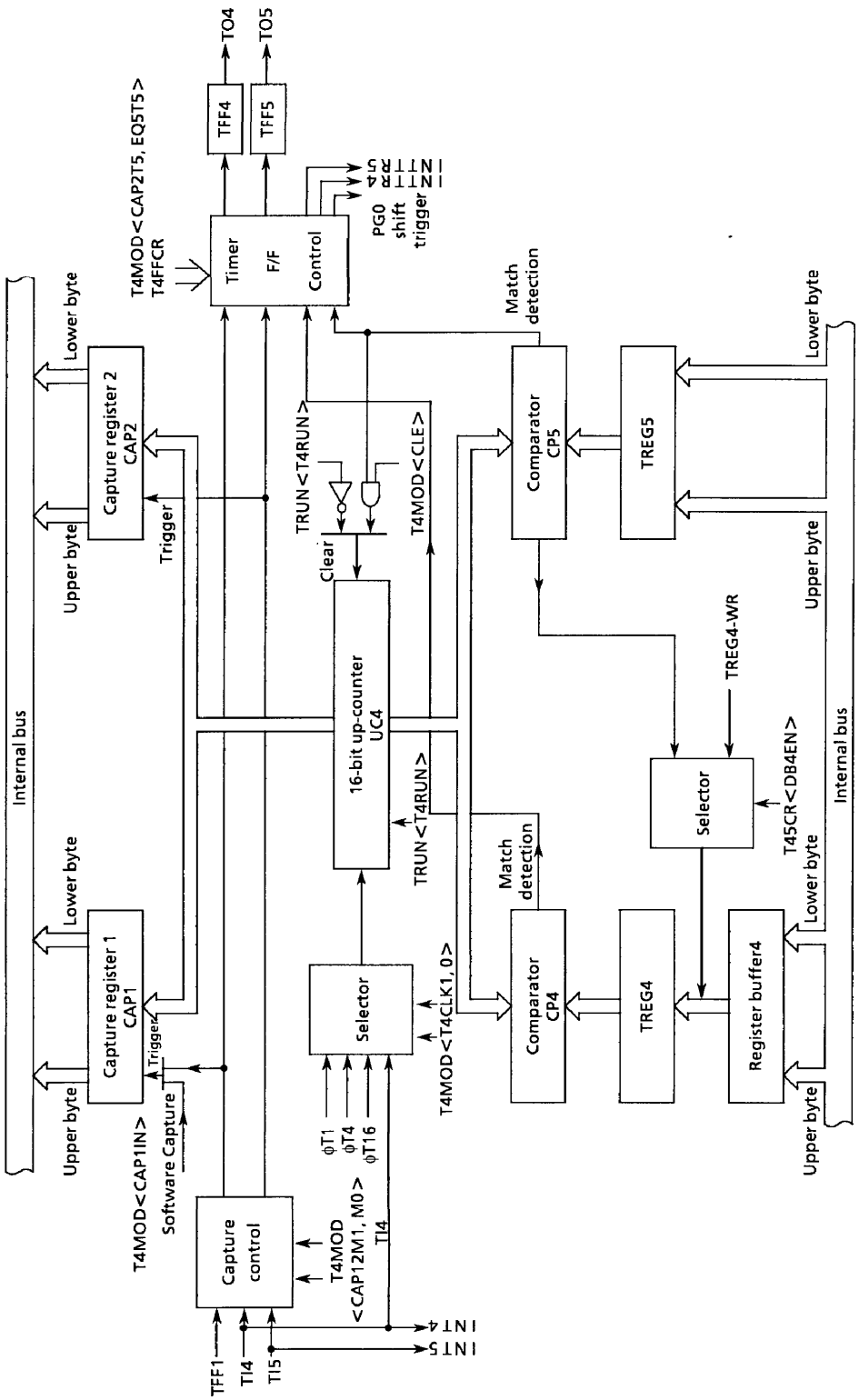


Figure 3.9 (1) Block Diagram of 16-Bit Timer (Timer 4)

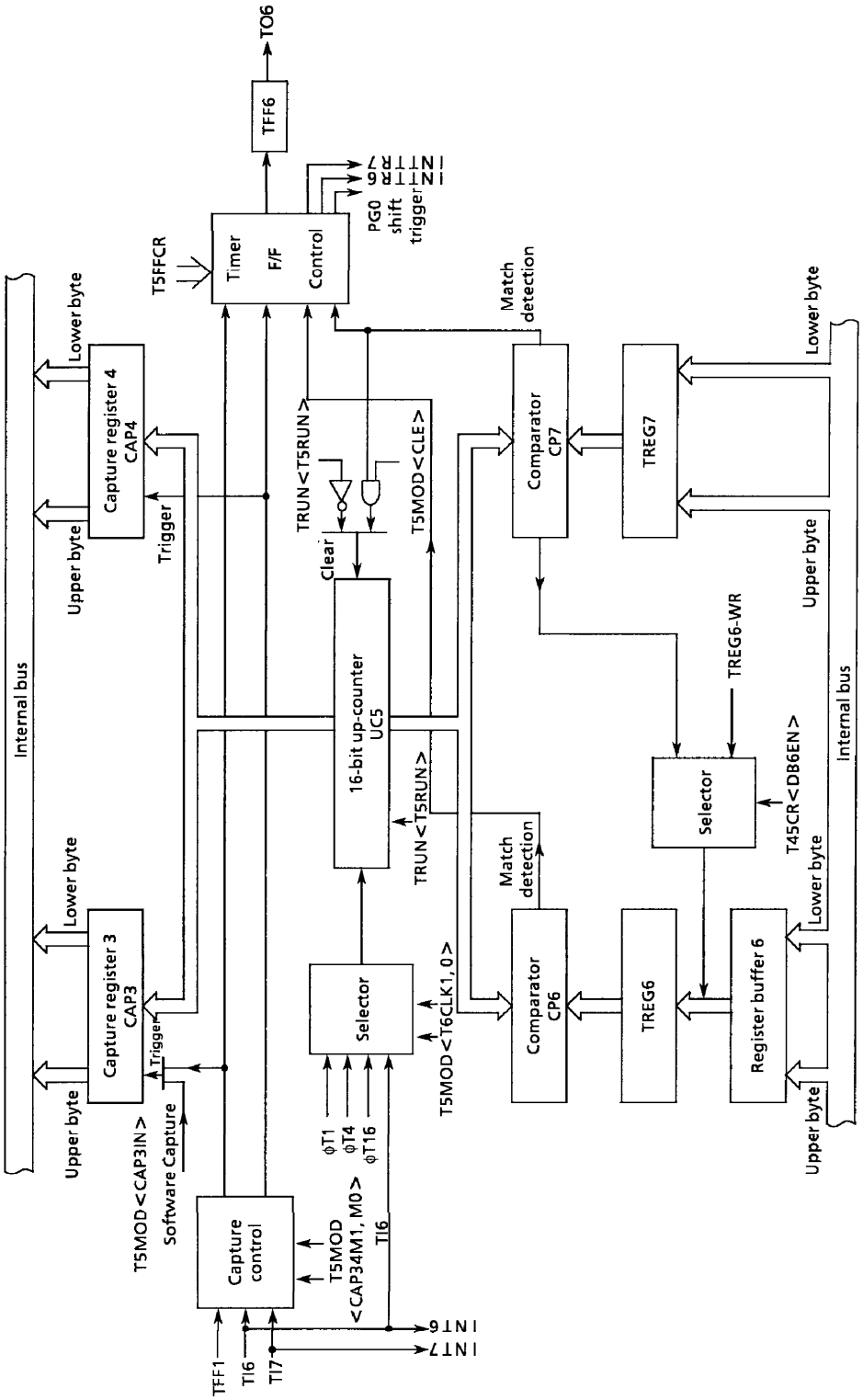


Figure 3.9 (2) Block Diagram of 16-Bit Timer (Timer 5)

T4MOD (0038H)

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

Timer 4 input clock

00	External clock (TI4)
01	ϕT1 (8 / fc)
10	ϕT4 (32 / fc)
11	ϕT16 (128 / fc)

Clearing the up-counter UC4

0	Clear disable
1	Clear by match with TREG5.

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

T4MOD (0038H)

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

Capture timing of timer4

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

Software capture

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

Timer flip-flop 5 (TFF5) invert trigger

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

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

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

T4FFCR (0039H)

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

Timer flip-flop 4 (TFF4) control

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

Timer flip-flop 4 (TFF4) invert trigger

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

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

Timer flip-flop 5 (TFF5) control

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

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

T5MOD (0048H)

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

Timer 5 input clock

00	External clock (TI4)
01	ϕT1 (8 / fc)
10	ϕT4 (32 / fc)
11	ϕT16 (128 / fc)

Clearing the up-counter UC5

0	Clear disable
1	Clear by match with TREG7

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

T5MOD (0048H)

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

Timer 5 Capture timing

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

Software capture

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

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

T5FFCR (0049H)

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

Timer flip-flop 6 (TFF6) control

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

Timer flip-flop 6 (TFF6) invert trigger

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

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

Figure 3.9 (8) 16-Bit Timer5 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
Function	Write "0" ※ Always read as "0"				PG1 shift Trigger 0: 8 bit Timer Trigger 3 (Timer 2, 3) 1: 16 bit Timer Trigger (Timer 5)	PG0 shift Trigger 0: 8 bit Timer Trigger 1 (Timer 0, 1) 1: 16 bit Timer Trigger (Timer 4)	0: Disable 1: Enable	
							Double buffer of TREG6	Double buffer of TREG4

Double buffer countrol

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		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler & Timer Run / Stop CONTROL 0: Stop & Clear 1: Run (Count up)							

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

UC4 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 ϕT1, ϕT4, and ϕT16 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 & clear of the counter is controlled by timer operation control register TRUN<T4RUN>, <T5RUN>.

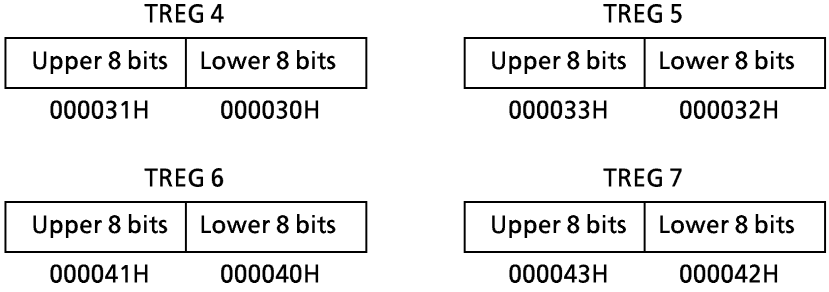
When clearing is enabled, up-counter UC4/UC5 will be cleared to zero each time it coincides matches the timer register TREG5, TREG7. The "clear enable/disable" is set by T4MOD<CLE> and T5MOD<CLE>.

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 interval time. 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 date load instruction or using 1 byte date load instruction twice for lower 8 bits and upper 1 bits in order.



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 <DB4E, DB6EN> = 1, and then write the following data in the register buffer.

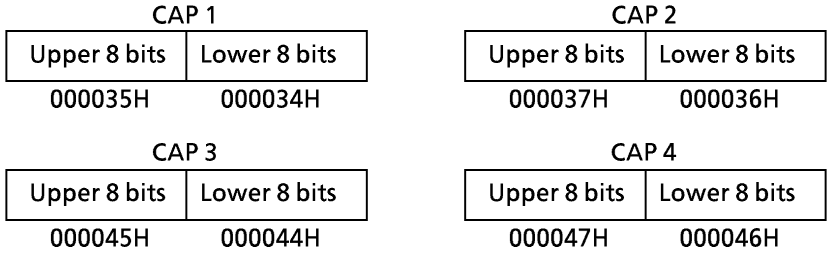
TREG4, TREG6 and register buffer are allocated to the same memory addresses 000030H / 000031H and 000040H / 000041H. When <DB4EN, DB6EN> = 0, same value will be written in both the timer register and register buffer. When <DB4EN, DB6EN> = 1, the value is written into only the register buffer.

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

③ Capture Register

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

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



④ Capture Input Control

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<CAP12M 1, 0> / T5MOD<CAP34M 1, 0>.

- When T4MOD<CAP12M 1, 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 / INT4) and TI6 pin (also used as PB4 / INT7) input, while data is loaded to CAP2 / CAP4 at the rise edge of 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 TI4 pin / TI6 pin input, while 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<PRRUN> 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 / 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 inversion can be set for each element by T4FFCR<CAP2T4, CAP1T4, EQ5T4, EQ4T4> / T5FFCR<CAP4T6, CAP3T6, EQ7T6, EQ6T6>. TFF4 / 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 (also 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 (also 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 have shows Timer 4 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
                (** = 01, 10, 11) disable the capture function.
TREG5     ← * * * * * * * *      Set the interval time (16 bits).
                * * * * * * * *
TRUN      ← 1 X - 1 - - - -      Start timer 4.
```

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

(2) 16-bit Event Counter Mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4 pin / 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.

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

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

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

Timer 4 and Timer 5 can be operated all the same, here shows Timer 4 only.

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

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

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

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

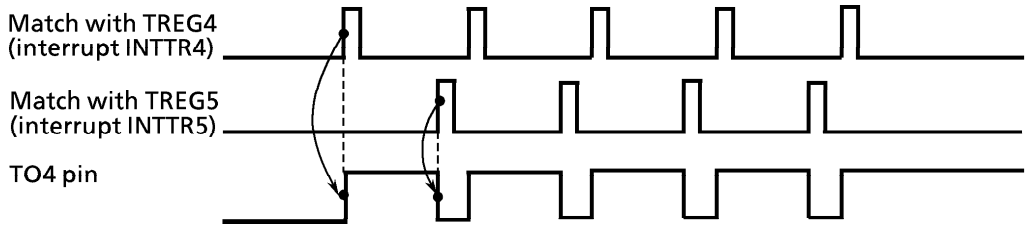


Figure 3.9 (11) Programmable Pulse Generation (PPG) Output Waveforms

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

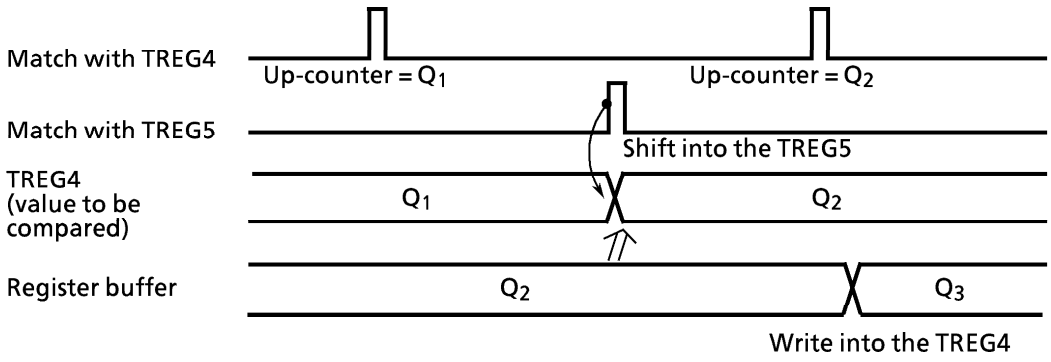


Figure 3.9 (12) Operation of Register Buffer

Shows the block diagram of this mode.

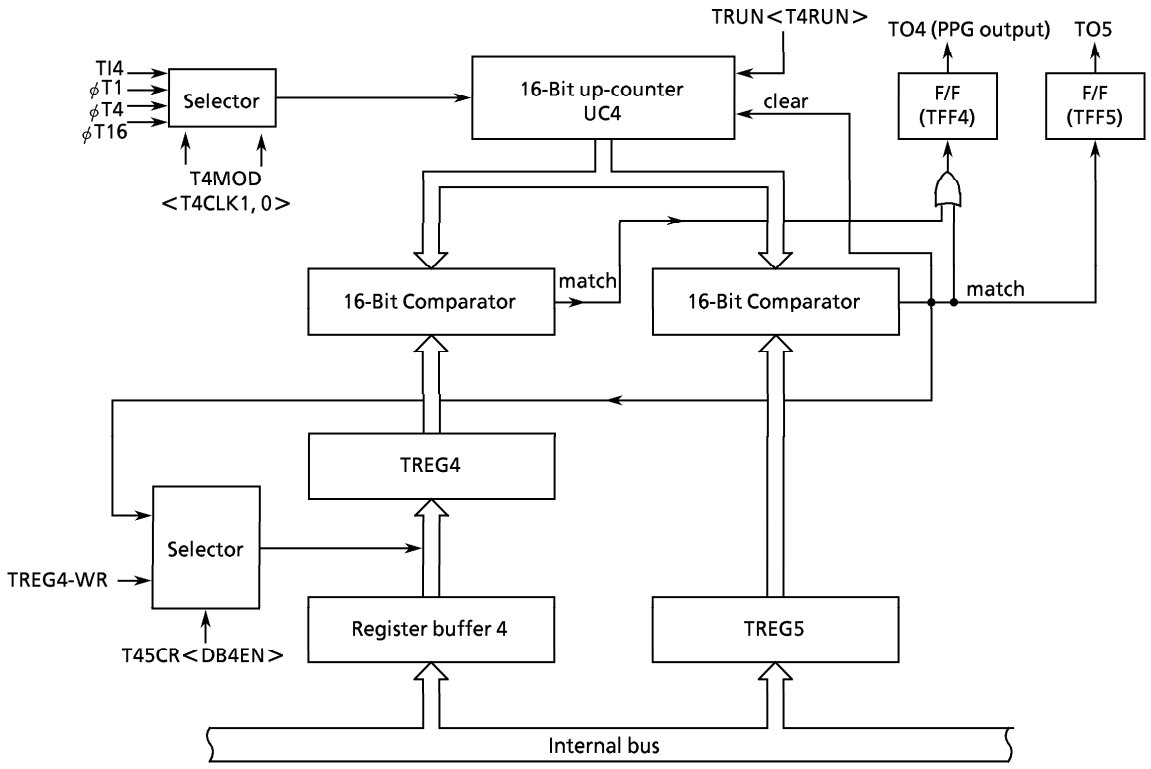


Figure 3.9 (13) Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

Timer 4 and Timer 5 can be operated all the same. Here shows Timer 4 only.

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

① One-shot pulse output from external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot Pulse Output from External Trigger Pulse

Set the up-counter UC4 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up-counter into capture register CAP1 at the rise edge of the TI4 pin. Then set to T4MOD<CAP12M1, 0>=01.

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

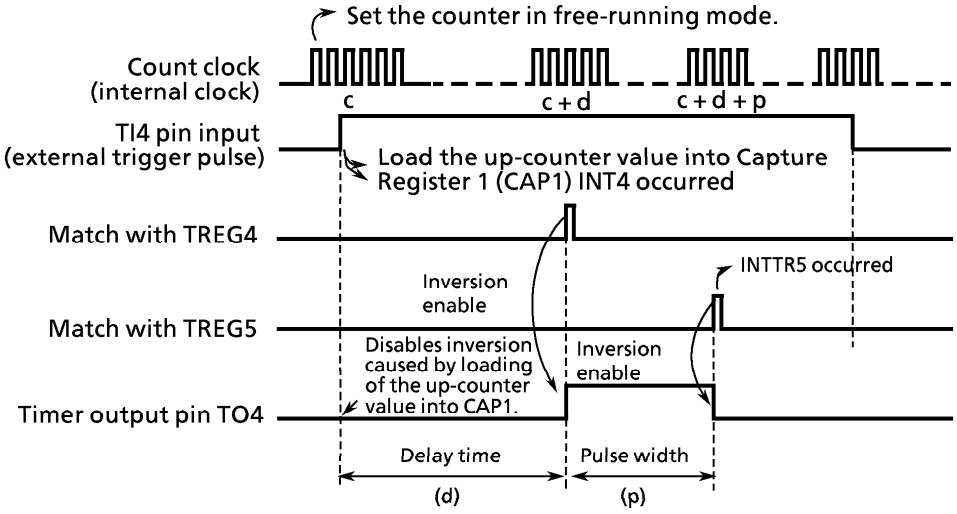
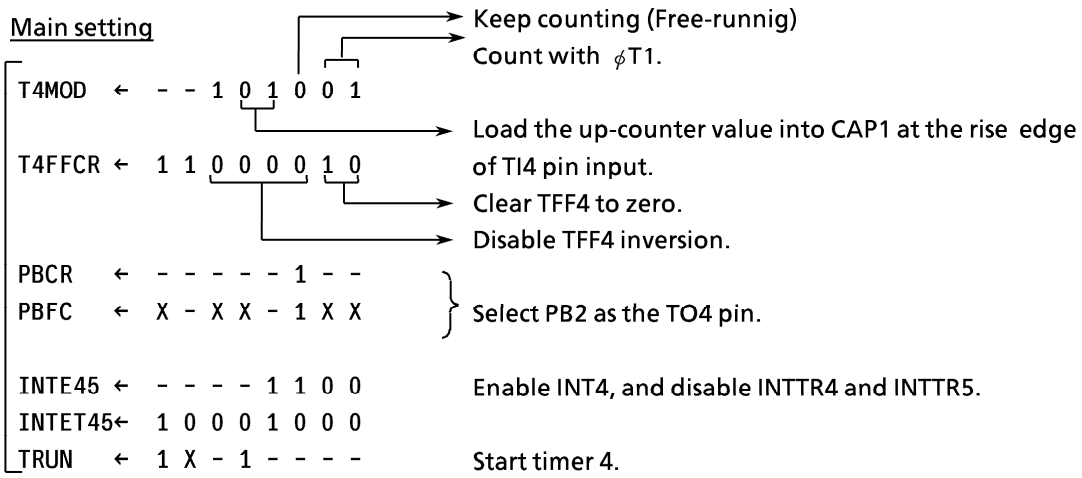


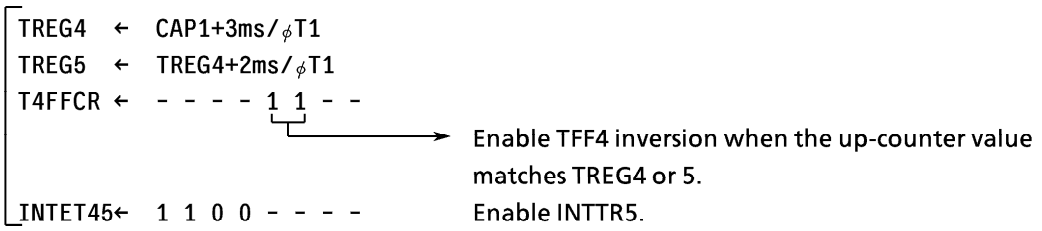
Figure 3.9 (14) One-Shot Pulse Output (with Delay)

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

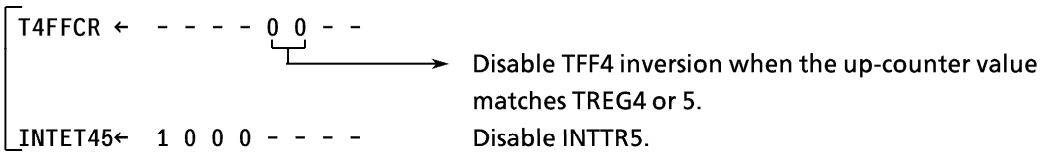
Main setting



Setting of INT4



Setting of INTTR5



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

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

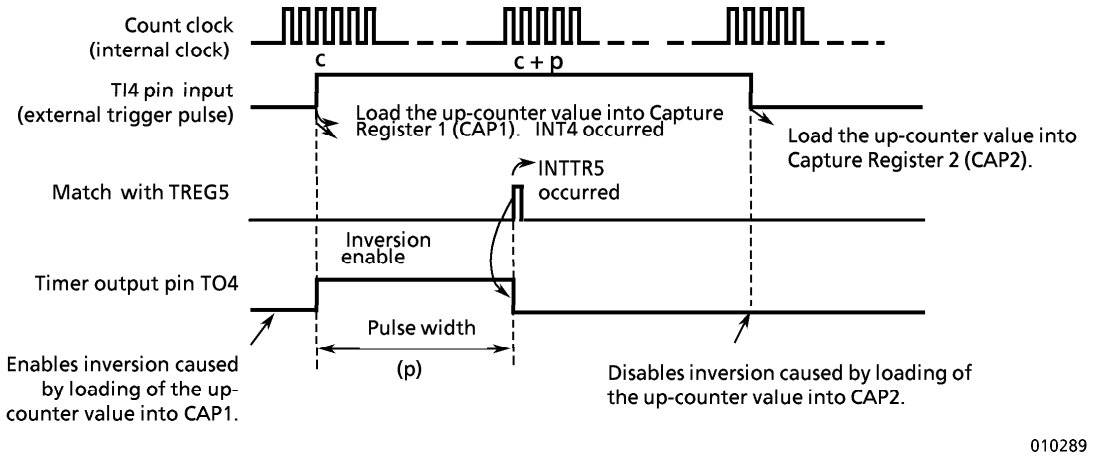


Figure 3.9 (15) One-Shot Pulse Output (without Delay)

② Frequency Measurement

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

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

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

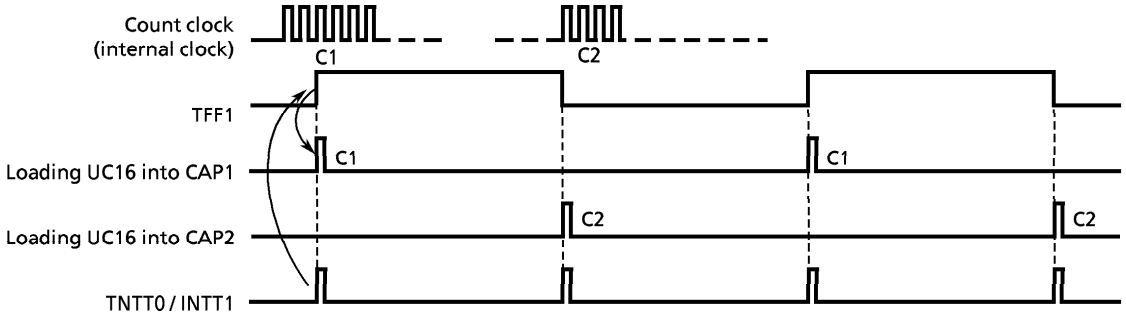


Figure 3.9 (16) Frequency Measurement

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

③ Pulse Width Measurement

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer / event counter counting (free-running) with the internal clock input, the external pulse is input through the TI4 pin. Then the capture function is used to load the UC4 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT4 occurs at the falling edge of TI4.

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

For example, if the internal clock is 0.8 microseconds and the difference between CAP1 and CAP2 is 100, the pulse width will be $100 \times 0.8\ \mu s = 80\ \mu s$.

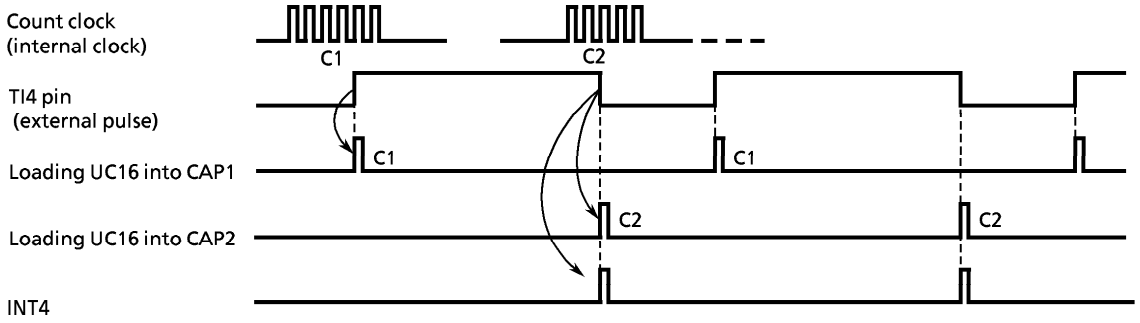


Figure 3.9 (17) Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD<CAP12M1, 0>=10), external interrupt INT4 occurs at the falling edge of TI4 pin input. In other modes, it occurs at the rising edge.

The width of "L" level can be measured from the difference between the first C2 and the second C1 at the second INT4 interrupt.

④ Time Difference Measurement

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

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

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

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

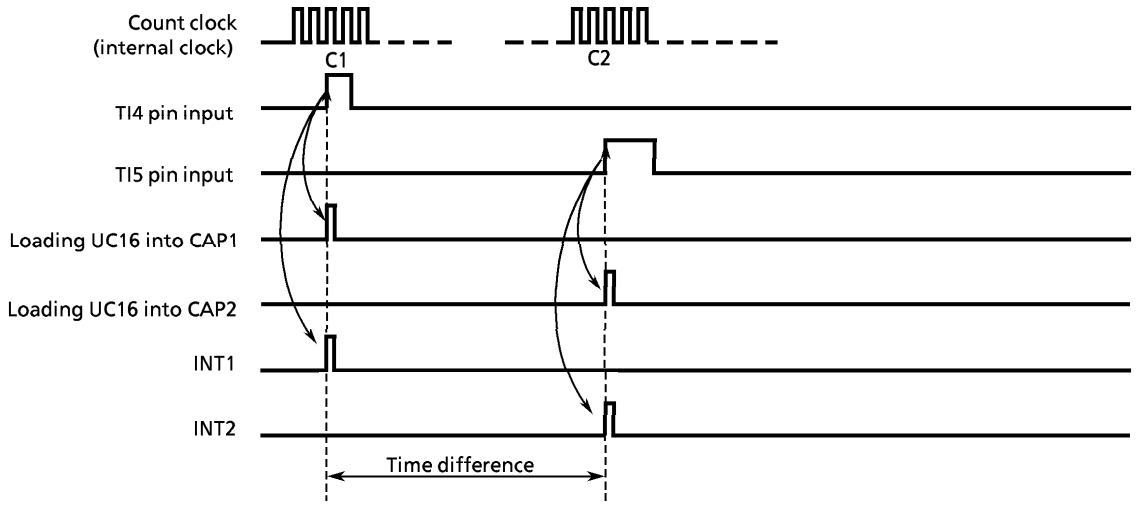


Figure 3.9 (18) Time Difference Measurement

(5) Different Phased Pulses Output Mode

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

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

This mode can be used only Timer 4.

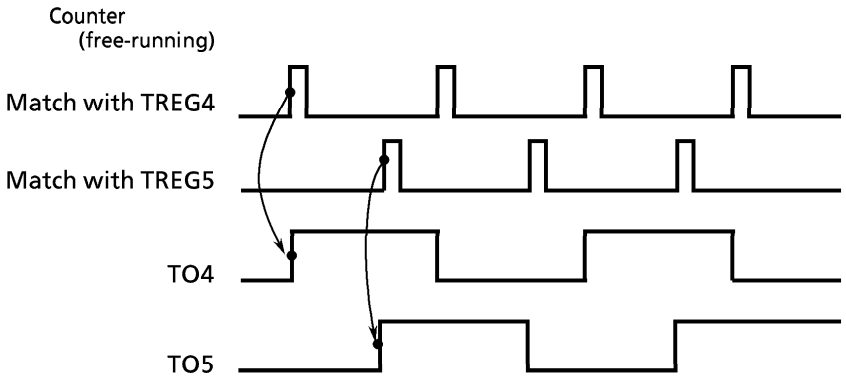


Figure 3.9 (19) Phase Output

Cycles (counter overflow time) of the above output waves are listed on table 3.9 (2).

The following table shows cycles (counter over flow time) of the above output wave.

	20 MHz	25 MHz
ϕT1	26.214 ms	20.97 ms
ϕT4	104.856 ms	83.88 ms
ϕT16	419.424 ms	335.54 ms

3.10 Stepping Motor Control / Pattern Generation Port

TMP95C061B contains 2 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 7.

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

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

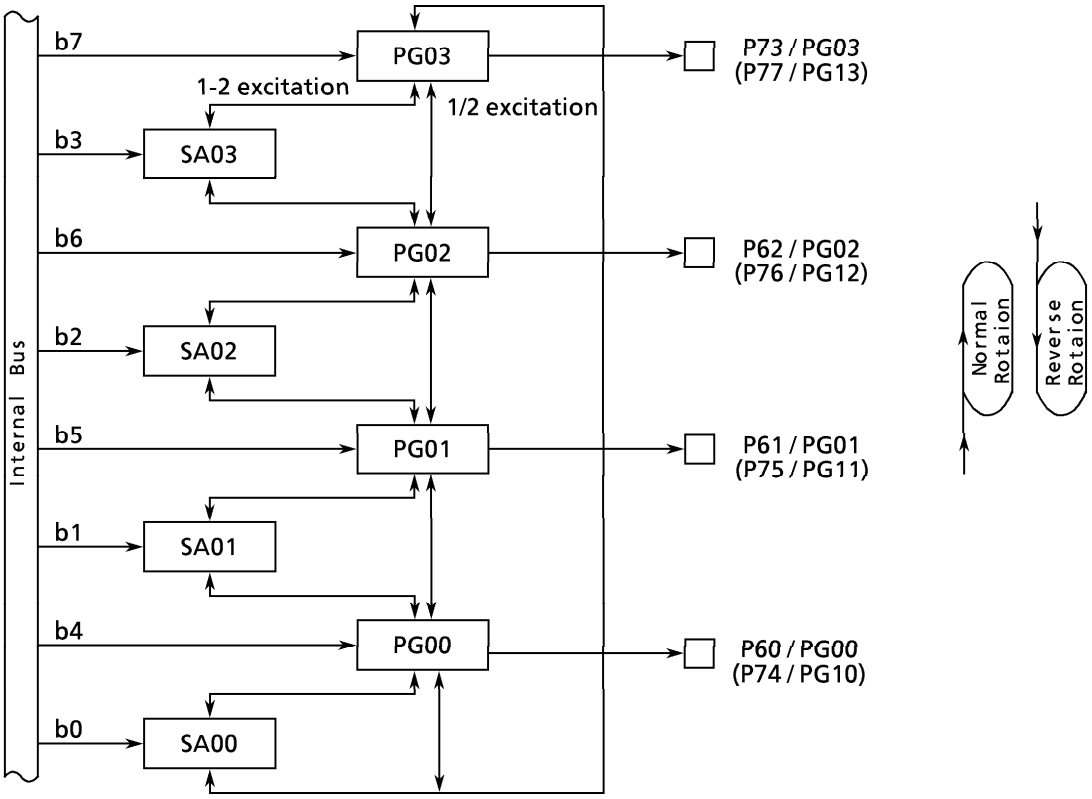


Figure 3.10 (1) PG Block Diagram

PG01CR (004EH)

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

PG0 Trigger input enable

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

Set the operation mode of PG0

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

PG0 (stepping motor control)
Rotaing direction control

0	Normal rotation / PG mode
1	Reverse rotation

Selecting PG0 write mode

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

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

PG01CR (004EH)

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

PG1 trigger input enable

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

Set the operation mode of PG1

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

PG1 (stepping motor control)
Rotaing direction control

0	Normal rotation / PG mode
1	Reverse rotation

Selecting PG1 write mode

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

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

PG0REG (004CH)

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

Prohibit Read modify write

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

PG1REG (004DH)

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

Prohibit Read modify write

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

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
Function	Fix at "0" Always read as "0"				PG1 Shift trigger 0: 8-bit timer trigger 3 (timer2, 3) 1: 16-bit timer trigger (timer5)	PG0 Shift trigger 0: 8-bit timer trigger 1 (timer0, 1) 1: 16bit timer trigger (timer4)	Double buffer 0: Disable 1: Enable Double buffer of TREG6	Double buffer of TREG4

Double buffer control

0	Disable
1	Enable

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

Selecting PG0 shift trigger

0	8-bit timer trigger (timer 0, 1)
1	16-bit timer trigger (timer 4)

Selecting PG1 shift trigger

0	8-bit timer trigger (timer 2, 3)
1	16-bit timer trigger (timer 5)

Write "0" to this bit, when this register is modified.

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

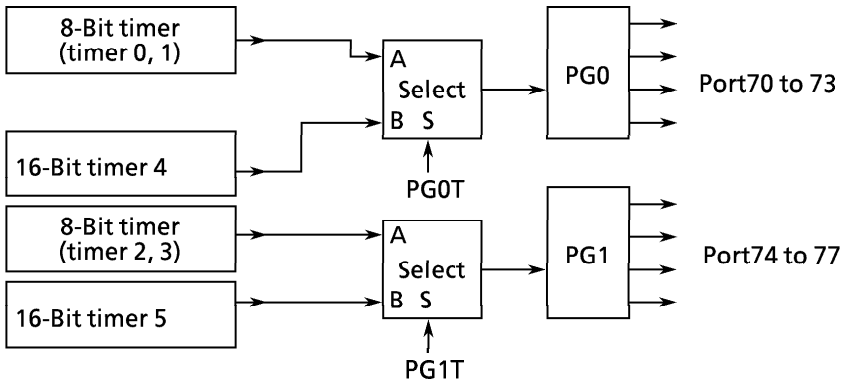


Figure 3.10 (6) Connection of Timer and Pattern Generator

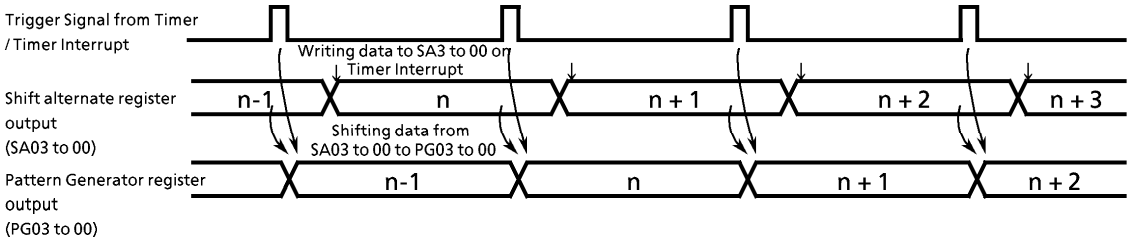
(1) Pattern Generation Mode

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

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

The output of this pattern generator is output to port 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

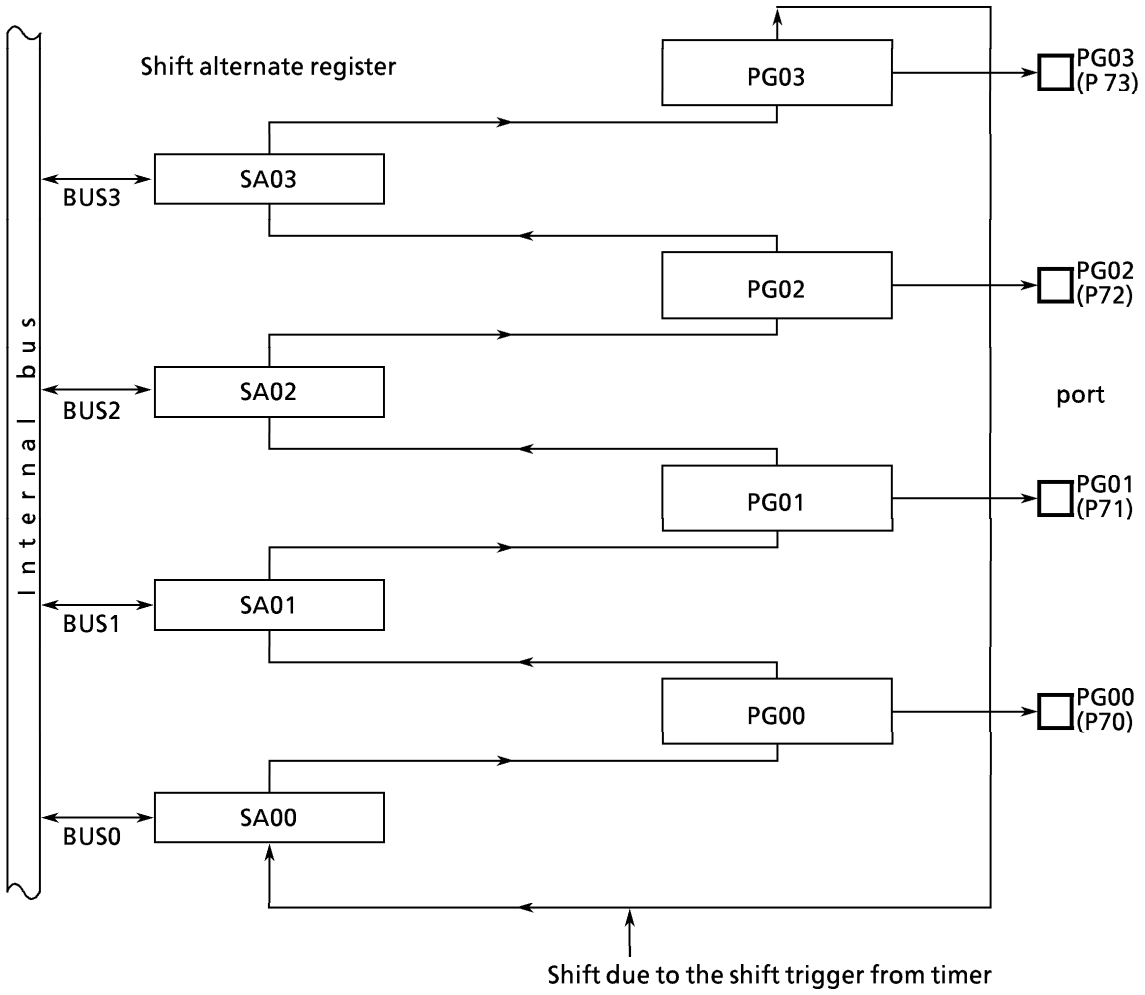


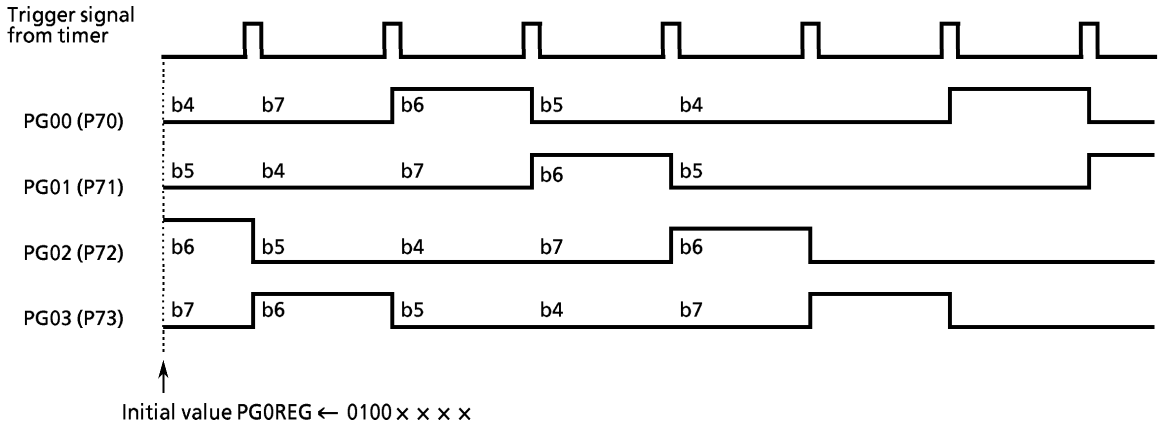
Figure 3.10 (7) Pattern Generation Mode Block Diagram (PG0)

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

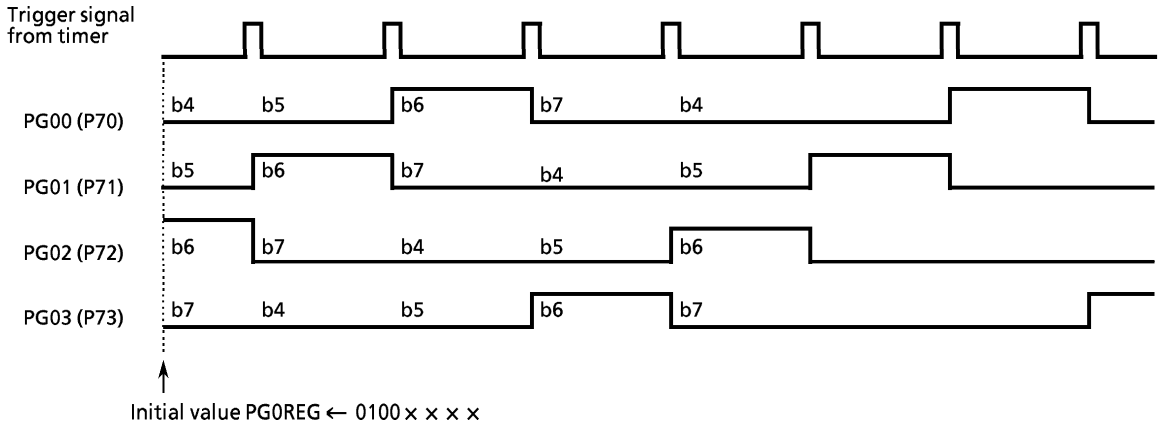
① 4-phase 1or 2 Excitation

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

010289

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

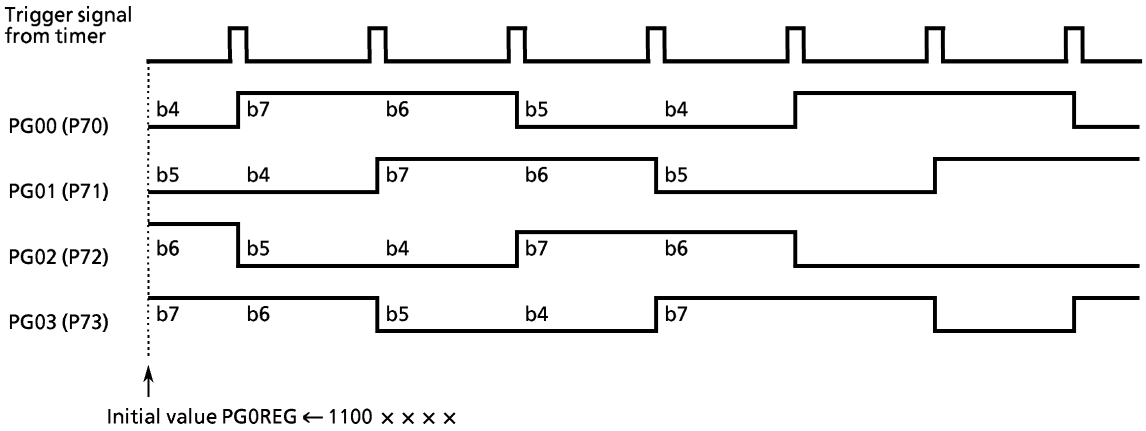


Figure 3.10 (9) Output Waveforms of 4-Phase 2 Excitation (Normal Rotation)

The operation when channel 0 is selected is explained below.

The output latch of PG0 (also used as P7) 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 excitation will be selected when only one bit is set to "1" during the initialization of PG, while 4-phase 2 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 or 2 excitation mode is selected.

Figure 3.10 (10) shows the block diagram.

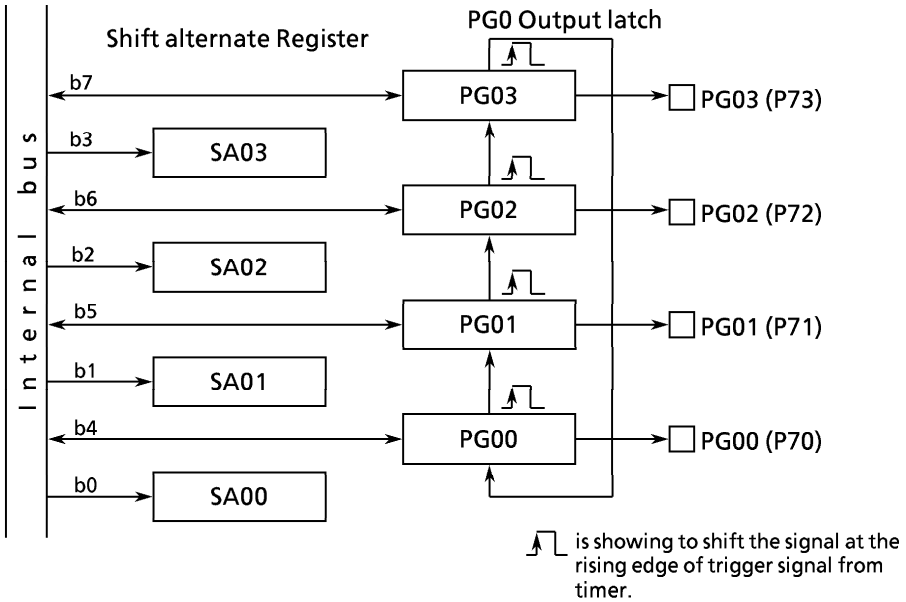
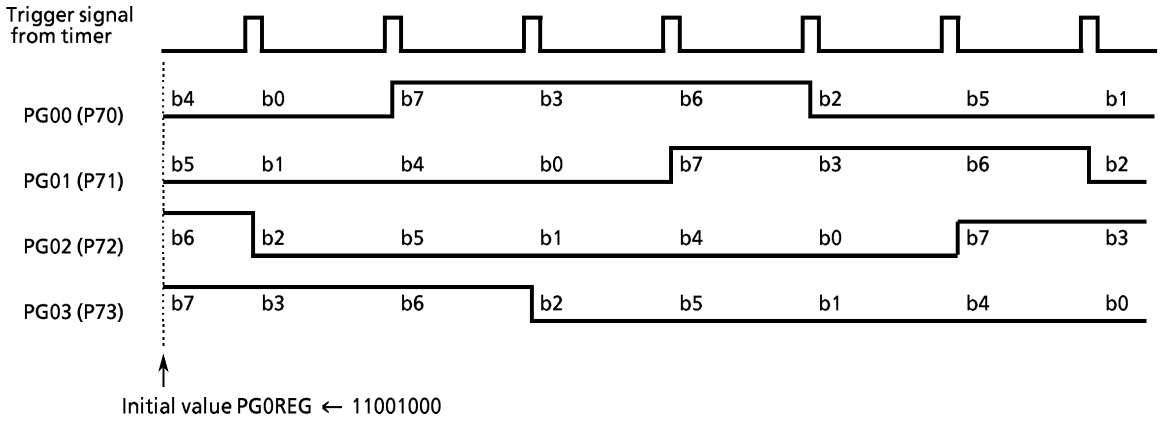


Figure 3.10 (10) Block Diagram of 4-Phase 1 or 2 Excitation (Normal Rotation)

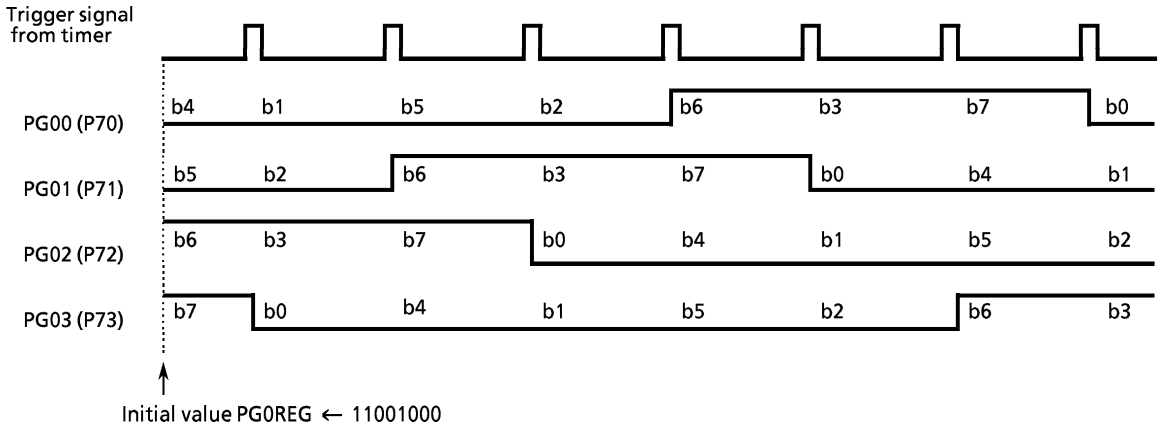
② 4-Phase 1-2 Excitation

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



Note : bn denotes the initial value PG0REG ← b7 b6 b5 b4 b3 b2 b1 b0

① Normal Rotation



② Reverse Rotation

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

The initialization for 4-phase 1-2 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.

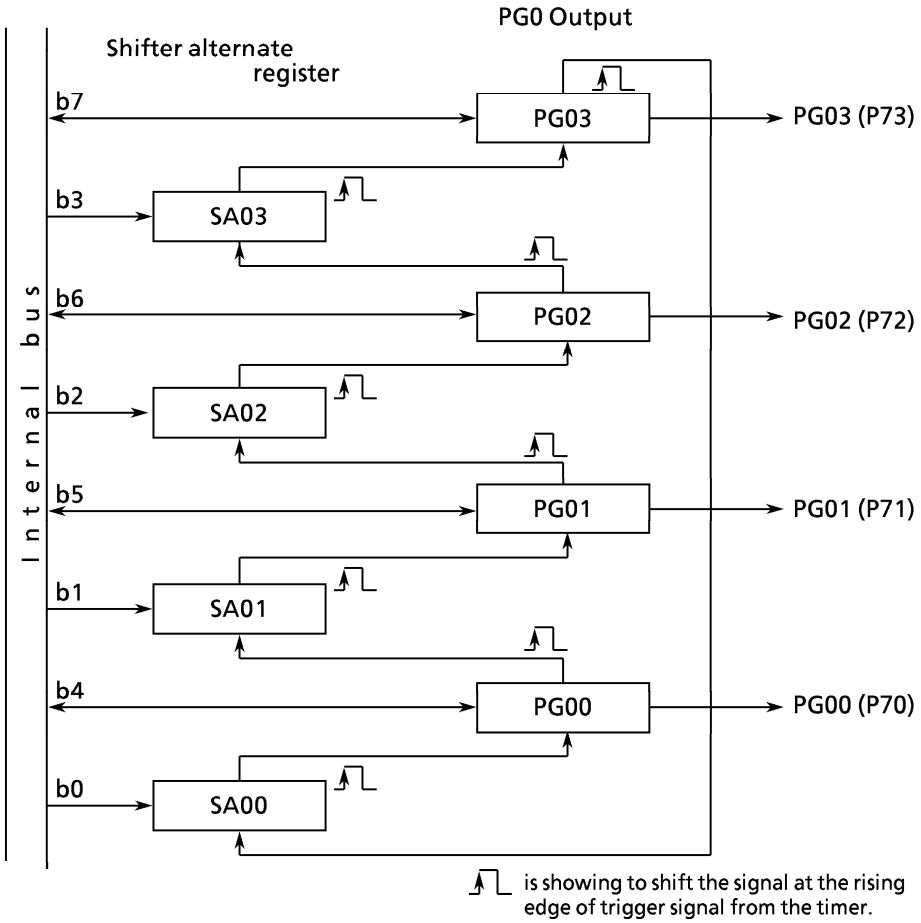


Figure 3.10 (12) Block Diagram of 4-Phase 1-2 Excitation (Normal Rotation)

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

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - - - - - 0     Stop timer 0, and clear it to zero.
TMOD    ← 0 0 X X - - 0 1     Set 8-bit timer mode and select φT1 as the input clock of timer 0.
TFFCR   ← X X X 0 1 0 1 0     Clear TFF1 to zero and enable the inversion trigger by timer 0.
TREG0   ← * * * * * * * *     Set the cycle in timer register.
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 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, TFF3, TFF4, TFF5, and TFF6) and differs as shown in Table 3.10 (1) depending on the operation mode of the timer.

Table 3.10 (1) Select of Trigger Signal

	TFF1 inversion	PG shift
8-bit timer mode	Selected by TFFCR <FF1IS> when the up-counter value matches TREG0 or TREG1 value.	←
16-bit timer mode	When the up-counter value matches with both TREG0 and TREG1 values (The value of up-counter = TREG1*2^8 + TREG0)	←
PPG output mode	When the up-counter value matches with both TREG0 and TREG1	When the up-counter value matches TREG1 value (PPG cycle)
PWM output mode	When the up-counter value matches TREG0 value and PWM cycle.	Trigger signal for PG is not generated.

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

Channel 1 of PG can be synchronized with the 16-bit timer Timer4 / Timer5. In this case, the PG shift trigger signal from the 16-bit timer is output only when the up-counter UC4 / UC5value matches TREG5 / TREG7.

When using a trigger signal from Timer4, 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 Timer5, 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 "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).

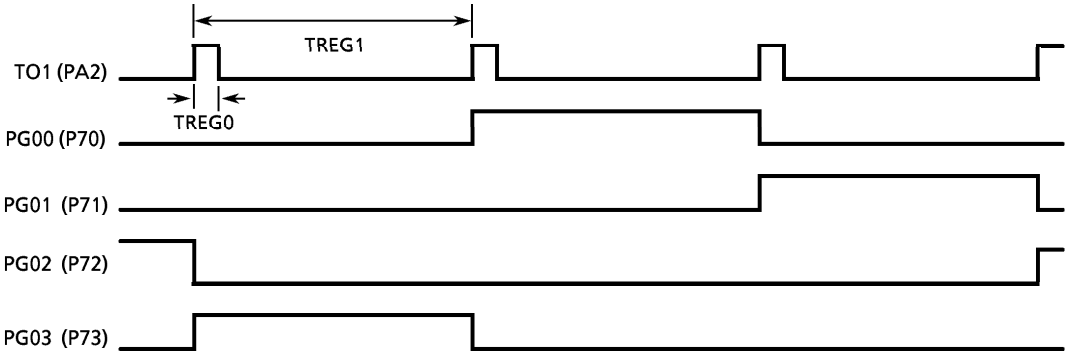


Figure 3.10 (13) Output Waveforms of 4-Phase 1 Excitation

Setting example:

```
          7 6 5 4 3 2 1 0
TRUN    ← - X - - - - 0 0   Stop timer 0, and clear it to zero.
T01MOD  ← 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

TMP95C061B contains 2 serial Input/Output channels. The serial channel has the following operation modes.

- I/O interface mode — Mode 0: To transmit and receive I/O data as well as the synchronizing signal SCLK for extending I/O.
- Asynchronous transmission (UART) mode
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2, a parity bit can be added. Mode 3 has 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.

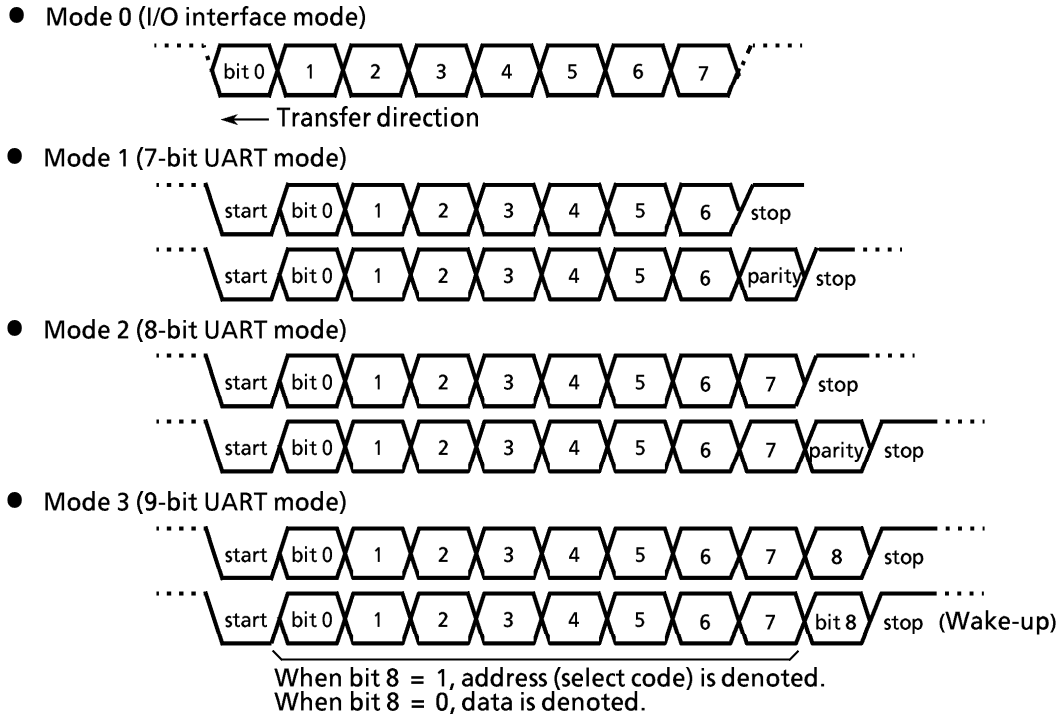


Figure 3.11 (1) Data Formats

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 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (there is no $\overline{\text{RTS}}$ pin, so any 1 port must be controlled by software), it is possible to halt data send until the CPU finishes reading receive data every time a frame is received. (Handshake function)

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

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

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of 4 clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the internal prescaler (shared by 8-bit / 16-bit timer) by the value 1 to 16 (2 to 16 in the channel 1).

In addition to the clock from the internal baud rate generator, an arbitrary baud rate can be obtained from the external input clock (SCLK0) in the serial channel 0. Moreover, in I/O interface mode, a sync signal (SCLK) can be input and data transfer performed using this external clock.

3.11.1 Control Registers

The serial channel is controlled by 3 control registers SC0CR, SC0MOD and BR0CR. Transmitted and received data are stored in register SC0BUF.

SC0MOD (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data Bit 8	Hand shake 0: CTS disable 1: CTS enable	Receiving Function 0: Receive disable 1: Receive enable	Wake up Function 0: disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO2 Trigger 01: baud rate generator 10: Internal clock Φ1 11: external clock (SCLK0)	

Serial transmission clock (UART)

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

Serial transmission mode

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

Wake-up Function

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

Receiving Function

0	Receive Disable
1	Receive Enable

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

0	Disable (always Transferable)
1	Enable

Transmission data bit 8

Note : There is SC1MOD (56H) in Channel1

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

SC0CR (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to Zero when read)			R/W	
After reset		0	0	0	0	0	0	0
Function	Received data Bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error Overrun	Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud Rate Generator 1: SCLK0 Pin input

Select I/O interface input clock (Note1)

0	Baud rate generater
1	SCLK0 Pin input

Edge selection in SCLK pin input mode

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

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

Enable parity addition

0	Prohibition (disable)
1	Permission (enable)

Addition / check of even parity

0	Odd parity
1	Even parity

Receving data bit 8

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 0, SC0CR)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	—		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fix at "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

0000	Divide by 16
0001	Divide by 1 (no division)
0010 to 1111	Divide by 2 to 15

Selecting the input clock of baud rate generator

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

Note : · Set TRUN<PRRUN> to "1" when the baud rate generator is used.
· The baud rate generator frequency can be divided by 1 in UART mode only. Do not use this setting in I / O interface mode.
· Don't read from or write to BR0CR register during sending or receiving.

Figure 3.11 (4) Baud Rate Generator Control Register (channel 0, BR0CR)

SC0BUF (50H)

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

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

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	—	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfered data Bit 8	Fix at "0"	Receiving Function 0: Receive disable 1: Receive Enable	Wake up Function 0: disable 1: enable	Serial Transmission mode 00: I/O interface mode 01: 7-Bit UART 10: 8-Bit UART 11: 9-Bit UART		Serial Transmission clock (UART) 00: TO2 Trigger 01: Baud rate generator 10: Internal clock Φ1 11: don't care	

Serial transmission clock (For UART)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	don't care

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

Serial transmission mode

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

Wake up Function

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

Receiving control

0	Receive disable
1	Receive enable

Transmission data bit 8

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

SC1CR (0055H)

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

Select I/O interface input clock (Note1)

0	Baud rate generate
1	SCLK1 Pin input

Edge selection in SCLK pin input mode

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

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

Enable parity addition

0	Disable
1	Enable

Addition / check of even parity

0	Odd Parity
1	Even Parity

Receiving data bit 8

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

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

BR1CR (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	—		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fix at "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

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

Selecting the input clock of baud rate generator

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

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

Figure 3.11 (8) Baud Rate Generator Control Register (channel 1, BR1CR)

SC1BUF (0054H)

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

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

P8FC (001BH)

	7	6	5	4	3	2	1	0
bit Symbol			P85F		P83F	P82F		P80F
Read/Write			W		W	W		W
			0		0	0		0
			0: PORT 1: SCLK1		0: PORT 1: TxD1	0: PORT 1: SCLK0		0: PORT 1: TxD0

Prohibit Read modify write

Setting TXD0 / TXD1 output

0	Port output
1	TxD output

Setting SCLK0 / SCLK1 output

0	Port output
1	SCLK output

Figure 3.11 (10) Port 8 Function Register (P8FC)

ODE (0058H)

	7	6	5	4	3	2	1	0
bit Symbol							ODE1	ODE0
Read/Write							R/W	
							0	0
							P83 0: CMOS 1: OPEN Drain	P80 0: CMOS 1: OPEN Drain

Setting P80 as Open-drain output

0	CMOS output
1	Open-drain output

Setting P83 as Open-drain

0	CMOS output
1	Open-drain output

Figure 3.11 (11) Port 8 Open Drain Enable Register (ODE)

3.11.2 Configuration

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

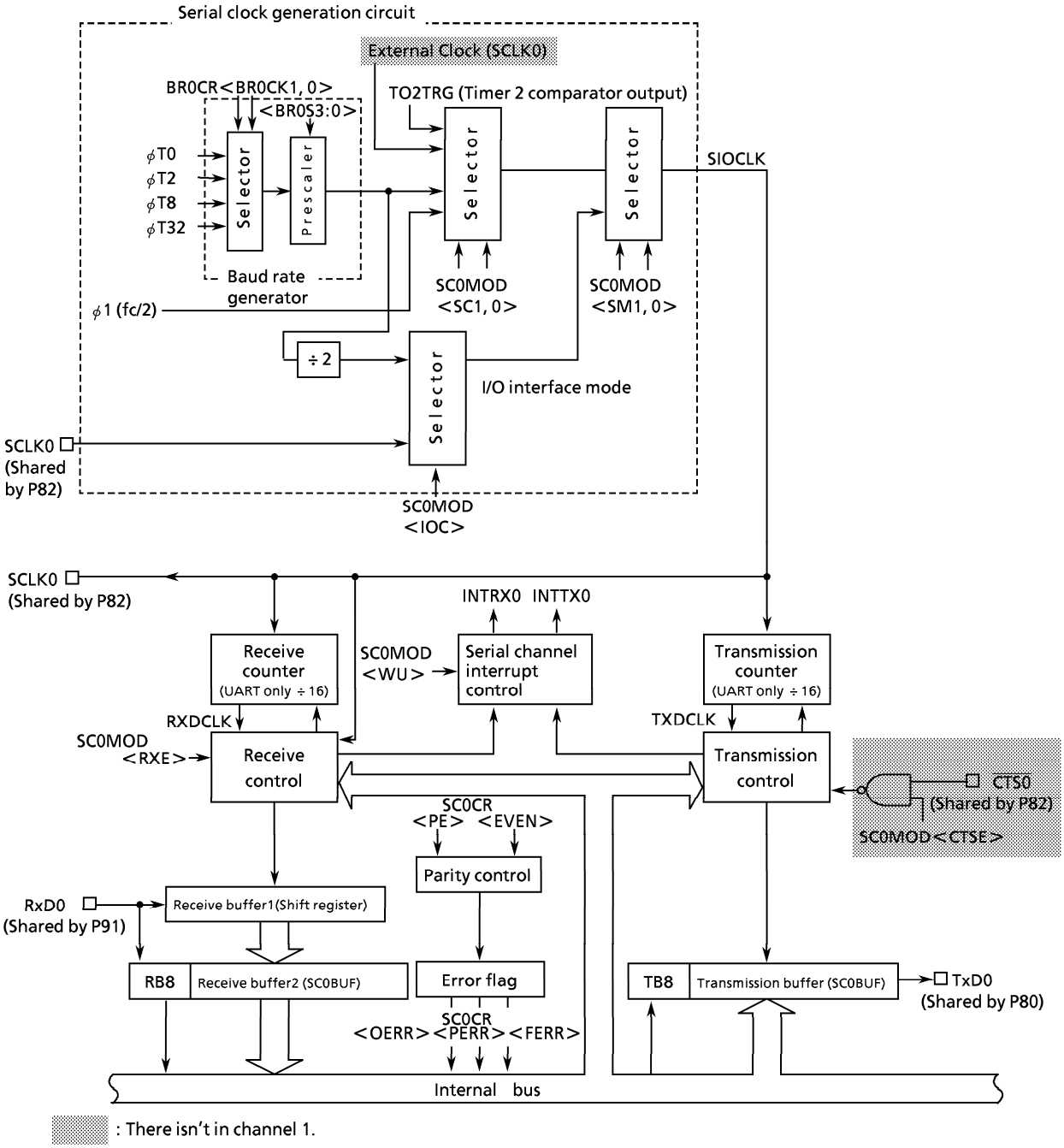


Figure 3.11 (12) Block Diagram of the Serial Channel 0

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

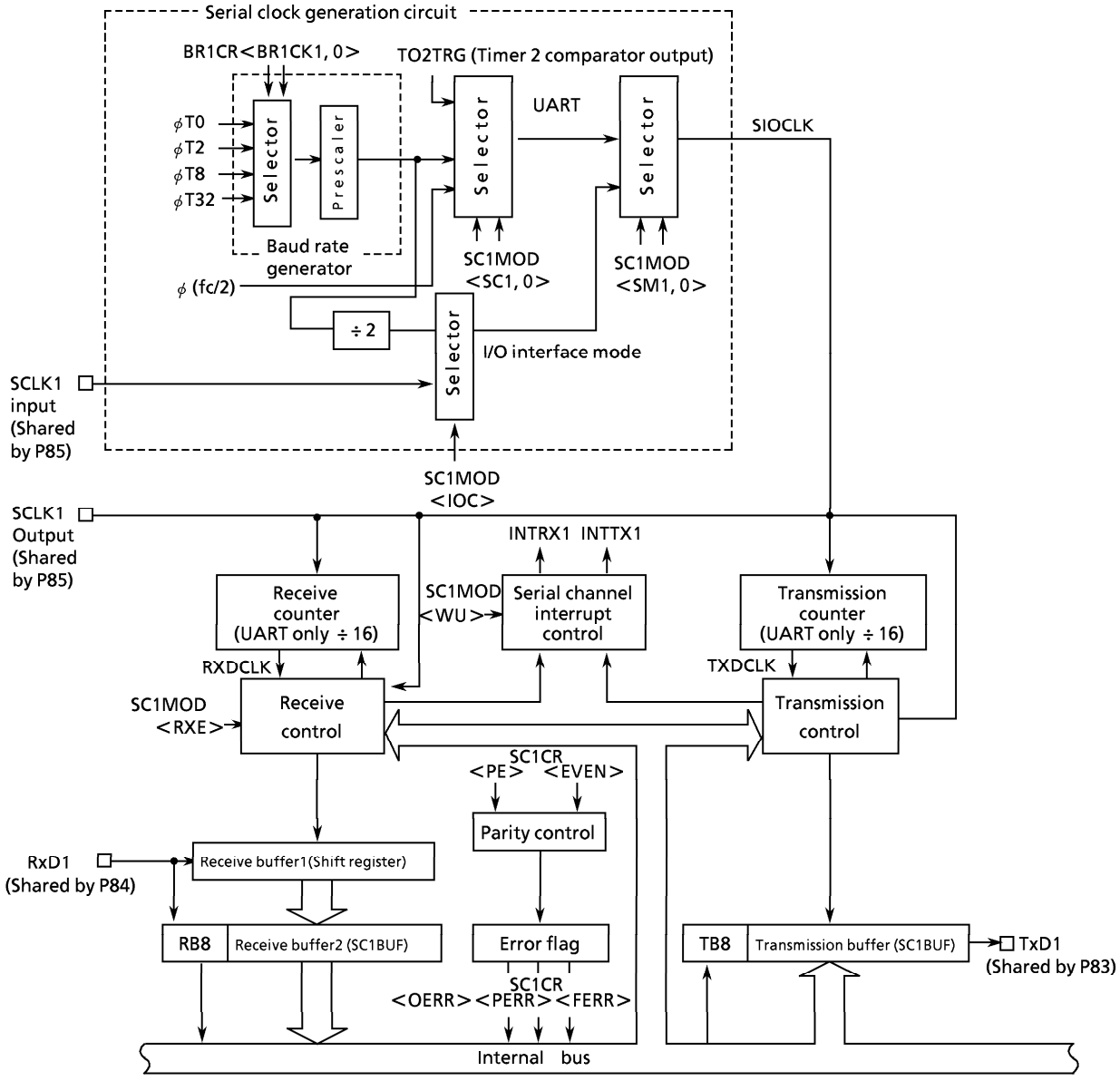


Figure 3.11 (13) Block Diagram of the Serial Channel 1

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, ϕT0 (4/fc), ϕT2 (16/fc), ϕT8 (64/fc), or ϕT32 (256/fc) are 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 1 to 16 values (2 to 16 in the channel 1) 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

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

Therelation of source clock and input clock is as below,

ϕT0 = 4/fc
ϕT2 = 16/fc
ϕT8 = 64/fc
ϕT32 = 256/fc

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

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

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.

Moreover, the external clock input can also be used as the serial clock. The baud rate in this case is determined as follows.

Baud Rate = external clock input ÷ 16

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

Unit (Kbps)

fc [MHz]	Input clock / Frequency divisor	ϕT0 (4/fc)	ϕT2 (16/fc)	ϕT8 (64/fc)	ϕT32 (256/fc)
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

Note1: 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 ϕT1) is used)

Unit (Kbps)

TREG2 \ fc	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

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

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

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

Input clock of Timer 2

ϕT1 = 8/fc

ϕT4 = 32/fc

ϕT16 = 128/fc

Note1: 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) In I/O interface mode

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

2) In universal asynchronous receiver transmitter (UART) mode

Basic clock SIOCLK is selected from one of the following depending on the setting of the<SC1, 0>bits of the SC0MOD or SC1MOD register: the clock from the baud rate generator, internal clock ϕ1 (500Kbps at fc=16 MHz), a match detect signal from timer 2 or an external clock (channel 0 only) .

③ Receiving Counter

The receiving counter is a 4-bit binary counter used in asynchronous communication (UART) mode and counts up by SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 bit of data, and the data bit is sampled three times at 7th, 8th and 9th clock.

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

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

④ Receiving Control

1) I/O interface mode

When in SCLK output mode with the setting of SC0CR/SC1CR<IOC>="0", RxD0/1 signal will be sampled at the rising edge of shift clock which is output to SCLK0/1 pin.

When in SCLK input mode with the setting SC0CR/SC1CR<IOC>="1" RxD0/1 signal will be sampled at the rising edge or falling edge of SCLK input according to the setting of SC0CR/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 are also evaluated by the rule of majority.

⑤ Receiving Buffer

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

Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data is stored in the receiving buffer 1, the stored data are transferred to 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> is still preserved. Reading data from receive data buffer 2 (SC0BUF / SC1BUF) clears interrupt request flags INTRX0<IRX0C> and INTRX1<IRX1C>.

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.

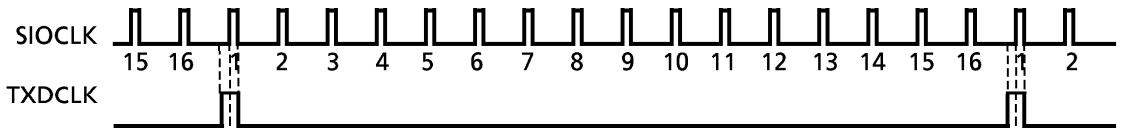


Figure 3.11 (14) Generation of Transmission Clock

⑦ Transmission Controller

1) I/O interface mode

In SCLK output mode with the setting of SC0CR/SC1CR<IOC> = "0", the data in the transmission buffer are output bit by bit to TxD0/1 pin at the rising edge of shift clock which is output from SCLK0/1 pin.

In SCLK input mode with the setting of SC0CR/SC1CR<IOC> = "1", the data in the transmission buffer are output bit by bit to TxD0/1 pin at the rising edge or falling edge of SCLK input according to the setting of SC0CR/SC1CR<SCLKS> register.

2) Asynchronous communication (UART) mode

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

Handshake function

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

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

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

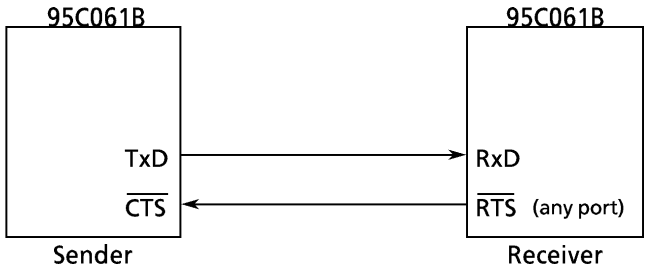
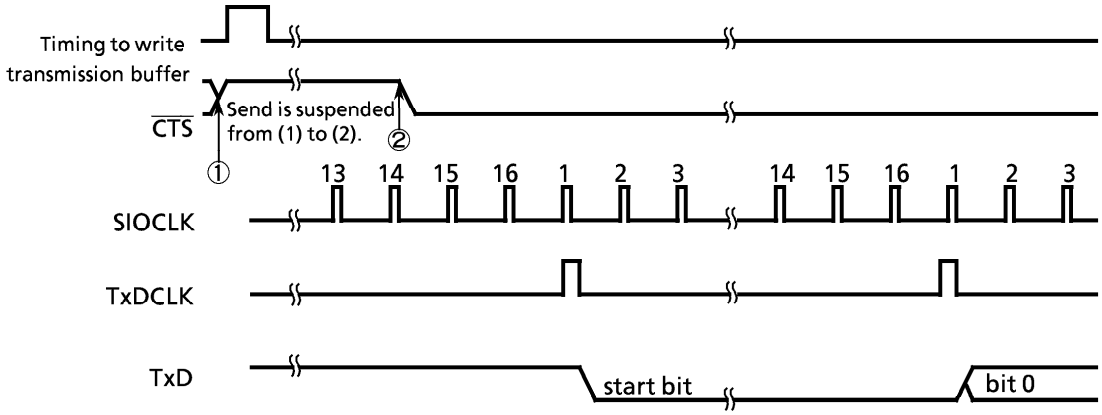


Figure 3.11 (15) Handshake Function



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

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

Figure 3.11 (16) Timing of $\overline{CTS}$ (Clear to send)

⑧ Transmission Buffer

Transmission buffer (SC0BUF / SC1BUF) shifts out 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 / SC1BUF), and data are transmitted after being stored in SC0BUF<TB7>/ SC1BUF<TB7> when in 7-bit UART mode while in SC0MOD <TB8>/ SC1BUF<TB8> when in 8 -bit UART mode. <PE> and <EVEN> must be set before transmission data are written in the transmission buffer.

For receiving, data are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SC0BUF / SC1BUF), and then compared with SC0BUF<RB7>/ SC1BUF when in 7-bit UART mode and with SC0CR<RB8>/ SC1CR 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 are stored in receiving buffer 2 (SCBUF0/1), an overrun error will occur.

2. Parity error <PERR>

   The parity generated for the data shifted in receiving buffer 2 (SCBUF0/1) is compared with the parity bit received from RxD0/1 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 (parity bit)	←
Overrun error timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit

Transmitting

Mode	9 Bit	8 Bit + parity	8 Bit, 7 Bit + parity, 7 Bit
Interrupt timing	Just before stop 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 (SC0BUF/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 (SC0BUF/SC1BUF) (that is, immediately after last 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 of for transmitting or receiving data to or from the external shifter register.

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

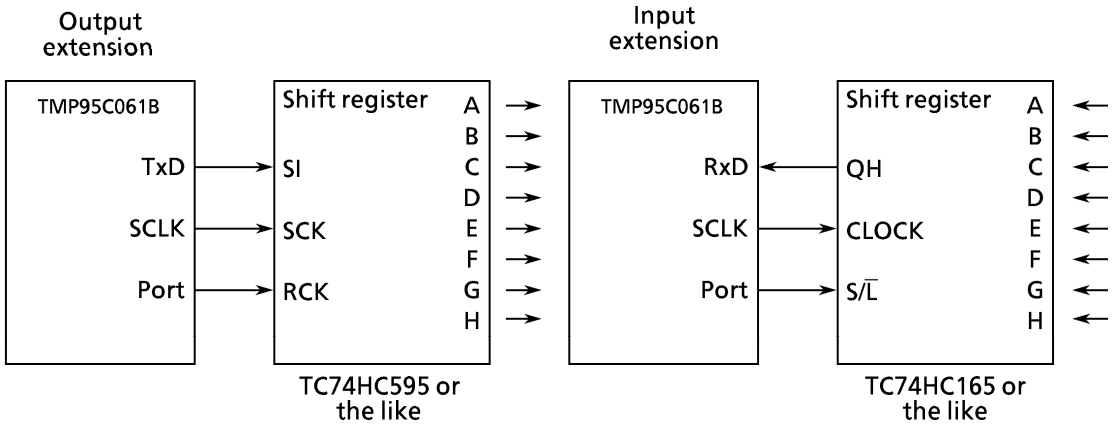


Figure 3.11 (18) Example of SCLK Output Mode Connection

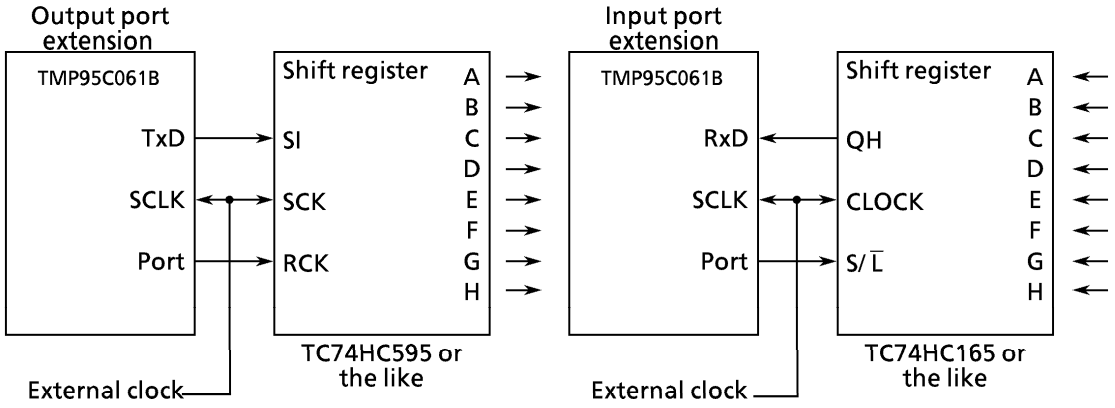


Figure 3.11 (19) Example of SCLK Input Mode Connection

① Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TxD0/1 pin and SCLK0/1 pin, respectively, each time the CPU writes data in the transmission buffer. When all data is output, INTES0<ITX0C> / INTES1<ITX1C> will be set to generate INTTX0/1 interrupt.

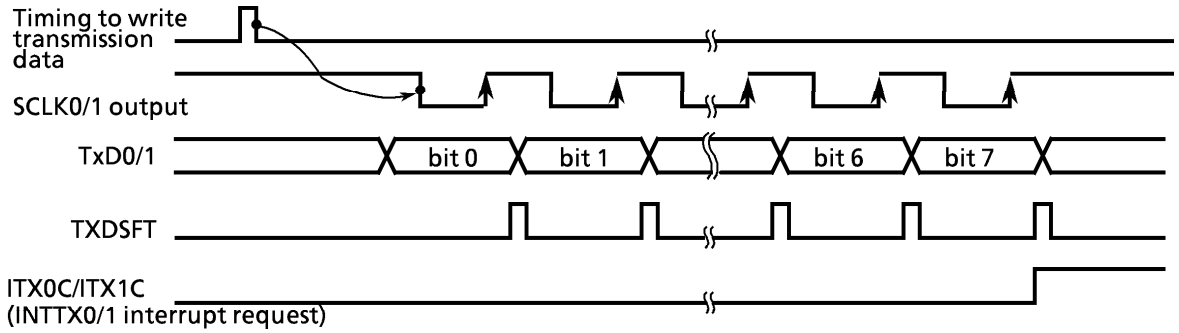


Figure 3.11 (19) Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK input mode, 8-bit data are output from TxD0/1 pin when SCLK input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES0<ITX0C>/INTES1<ITX1C> will be set to generate INTTX0/1 interrupt.

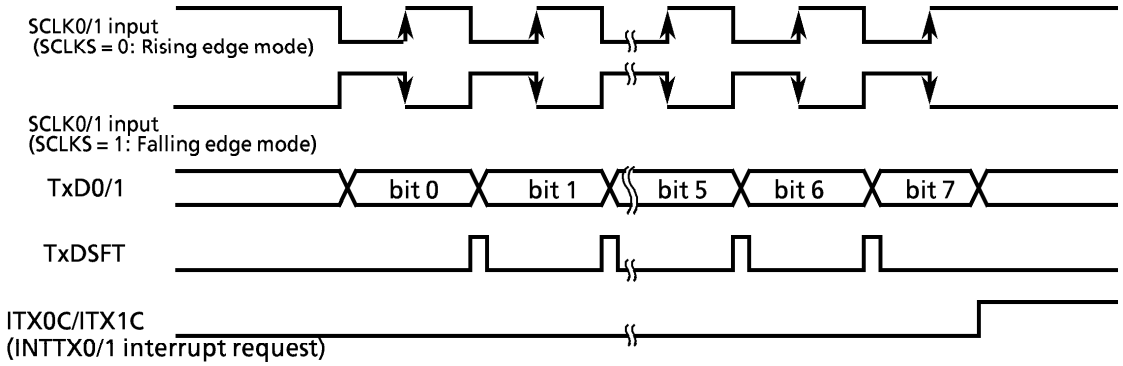


Figure 3.11 (20) Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

In SCLK output mode, synchronous clock is outputted from SCLK0/1 pin and the data are shifted in the receiving buffer 1 whenever the receive interrupt flag INTES0<IRX0C>/INTES1<IRX1C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred in the receiving buffer 2 (SC0BUF/SC1BUF) at the timing shown below, and INTES0<IRX0C> / INTES1<IRX1C> will be set again to generate INTRX0/1 interrupt.

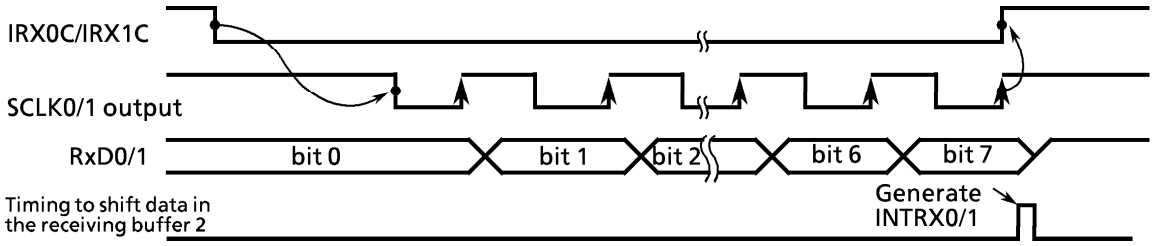


Figure 3.11 (21) Receiving Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK input mode, the data is shifted in the receiving buffer 1 when SCLK input becomes active while the receive interrupt flag INTES0<IRX0C>/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 (SC0BUF/SC1BUF) at the timing shown below, and INTES0<IRX0C> / INTES1<IRX1C> will be set again to generate INTRX0/1 interrupt.

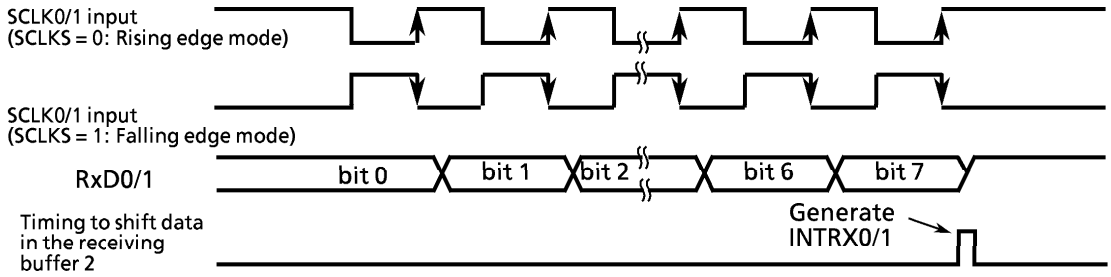


Figure 3.11 (22) Receiving Operation in I/O Interface Mode (SCLK Input Mode)

Note : For data receiving, the system must be placed in the receive enable state (SC0MOD/SC1MOD<RXE> = "1")

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

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).

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   }  Select P80 as the TxD0 pin.
P8FC    ← X X - X - - X 1   }
SC0MOD  ← X 0 - X 0 1 0 1      Set 7-bit UART mode.
SC0CR   ← X 1 1 X X X 0 0      Add an even parity.
BR0CR   ← 0 X 1 0 0 1 0 1      Set transfer rate at 2400 bps.
TRUN    ← 1 X - - - - - -      Start the prescaler for the baud rate generator.
INTES0  ← 1 1 0 0 - - - -      Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF  ← * * * * * * * *      Set data for transmission.
```

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

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

8-bit UART mode can be specified by setting SC0MOD<SM1,0> / 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> / SC1CR<PE>, and 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 -       Select P81 (RxD0) as the input pin.
SC0MOD  ← - 0 1 X 1 0 0 1       Enable receiving in 8-bit UART mode.
SC0CR   ← X 0 1 X X X 0 0       Add an odd parity.
BR0CR   ← 0 X 0 1 0 1 0 1       Set transfer rate at 9600 bps.
TRUN    ← 1 X - - - - - -       Start the prescaler for the baud rate generator.
INTES0  ← - - - - 1 1 0 0       Enable 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

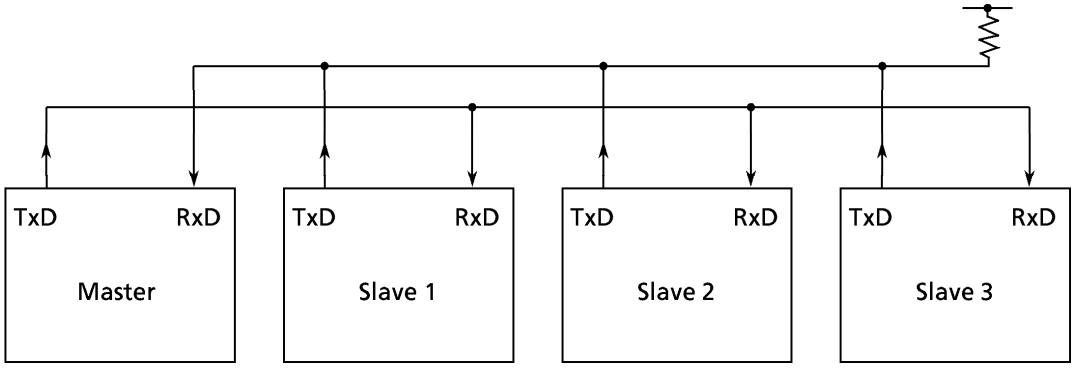
(4) Mode 3 (9-bit UART Mode)

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>/SC1MOD <TB8>, while in receiving it is stored in SC0CR<RB8>/SC1CR<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 INTRX0 / INTRX1 occurs only when<RB8> = 1.



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

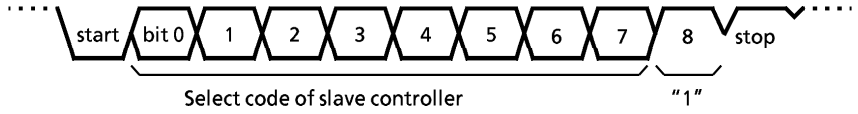
Figure 3.11 (23) Serial Link Using Wake-Up Function

Protocol

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

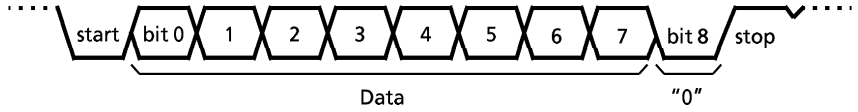
② Set SC0MOD<WU>/ SC1MOD<WU> bit of each slave controller to "1" to enable data receiving.

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



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

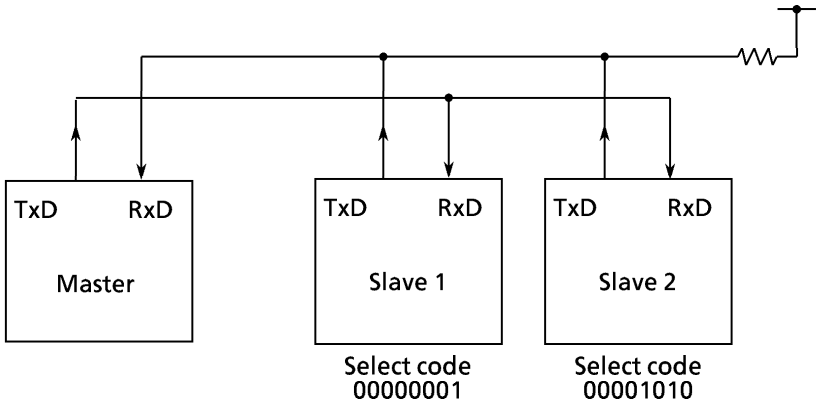
⑤ The master controller transmits data to the specified slave controller whose SC0MOD<WU>/ SC1MOD<WU> bit is cleared to "0". The MSB (bit 8)<TB8> is cleared to "0".



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 ϕ1 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
P8FC    ← X X - X - X X 1     } Select P80 as TxD0 pin and P81 as RxD0 pin.
INTES0  ← 1 1 0 0 1 1 0 1     Enable INTTX0 and set the interrupt level 4.
                              Enable INTTX0 and set the interrupt level 5.
SC0MOD  ← 1 0 1 0 1 1 1 0     Set φ1 as the transmission clock in 9-bit UART mode.
SC0BUF  ← 0 0 0 0 0 0 0 1     Set the select code for slave controller 1.
```

INTTX0 interrupt

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

- Setting the slave controller 2

Main

```
P8CR    ← X X - - - - 0 1
P8FC    ← X X - X - X X 1     } Select P81 as RxD0 pin and P80 as TxD0 pin (open drain output).
ODE     ← X X X X X X - 1
INTES0  ← 1 1 0 1 1 1 1 0     Enable INTRX0 and INTTX0.
SC0MOD  ← 0 0 1 1 1 1 1 0     Set <WU> to "1" in the 9-bit UART transmission mode
                              with transfer clock φ1 (fc/2).
```

INTRX0 interrupt

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

3.12 Analog / Digital Converter

TMP95C061B contains an 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. 4-channel analog input pins (AN3 to AN0) are shared by input-only port P9 and so can be used as input port.

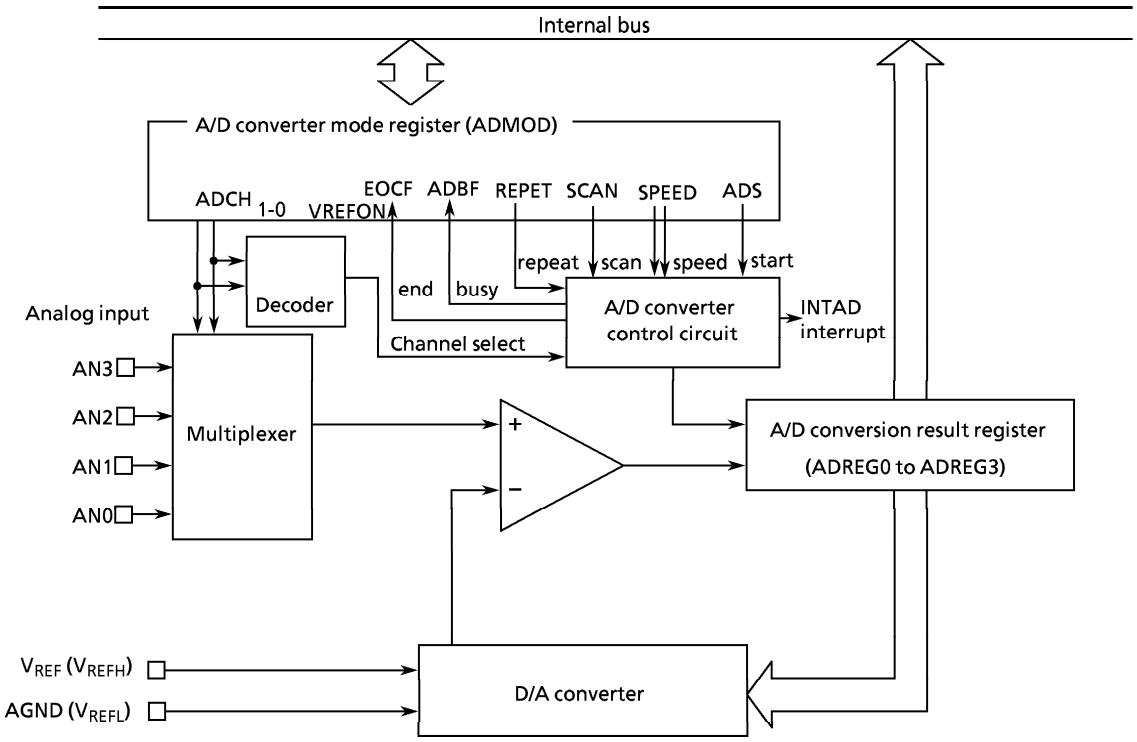


Figure 3.12 (1) Block Diagram of A/D Converter

Note1 : 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.

Note2 : 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. ADMOD<ADCS> set the "0".
The ladder resister between V_{REF} (V_{REFH}) – AGND (V_{REFL}) cannot be disconnected internally.

ADMOD (006DH)

	7	6	5	4	3	2	1	0
bit Symbol	EOCF	ADBF	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	A/D Conversion End Flag 1: END	A/D Conversion BUSY Flag 1: BUSY	Repeat Mode 0: Single Mode 1: Repeat Mode	Scan Mode 0: Conversion Channel Fixed 1: Channel Scan Mode	A/D Conversion Speed 0: High Speed Mode 1: Low Speed Mode	A/D Conversion Start 1: START Always read as "0"	Analog Input Channel Selection	

Analog Input Channel Selection

	Normal	Scan Mode
00	AN0	AN0
01	AN1	AN0→AN1
10	AN2	AN0→AN1→AN2
11	AN3	AN0→AN1→AN2→AN3

A/D Conversion Start

0	—
1	Start Conversion

Note: Always "0" when data is read

A/D Conversion Speed

0	A/D High Speed Conversion Mode : 160 States = 12.8 μs (at 25 MHz)
1	A/D Low Speed Conversion Mode : 320 States = 25.6 μs (at 25 MHz)

A/D Scan Mode

0	Fixed Channel Mode
1	Channel Scan Mode

A/D Repeat Mode

0	Single Mode
1	Repeat Mode

A/D BUSY flag

0	NOT BUSY
1	BUSY

A/D End Flag

0	NOT END
1	END

Figure 3.12 (2) A/D Control Register

ADREG0L (0060H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN0 are stored.							

ADREG0H (0061H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of A/D result for AN0 are stored.							

ADREG1L (0062H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of A/D result for AN1 are stored.							

ADREG1H (0063H)

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

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 the low analog reference voltage is applied to AGND (V_{REFL}) pin.

The reference voltage between VREF (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 <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 <ADCH1,0>, such as AN0→AN1, AN0→AN1→AN2, 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 conversion starts, conversion busy flag ADMOD<ADBF> which indicates "conversion is in progress" will be set to "1".

Don't set ADMOD<ADS> to "1" during a conversion. When ADMOD<ADS> is written "1" during A/D conversion, the conversion is finished halfway and new A/D conversion is started. In the case of conversion channel scan mode, the conversion channel returns to channel 0 and new conversion is started.

(4) A/D Conversion Mode

Both fixed A/D conversion channel mode and 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<SPEED1:0> 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", ADMOD1 <ADBF> flag will be reset to "0", and INTAD interrupt will be enabled when A/D conversion of specified channel ends in fixed conversion channel mode or when A/D conversion of the last channel ends in channel scan mode.

- A/D conversion repeat mode

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

  When A/D conversion changes to the halt state of IDLE and STOP mode, even if in A/D converting state, A/D converter immediately stops the operation. After releasing the halt, the conversion does not restart.

(7) Storing the A/D Conversion Result

The results of A/D conversion are stored in ADREG 0 to 3 register for each channel. In repeat mode, the registers are up dated when ever conversion ends.

ADREG 0 to 3 are read-only registers.

(8) Reading the A/D Conversion Result

The results of A/D conversion are stored in ADREG 0 to 3 registers.

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

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 Detection Timer)

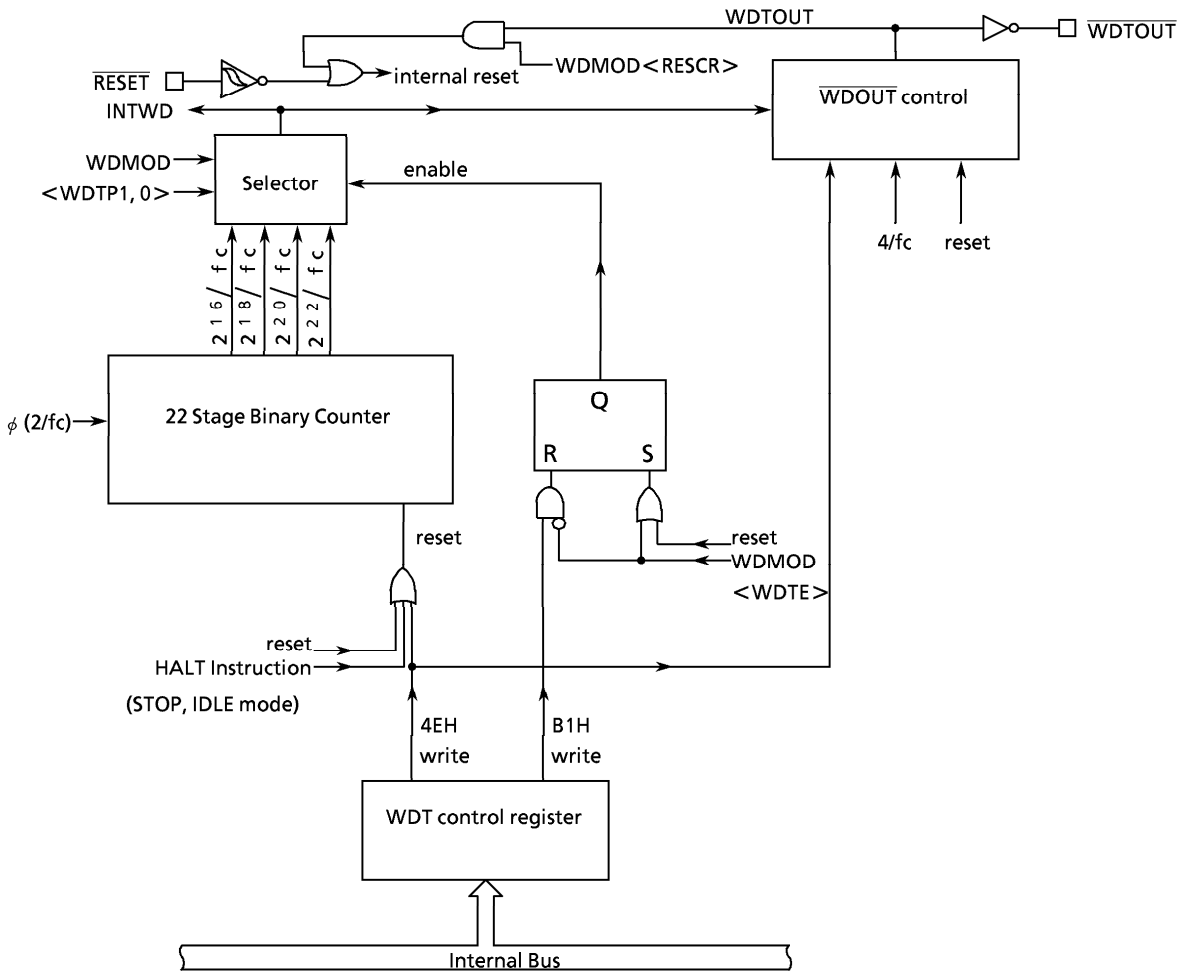
TMP95C061B contains a watchdog timer to detect a runaway CPU condition.

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 $\overline{\text{WDTOUT}}$ 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

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



Figuer 3.13 (1) Block Diagram of Watchdog Timer

The watchdog timer is a 22-stage binary counter which uses ϕ(2/fc) 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<WDTD1, 0> register generates a watchdog interrupt, and outputs watchdog timer out when an overflow occurs.

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

(Example)

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

In other words, the $\overline{\text{WDTOUT}}$ keeps outputting "0" until the clear code is written.

The watchdog timer out pin can also be connected to the reset pin internally. In this case, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs 0 at 8 to 20 states (640 ns to 1.6 μs @ fc= 25 MHz) and resets itself.

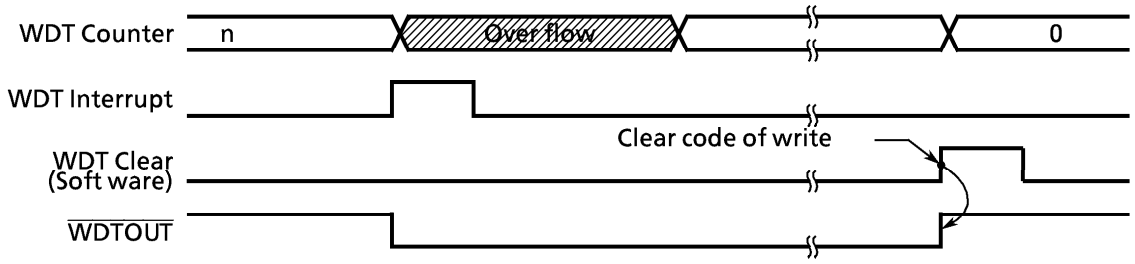


Figure 3.13 (2) Normal Mode

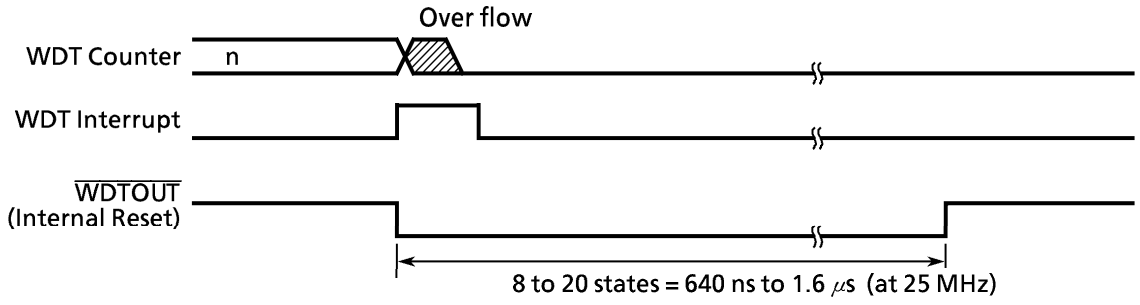


Figure 3.13 (3) Reset Mode

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 $2^{16}/f_{SYS}$ is set. (The number of states is approx. 32,768.)

② Watchdog timer enable/disable control register <WDTE>

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

To disable, it is necessary to 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 of binary counter the watchdog timer function.

- Disable control

  By writting the disable code (B1H) in this WDCR register after clearing WDMOD<WDTE> to "0", the watchdog timer can be disabled. However, the binary counter continues its operation also after the watchdog timer was disabled.

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

- Enable control

  Set WDMOD<WDTE>to "1".

  Clear the binary counter before setting the watchdog timer enable. The binary counter continues to count up also after setting the watchdog timer disable, so if the watchdog timer is set enable without clearing the binary counter, the watchdog timer out ($\overline{\text{WDTOUT}}$) signal is output at a different timing from the detecting time which is selected by WDMOD<WDTP1, 0> register.

- Watchdog timer clear control

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

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

The binary counter is cleared when the clear code is written, when reset, and when the device enters standby state in IDLE or STOP mode by execution of the HALT instruction.

In the case of using the watchdog timer as an interval timer, clear the binary counter in the watchdog timer interrupt sequence. If the binary counter is not cleared in the interrupt sequence, it is cleared by an overflow after it counted up until 22-stage.

WDMOD (006EH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 1: Enable	Select detecting time 00: $2^{16}/f_{SYS}$ 01: $2^{18}/f_{SYS}$ 10: $2^{20}/f_{SYS}$ 11: $2^{22}/f_{SYS}$		Warm - Up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDEL2 mode care		1: Internally connects WDT out to the reset pin	1: Drive the pin even in STOP mode

DRVE (explanation by standby function)

Watchdog timer out control

0	—
1	Connects WDT out to a reset

Select the standby mode HALT instruction

00	RUN mode (Only the CPU stops)
01	STOP mode (All circuits stop)
10	IDLE mode (Only the oscillator operates)
11	Don't care

Select the warm-up time on return from STOP mode

0	2^{14}/fc (approx. 655 μs at 25 MHz)
1	2^{16}/fc (approx. 2.6 ms at 25 MHz)

Select the detecting time of watchdog timer

00	2^{15}/fc (approx. 2.6 ms at 25 MHz)
01	2^{17}/fc (approx. 10.5 ms at 25 MHz)
10	2^{19}/fc (approx. 41.9 ms at 25 MHz)
11	2^{21}/fc (approx. 168 ms at 25 MHz)

Watchdog timer Enable / Disable control

0	Disable
1	Enable

Figure 3.13 (4) Watchdog Timer Mode Register

WDCR (006FH)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
After reset	–							
Function	B1H : WDT disable code 4EH : WDT clear code							

Disable / clear WDT

B1H	Disable code
4EH	Clear code
Others	—

Figure 3.13 (5) Watchdog Timer Control Register

3.13.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1, 0>register and outputs a low level signal. The watchdog timer must be zero-cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (runaway) due to the INTWD Interrupt and it is possible to return to normal operation by an anti-mulfunction program. By connecting the watchdog timer out pin to peripheral devices' resets, a CPU malfunction can also be acknowledged to other devices.

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE and STOP modes. In the RUN mode, the watchdog timer is enabled. When the bus is released ($\overline{BUSAK}$='L'), WDT continues counting up.

However, the function can be disabled when entering the RUN 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 IDLE 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 TMP95C061B supports a bus request pin ($\overline{\text{BUSRQ}}$: also used as P53) and a bus acknowledge pin ($\overline{\text{BUSAK}}$: also used as P54). Set these pins using P5CR and P5FC.

3.14.1 Operation description

When 0 is input to the $\overline{\text{BUSRQ}}$ pin, the TMP95C061B acknowledges a bus request. When the current bus cycle ends, the TMP95C061B sets the address bus (A23 to A0) and bus control signals ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{HWR}}$, R/$\overline{\text{W}}$, $\overline{\text{CS0}}$ to $\overline{3}$) to high, then sets these signals and the data bus (D15 to D0) output buffer to off, and sets the $\overline{\text{BUSAK}}$ 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 TMP95C061B 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 bis 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 $\overline{\text{RD}}$ $\overline{\text{WR}}$	—	First sets all bits to high, then sets them to high impedance.
P52 ($\overline{\text{HWR}}$) P55 (R/$\overline{\text{W}}$)	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 ($\overline{\text{CS0}}$) P61 ($\overline{\text{CS1}}$) P62 ($\overline{\text{CS2}}$) P63 ($\overline{\text{CS3}}$)	No status change.	First sets all bits to high, then sets them to high impedance.

For P63 ($\overline{\text{CAS}}$), P64 ($\overline{\text{RAS}}$), P65 ($\overline{\text{REFOUT}}$), see the description of "3.7 (5) Bus release mode".

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V cc	– 0.5 to 6.5	V
Input Voltage	V IN	– 0.5 to Vcc + 0.5	V
Output Current (total)	Σ IOL	120	mA
Output Current (total)	Σ IOH	– 120	mA
Power Dissipation (Ta = 70°C)	P D	600	mW
Soldering Temperature (10 s)	T SOLDER	260	°C
Storage Temperature	T STG	– 65 to 150	°C
Operation Temperature	T OPR	– 20 to 70	°C

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

4.2 DC characteristics

Vcc = 5 V ± 10%, TA = – 20 to 70°C (8 to 25 MHz)

(Typical values are for Ta = 25°C and Vcc = 5 V unless otherwise noted)

Parameter	Symbol	Test Condition	Min	Max	Unit
Input Low Voltage (D0 to 15) P5, P7, P8, P9, PA, PB RESET,NMI,INT0 (PB7) EA, AM8 / 16 X1	V IL V IL1 V IL2 V IL3 V IL4		– 0.3 – 0.3 – 0.3 – 0.3 – 0.3	0.8 0.3Vcc 0.25Vcc 0.3 0.2Vcc	V V V V V
Input High Voltage (D0 to 15) P5, P7, P8, P9, PA, PB RESET, NMI, INT0 (PB7) EA, AM8 / 16 X1	V IH V IH1 V IH2 V IH3 V IH4		2.2 0.7Vcc 0.75Vcc Vcc – 0.3 0.8Vcc	Vcc + 0.3 Vcc + 0.3 Vcc + 0.3 Vcc + 0.3 Vcc + 0.3	V V V V V
Output Low Voltage	V OL	I OL = 1.6 mA		0.45	V
Output High Voltage	V OH V OH1 V OH2	I OH = – 400 μA I OH = – 100 μA I OH = – 20 μA	2.4 0.75Vcc 0.9Vcc		V V V
Darlington Drive Current (8 Output Pins max.)	I DAR	V EXT = 1.5 V R EXT = 1.1 kΩ	– 1.0	– 3.5	mA
Input Leakage Current Output Leakage Current	I LI I LO	0.0 ≦ Vin ≦ Vcc 0.2 ≦ Vin ≦ Vcc – 0.2	0.02 (Typ) 0.05 (Typ)	± 5 ± 10	μA μA
Operating Current (RUN) IDLE STOP (Ta = – 20 to 70°C) STOP (Ta = 0 to 50°C)	I cc	fc = 25 MHz 0.2 ≦ Vin ≦ Vcc – 0.2 0.2 ≦ Vin ≦ Vcc – 0.2	37 (Typ) 3.5 (Typ) 0.5 (Typ)	50 10 50 10	mA mA μA μA
Power Down Voltage (at STOP)	V STOP	V IL2 = 0.2 Vcc, V IH2 = 0.8 Vcc	2.0	6.0	V
RESET Pull Up Resistance	R RST		50	150	kΩ
Pin Capacitance	C IO	fc = 1 MHz		10	pF
Schmitt Width RESET, NMI, INT0 (PB7)	V TH		0.4	1.0 (Typ)	V
Pull Up Resistance	R K		50	150	kΩ

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

4.3 AC Electrical Characteristics

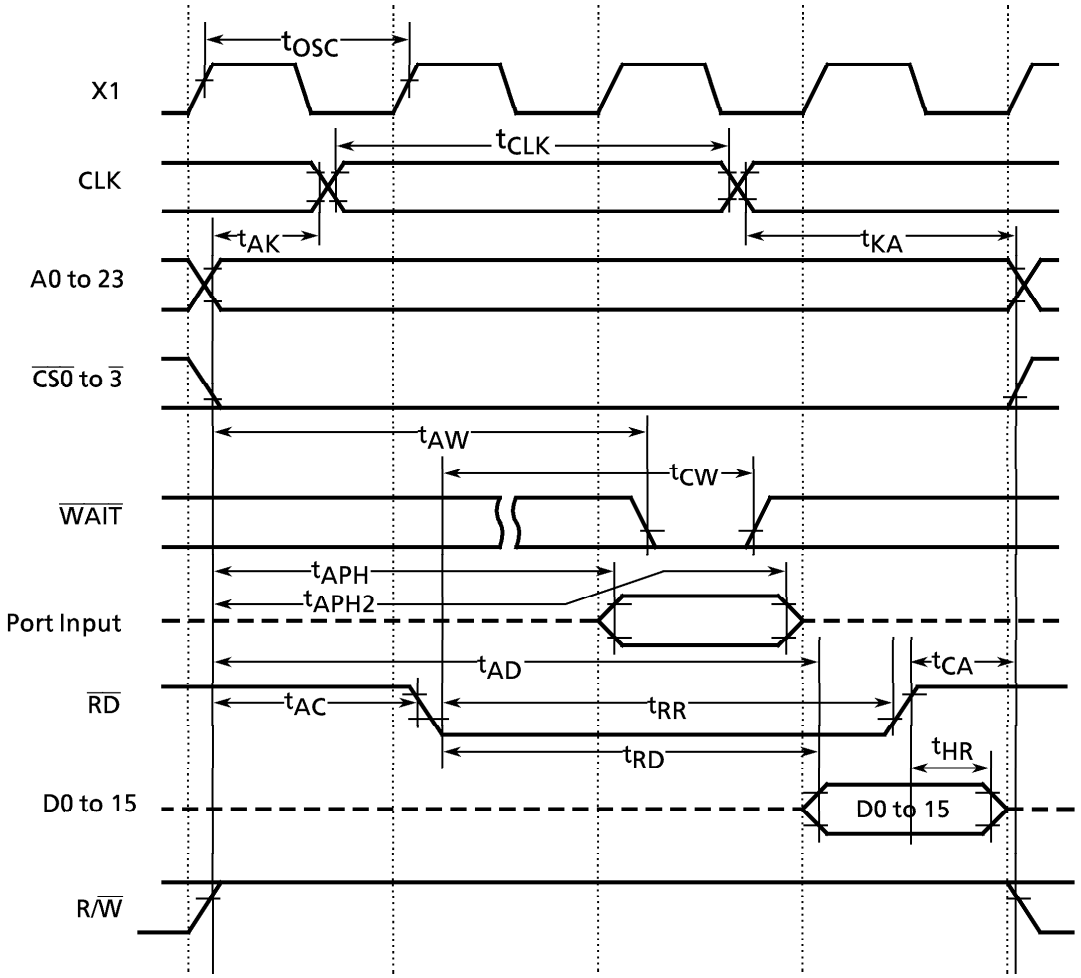
Vcc = 5 V ± 10%, TA = – 20 to 70°C
(8 MHz to 25 MHz)

No.	Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Osc. Period (= x)	t_{OSC}	40	125	50		40		ns
2	CLK width	t_{CLK}	2x – 40		60		40		ns
3	A0 to 23 Valid → CLK Hold	t_{AK}	0.5x – 20		5		0		ns
4	CLK Valid → A0 to 23 Hold	t_{KA}	1.5x – 60		5		0		ns
5	A0 to 23 Valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{AC}	1.0x – 20		30		20		ns
6	$\overline{RD}$ / $\overline{WR}$ rise → A0 to 23 Hold	t_{CA}	0.5x – 20		5		0		ns
7	A0 to 23 Valid → D0 to 15 input	t_{AD}		3.5x – 35		140		105	ns
8	$\overline{RD}$fall → D0 to 15 input	t_{RD}		2.5x – 40		85		60	ns
9	$\overline{RD}$ Low width	t_{RR}	2.5x – 40		85		60		ns
10	$\overline{RD}$rise → D0 to 15 Hold	t_{HR}	0		0		0		ns
11	$\overline{WR}$ Low width	t_{WW}	2.5x – 40		85		60		ns
12	D0 to 15 Valid → $\overline{WR}$rise	t_{DW}	2.0x – 40		60		40		ns
13	$\overline{WR}$ rise → D0 to 15 Hold	t_{WD}	0.5x – 10		15		10		ns
14	A0 to 23 Valid → $\overline{WAIT}$ input (1 WAIT + n mode)	t_{AW}		3.5x – 90		85		50	ns
15	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ Hold (1 WAIT + n mode)	t_{CW}	2.5x + 0		125		100		ns
16	A0 to 23 Valid → PORT input	t_{APH}		2.5x – 90		35		10	ns
17	A0 to 23 Valid → PORT Hold	t_{APH2}	2.5x + 50		175		150		ns
18	$\overline{WR}$ rise → PORT Valid	t_{CP}		200		200		200	ns

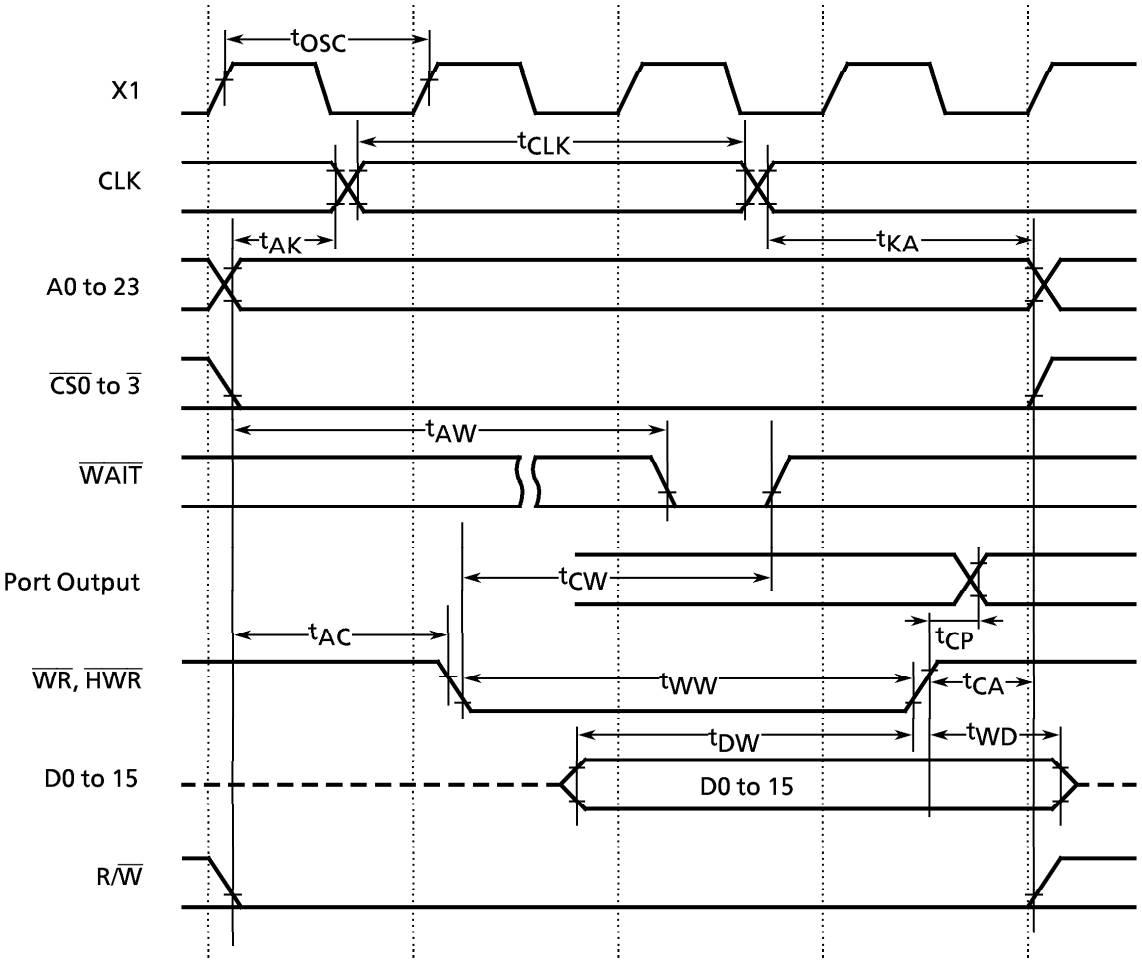
AC Measuring Conditions

- Output Level : High 2.2 V / Low 0.8 V , CL = 50 pF
  (However, D0 to D15, A0 to A23, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, CLK, $\overline{CS0}$ to $\overline{CS3}$, CL = 100 pF)
- Input Level : High 2.4 V / Low 0.45 V (D0 to D15)
  High 0.8 Vcc / Low 0.2 Vcc (exsept for D0 to D15)

(1) Read Cycle


tOSC
X1
tCLK
CLK
tAK
tKA
A0 to 23
CS0 to 3
tAW
tCW
WAIT
tAPH
tAPH2
Port Input
tAD
tCA
RD
tAC
tRR
tRD
tHR
D0 to 15
D0 to 15
R/W

(2) Write Cycle



4.4 DRAM Controller AC Electrical Characteristics

Vcc = 5 V ± 10%, TA = – 20 to 70°C
(8 MHz to 25 MHz)

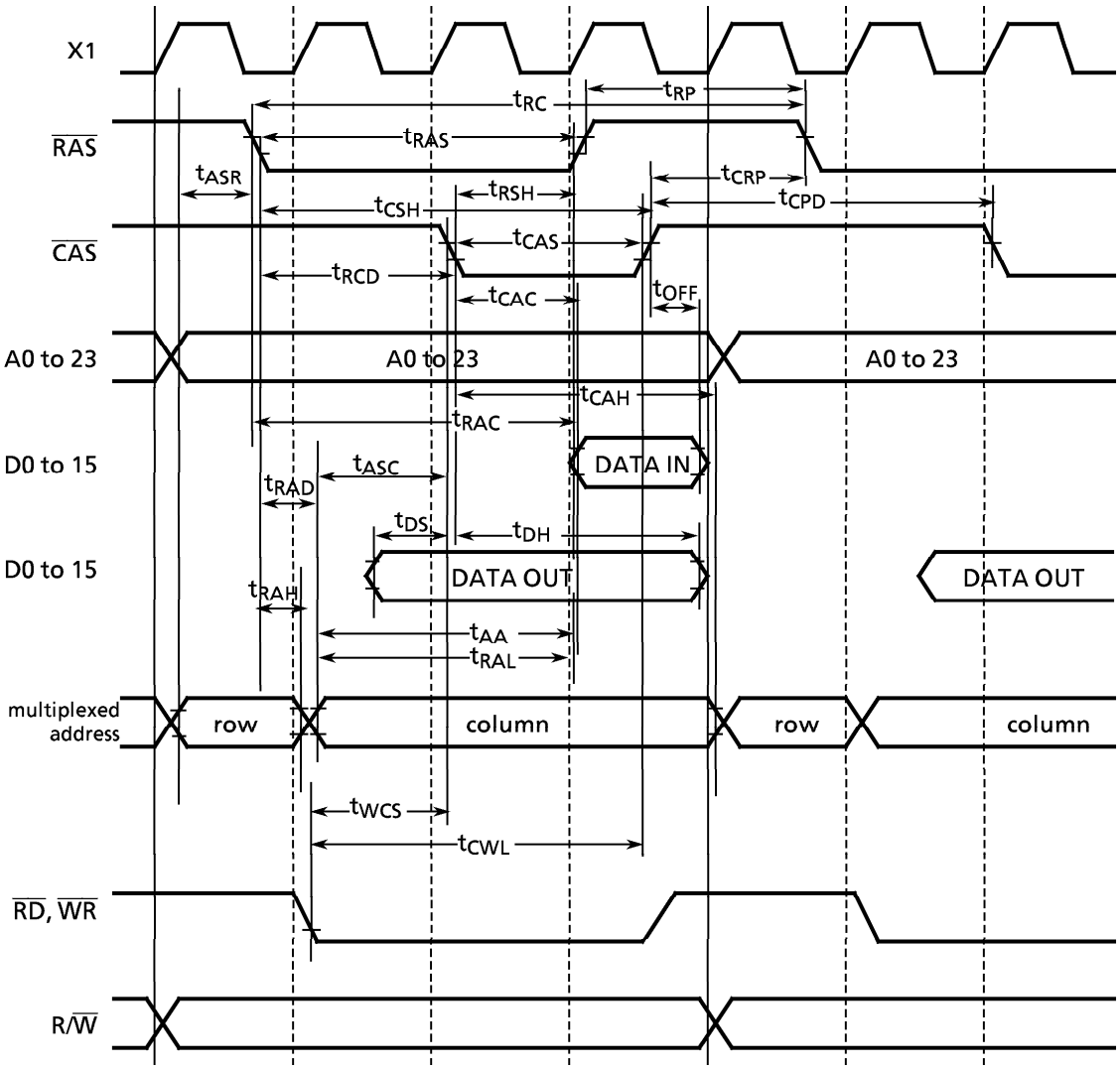
No.	Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$\overline{RAS}$ cycle time	t_{RC}	4X		200		160		ns
2	$\overline{RAS}$ access time	t_{RAC}		3X-40		110		80	ns
3	$\overline{CAS}$ access time	t_{CAC}		1.5X-35		40		25	ns
4	column address access time	t_{AA}		2.5X-55		70		45	ns
5	Input data hold time	t_{OFF}	0		0		0		ns
6	$\overline{RAS}$ precharge time	t_{RP}	1.5X-10		65		50		ns
7	$\overline{RAS}$ low pulse width	t_{RAS}	2.5X-30		95		70		ns
8	$\overline{RAS}$ hold time	t_{RSH}	1X-15		35		25		ns
9	$\overline{CAS}$ hold time	t_{CSH}	3X-35		115		85		ns
10	$\overline{CAS}$ low pulse width	t_{CAS}	1.5X-15		65		45		ns
11	$\overline{RAS}$ – $\overline{CAS}$ delay time	t_{RCD}	1.5X-40	1.5X	35	75	20	60	ns
12	$\overline{RAS}$ column address delay time	t_{RAD}	0.5X-5	0.5X + 20	20	45	15	40	ns
13	$\overline{CAS}$ – $\overline{RAS}$ precharge time	t_{CRP}	1X-35		15		5		ns
14	$\overline{CAS}$ precharge time	t_{CPD}	2.5X-35		90		65		ns
15	Low address setup time	t_{ASR}	0.5X-15		10		5		ns
16	Low address hold time	t_{RAH}	0.5X-5		20		15		ns
17	Column address setup time	t_{ASC}	1X-25		25		15		ns
18	Column address hold time	t_{CAH}	2X-35		65		45		ns
19	Column address $\overline{RAS}$ read time	t_{RAL}	2X-30		70		50		ns
20	Write command $\overline{CAS}$ read time	t_{CWL}	2.5X-35		90		65		ns
21	Data output setup time	t_{DS}	0.5X-15		10		5		ns
22	Data output hold time	t_{DH}	2X-35		65		45		ns
23	Write command setup time	t_{WCS}	1X-30		20		10		ns
24	$\overline{CAS}$ hold time	t_{CHR}*1	2X-50		50		30		ns
25	$\overline{RAS}$ precharge $\overline{CAS}$ active time	t_{RPC}*	1.5X-30		45		30		ns
26	$\overline{CAS}$ setup time	t_{CSR}*	0.5X-10		15		10		ns
27	$\overline{RAS}$ precharge time	t_{RPS}*2	4X-20		180		140		ns
28	$\overline{CAS}$ hold time	t_{CHS}*2	0		0		0		ns
29	refresh setup time	t_{CFL}*	1X-5		45		35		ns
30	refresh hold time	t_{CFH}*	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

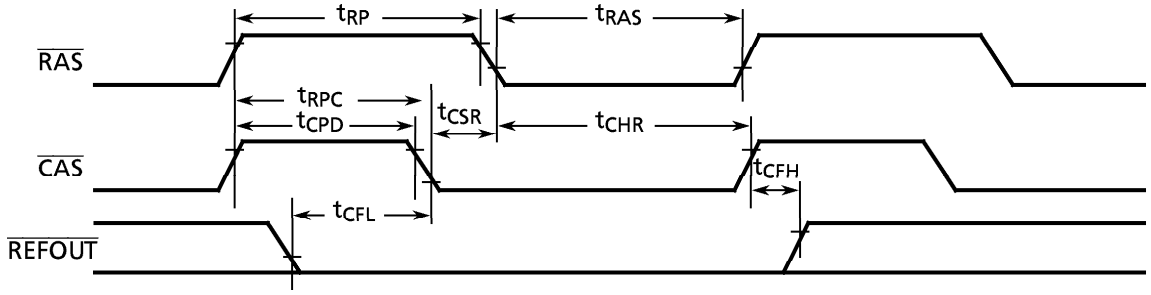
AC Measuring Conditions

- Output Level : High 2.2 V / Low 0.8 V , CL = 50 pF
  (However CL = 100pF for D0 to D15, A0 to A23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$)
- Input Level : High 2.4 V / Low 0.45 V (D0 to D15)
  High 0.8 Vcc / Low 0.2 Vcc (Except for D0 to D15)

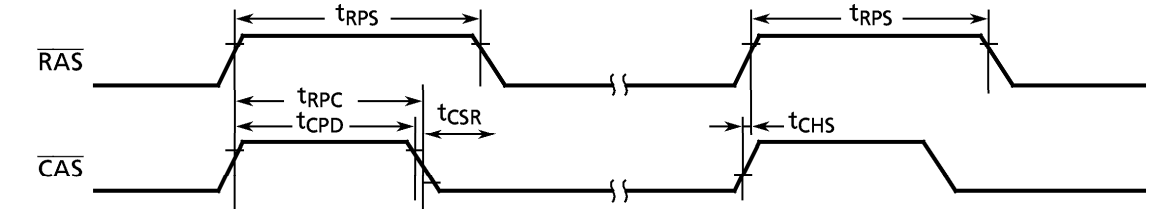
(1) Read/Write Access Cycle



(2) $\overline{CAS}$ before $\overline{RAS}$ interval refresh cycle



(3) $\overline{CAS}$ bifore $\overline{RAS}$ self-refresh cycle



4.5 A/D Conversion Characteristics

Vcc = 5V ± 10%, TA = – 20 to 70°C (8 to 25 MHz)

Parameter		Symbol	Min	Typ.	Max	Unit
Analog reference voltage		V_{REF} (V_{REFH})	Vcc – 1.5		Vcc	V
Analog reference voltage		A_{GND} (V_{REFL})	Vss		Vss	
Analog input voltage range		V_{AIN}	Vss		Vcc	
Anlog current for analog reference voltage		I_{REF}		0.5	1.5	mA
4 ≦ fc ≦ 16 MHz	slow mode	Error (Quantize error of ± 0.5 LSB not included)		± 1.5	± 4.0	LSB
	fast mode			± 3.0	± 6.0	
16 < fc ≦ 25 MHz	slow mode			± 1.5	± 4.0	
	fast mode			± 4.0	± 8.0	

4.6 Serial Channel Timing

(1) SCLK Input Mode (I/O Interface Mode)

Vcc = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	16X		0.8		0.64		μs
Output Data → Rising edge of SCLK	t_{OSS}	t_{SCY}/2 − 5X − 50		100		70		ns
SCLK rising edge→ Output Data hold	t_{OHS}	5X − 100		150		100		ns
SCLK rising edge→ Input Data hold	t_{HSR}	0		0		0		ns
SCLK rising edge→ effective data input	t_{SRD}		t_{SCY} − 5X − 100		450		340	ns

(2) SCLK Output Mode (I/O Interface Mode)

Vcc = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	16X	8192X	0.8	409.6	0.64	327.6	μs
Output Data → SCLK rising edge	t_{OSS}	t_{SCY} − 2X − 150		550		410		ns
SCLK rising edge→ Output Data hold	t_{OHS}	2X − 80		20		0		ns
SCLK rising edge→ Input Data hold	t_{HSR}	0		0		0		ns
SCLK rising edge→ effective data input	t_{SRD}		t_{SCY} − 2X − 150		550		410	ns

(3) SCLK0 Input Mode (UART Mode)

Vcc = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	4X + 20		220		180		ns
SCLK Low level Pulse width	t_{SCYL}	2X + 5		105		85		ns
SCLK High level Pulse width	t_{SCYH}	2X + 5		105		85		ns

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

Vcc = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

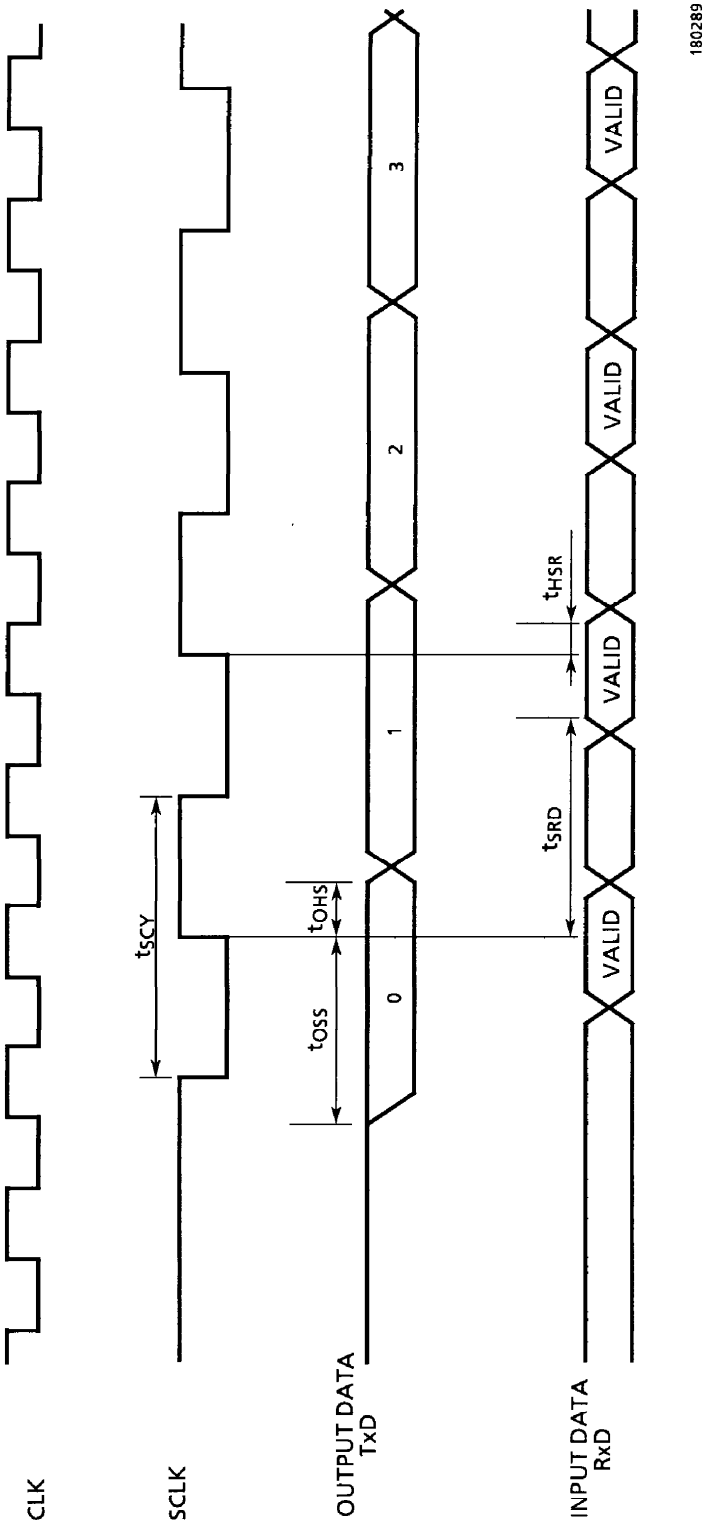
Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock Cycle	t_{VCK}	8X + 100		500		420		ns
Low level clock Pulse width	t_{VCKL}	4X + 40		240		200		ns
High level clock Pulse width	t_{VCKH}	4X + 40		240		200		ns

4.8 Interrupt Operation

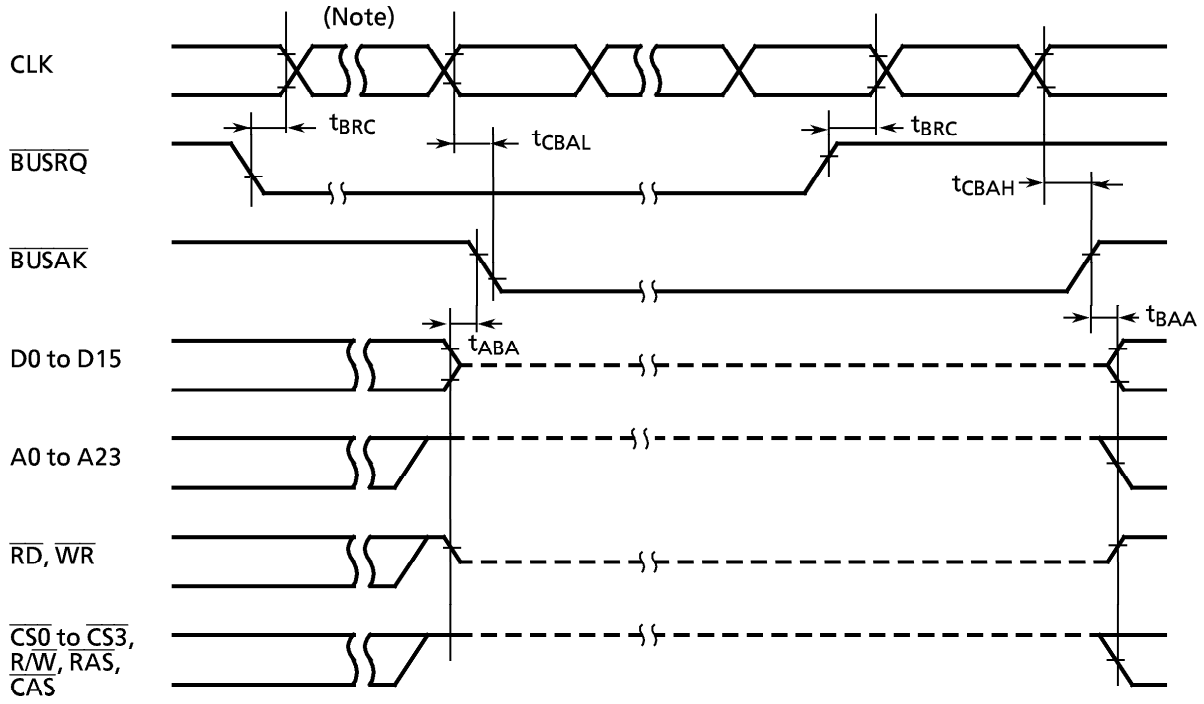
Vcc = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 Low level Pulse width	t_{INTAL}	4X		200		160		ns
$\overline{NMI}$, INT0 High level Pulse width	t_{INTAH}	4X		200		160		ns
INT4 to INT7 Low level Pulse width	t_{INTBL}	8X + 100		500		420		ns
INT4 to INT7 High level Pulse width	t_{INTBH}	8X + 100		500		420		ns

4.9 Timing Chart for I/O Interface Mode



4.10 Timing Chart for Bus Request ($\overline{BUSRQ}$) / Bus Acknowledge ($\overline{BUSAK}$)



Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{BUSQR}$ set-up time for CLK	t_{BRC}	120		120		120		ns
CLK→ $\overline{BUSAK}$ falling edge	t_{CBAL}		2.0x + 120		220		200	ns
CLK→ $\overline{BUSAK}$ rising edge	t_{CBAH}		0.5x + 40		65		60	ns
Floating time to $\overline{BUSAK}$ fall	t_{ABA}	0	80	0	80	0	80	ns
Floating time to $\overline{BUSAK}$ rise	t_{BAA}	0	80	0	80	0	80	ns

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

4.11 Typical Characteristics

V_{CC} = 5V, Ta = 25°C, unless otherwise noted.

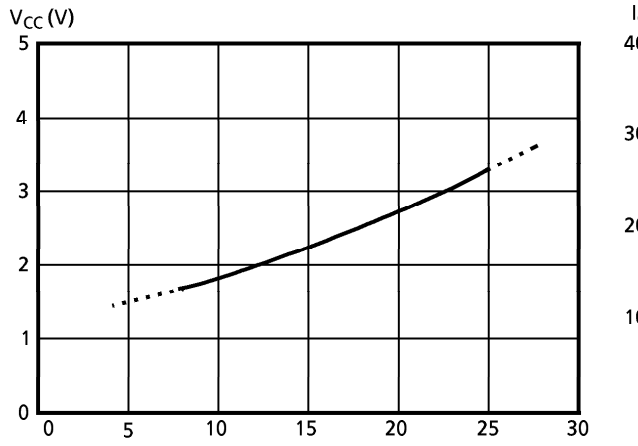


Figure 4.1 V_{CC} – f_{OSC} TYPICAL CURVE

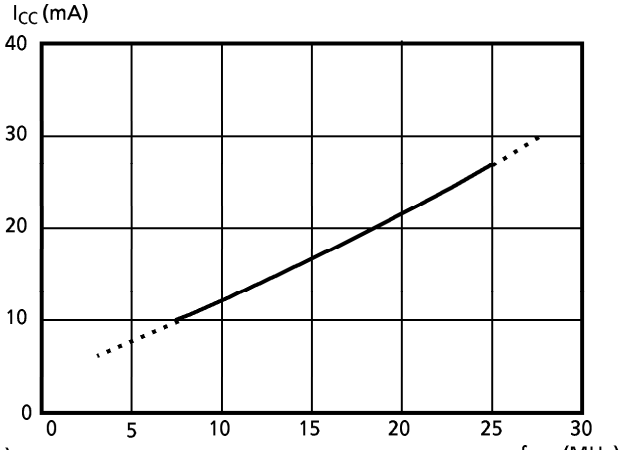


Figure 4.2 I_{CC} – f_{OSC} TYPICAL CURVE

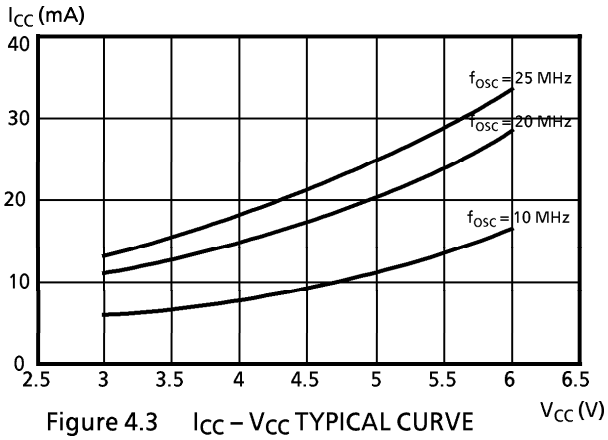


Figure 4.3 I_{CC} – V_{CC} TYPICAL CURVE

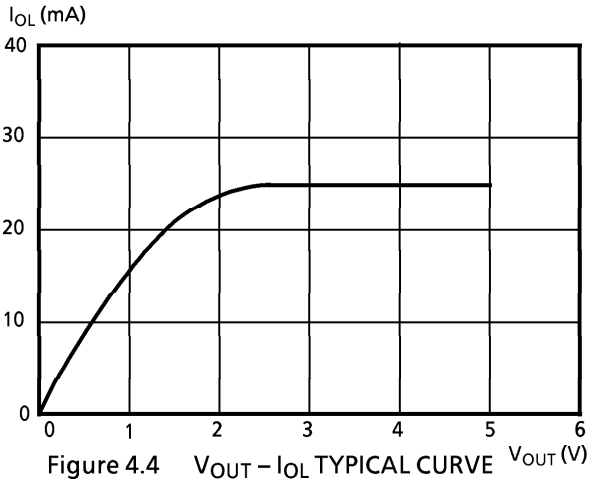


Figure 4.4 V_{OUT} – I_{OL} TYPICAL CURVE

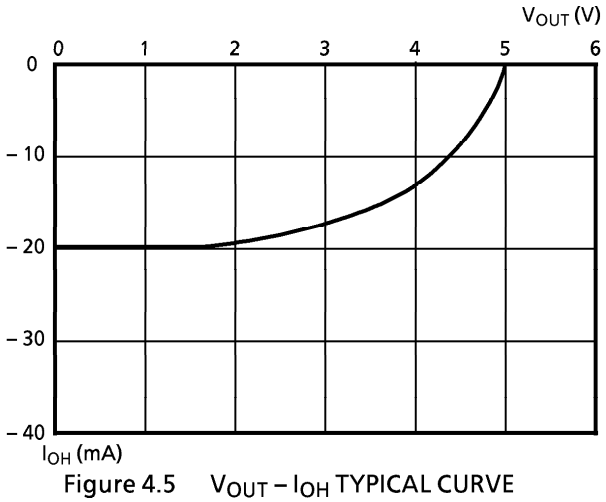


Figure 4.5 V_{OUT} – I_{OH} TYPICAL CURVE

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

Table5 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		28H	T23MOD	48H	T5MOD	68H	B0CS
9H	P2FC	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	P5CR	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H	P5FC	31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	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	P8CR	3AH	T45CR	5AH	DREFCR	7AH	INTETC23
1BH	P8FC	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
P5	PORT5	0DH			P55	P54	P53	P52		RDE
					* R/W					
					Input mode (Pulled-up)				–	
					1	1	1	1		1
P6	PORT6	12H			P65	P64	P63	P62	P61	P60
					R/W					
					Output mode					
					1	1	1	0	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
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
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 $\overline{RD}$ strobe form $\overline{RD}$ pin (for PSRAM), even when the internal address is accessed.
If "RDE" remains "1", the $\overline{RD}$ 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 / EX / ADD / ADC / SUB / SBC / INC / DEC / RLC / RRC / RL / RR / SLA / SRA / SLL / SRL / RLD / RRD / AND / OR / XOR Instruction)

* R/W ; RMW instructions are prohibited for controlling ON/OFF of the pull-up resistor.

(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
						0 : IN	1 : OUT			
P2FC	PORT2 Function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			1	1	1	1	1	1	1	1
						0 : PORT	1 : A23 to A16			
P5CR	PORT5 Control	10H (Prohibit RMW)			P55C	P54C	P53C	P52C		
					W					
					0	0	0	0		
						0 : IN	1 : OUT			
P5FC	PORT5 Function	11H (Prohibit RMW)			P55F	P54F	P53F	P52F		
					W					
					0	0	0	0		
					0 : PORT 1 : R/$\overline{W}$	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$	0 : PORT 1 : $\overline{HWR}$		
P6FC	PORT6 Function	15H (Prohibit RMW)			P65F	P64F	P63F	P62F	P61F	P60F
					W					
					0	0	0	0	0	0
						0 : PORT	1 : $\overline{CS/CAS}$, $\overline{RAS}$, $\overline{REFOUT}$			
P7CR	PORT7 Control	16H (Prohibit RMW)	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0	0	0	0	0	0	0	0
						0 : IN	1 : OUT			
P7FC	PORT7 Function	17H (Prohibit RMW)	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
			W							
			0	0	0	0	0	0	0	0
			0 : PORT		1 : PG1-OUT			0 : PORT	1 : PG0-OUT	
P8CR	PORT8 Control	1AH (Prohibit RMW)			P85C	P84C	P83C	P82C	P81C	P80C
					W					
					0	0	0	0	0	0
							0 : IN	1 : OUT		
P8FC	PORT8 Function	1BH (Prohibit RMW)			P85F		P83F	P82F		P80F
					W		W	W		W
					0		0	0		0
					0 : PORT 1 : SCLK1		0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0		0 : PORT 1 : TxD0

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 1 : TO3	0 : PORT 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 1 : TO6			0 : PORT 1 : TO5	0 : PORT 1 : TO4		

(3) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
			Prescaler & Timer Run / Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							
TREG0	8 bit Timer Register 0	22H (Prohibit RMW)	–							
			W							
			Undifined							
TREG1	8 bit Timer Register 1	23H (Prohibit RMW)	–							
			W							
			Undifined							
T01 MOD	8 bit Timer 0,1 Source CLK & MODE	24H (Prohibit RMW)	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1 10 : 2^7 – 1 11 : 2^8 – 1	PWM Cycle	00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 INPUT 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TFFCR	8 bit Timer Flip-Flop Control	25H (Prohibit RMW)	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			–		0	0	–		0	0
			00 : Invert TFF3 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		1 : TFF3 Invert Enable	1: Inversion of Timer 3	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	1: Inversion of Timer 1
TREG2	8 bit Timer Register 2	26H (Prohibit RMW)	–							
			W							
			Undifined							
TREG3	8 bit Timer Register 3	27H (Prohibit RMW)	–							
			W							
			Undifined							
T23 MOD	8 bit Timer 2,3 Source CLK & MODE	28H (Prohibit RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1 PWM 10 : 2^7 – 1 Cycle 11 : 2^8 – 1		00 : TO2TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : – 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TRDC	Timer Reg. Double Buffer Control Reg.	29H							TR2DE	TR0DE
									R/W	
									0	0
									0 : Double Buffer Disable 1 : Double Buffer Enable	

Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4L	16 bit Timer Register4L	30H (Prohibit RMW)	–							
			W							
			Undefined							
TREG4H	16 bit Timer Register4H	31H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5L	16 bit Timer Register5L	32H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5H	16 bit Timer Register5H	33H (Prohibit RMW)	–							
			W							
			Undefined							
CAP1L	Capture Register1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register2H	37H	–							
			R							
			Undefined							
T4MOD	16 bit Timer 4 Source CLK & MODE	38H (Prohibit RMW)	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFF5 INV TRG 0 : TRG Disable 1 : TRG Enable		0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI4 ↑ TI5 ↑ 10 : TI4 ↑ TI4 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC4 Clear Enable	Source Clock 00 : TI4 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T4FFCR	16 bit Timer 4 Flip-Flop Control	39H (Prohibit RMW)	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
			W		R/W				W	
			–		0	0	0	0	–	
			00 : Invert TFF5 01 : Set TFF5 10 : Clear TFF5 11 : Don't care		TFF4 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF4 01 : Set TFF4 10 : Clear TFF4 11 : Don't care	
T45CR	T4, T5 Control	3AH	–				PG1T	PG0T	DB6EN	DB4EN
			R/W				R/W			
			0				0	0	0	0
			Fix at "0"				PG1 shift trigger 0 : timer2, 3 1 : timer5	PG0 shift trigger 0 : timer0, 1 1 : timer4	1 : Double Buffer Enable	

Timer Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG6L	16bit Timer Register6L	40H (Prohibit RMW)	–							
			W							
			Undifined							
TREG6H	16 bit Timer Register6H	41H (Prohibit RMW)	–							
			W							
			Undifined							
TREG7L	16 bit Timer Register7L	42H (Prohibit RMW)	–							
			W							
			Undifined							
TREG7H	16 bit Timer Register7H	43H (Prohibit RMW)	–							
			W							
			Undifined							
CAP3L	Capture Register3L	44H	–							
			R							
			Undifined							
CAP3H	Capture Register3H	45H	–							
			R							
			Undifined							
CAP4L	Capture Register4L	46H	–							
			R							
			Undifined							
CAP4H	Capture Register4H	47H	–							
			R							
			Undifined							
T5MOD	16 bit Timer 5 Source CLK & MODE	48H (Prohibit RMW)			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
					W	R/W				
					1	0	0	0	0	0
					0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI6 ↑ TI7 ↑ 10 : TI6 ↑ TI6 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC5 Clear Enable	Source Clock 00 : TI6 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T5FFCR	16 bit Timer 5 Flip-Flop Control	49H (Prohibit RMW)			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	–	
					TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(4) Pattern Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG0REG	PG0 Register	4CH (Prohibit RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0	0	0	0	Undefined			
PG1REG	PG1 Register	4DH (Prohibit RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			W				R/W			
			0	0	0	0	Undefined			
PG01CR	PG0, 1 Contorol	4EH	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0	0	0	0	0	0	0	0
			0: 8-bit write 1: 4-bit write	0: Normal Rotation 1: Reverse Rotation	0: 4-bit Step 1: 8-bit Step	PG1 trigger input enable 1: Enable	0: 8-bit write 1: 4-bit write	0: Normal Rotation 1: Reverse Rotation	0: 4-bit Step 1: 8-bit Step	PG0 trigger input enable 1: Enable

(5) Watch Dog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode	6EH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT Enable	00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warming up Time 0: 2^{14}/fc 1: 2^{16}/fc	Standby Mode 00: RUN Mode 01: STOP Mode 10: IDLE Mode 11: Don't care		1: Connect internally WDT out pin to Reset Pin	1: Drive the pin in STOP mode
WDCR	Watch Dog Timer Control Register	6FH (Prohibit RMW)	–							
			W							
			–							
			B1H: WDT Disable Code 4EH: WDT Clear Code							

(6) Serial Channel

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	50H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	51H	RB8	EVEN	PE	OERR	PERR	FERR	SCLK	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
				0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	1: Input SCLK0 pin
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Trans-mission 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: External clock (SCLK0)	
BR0CR	Baud Rate Control	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Set frequency divisor 0 to F			
SC1BUF	Serial Channel 1 Buffen	54H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
				0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	1: Input SCLK1 pin
SC1-MOD	Serial Channel 1 Mode	56H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Trans-mission data bit 8	Fix at "0"	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7bit 10: UART 8bit 11: UART 9bit		00: TO2 Trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	
BR1CR	Baud Rate Control	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (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
AD REG0L *1)	AD Result Reg 0 low	60H	ADR01	ADR00						
			R							
			Undefined		1	1	1	1	1	1
AD REG0H	AD Result Reg 0 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG1L *1)	AD Result Reg 1 low	62H	ADR11	ADR10						
			R							
			Undefined		1	1	1	1	1	1
AD REG1H	AD Result Reg 1 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG2L *1)	AD Result Reg 2 low	64H	ADR21	ADR20						
			R							
			Undefined		1	1	1	1	1	1
AD REG2H	AD Result Reg 2 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG3L *1)	AD Result Reg 3 low	66H	ADR31	ADR30						
			R							
			Undefined		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	ADBF	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
INTE-0AD	INTerrupt Enable 0 & A/D	70H (Prohibit RMW)	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5	71H (Prohibit RMW)	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE67	INTerrupt Enable 6/7	72H (Prohibit RMW)	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET01	INTerrupt Enable Timer 1/0	73H (Prohibit RMW)	INTT1 (Timer 1)				INTT0 (Timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET23	INTerrupt Enable Timer 3/2	74H (Prohibit RMW)	INTT3 (Timer 3)				INTT2 (Timer 2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET45	INTerrupt Enable Treg 5/4	75H (Prohibit RMW)	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET67	INTerrupt Enable Treg 7/6	76H (Prohibit RMW)	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	
INTES0	INTerrupt Enable Serial 0	77H (Prohibit RMW)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES1	INTerrupt Enable Serial 1	78H (Prohibit RMW)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETC 01	INTerrupt Enable TC 0/1	79H (Prohibit RMW)	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETC 23	INTerrupt Enable TC 2/3	7AH (Prohibit RMW)	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

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

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

Interrupt Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Contorol	7BH (Prohibit RMW)						I0IE W 0 1: INT0 input enable	I0LE W 0 0: INT0 edge mode 1: INT0 level mode	NMIREE W 0 1: Operate even at $\overline{\text{NMI}}$ rise edge
DMA0V	DMA 0 request Vector	7CH (Prohibit RMW)				Micro DMA0 Start vector DMA0V8 W 0	 DMA0V7 0	 DMA0V6 0	 DMA0V5 0	 DMA0V4 0
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)				Micro DMA1 Start vector DMA1V8 W 0	 DMA1V7 0	 DMA1V6 0	 DMA1V5 0	 DMA1V4 0
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)				Micro DMA2 Start vector DMA2V8 W 0	 DMA2V7 0	 DMA2V6 0	 DMA2V5 0	 DMA2V4 0
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)				Micro DMA3 Start vector DMA3V8 W 0	 DMA3V7 0	 DMA3V6 0	 DMA3V5 0	 DMA3V4 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)				B0E	–	B0BUS	B0C1	B0C0
						W	–	W	W	W
						0	–	0	0	0
						0: B0CS 1: master bit	–	0: 16 BIT 1: 8 BIT	00: 2WAIT 01: 1WAIT 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 1: master bit	–	0: 16 BIT 1: 8 BIT	00: 2WAIT 01: 1WAIT 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 1: master bit	0: 16M area 1: MREG setting	0: 16 BIT 1: 8 BIT	00: 2WAIT 01: 1WAIT 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 1: master bit	0: $\overline{CS3}$ output 1: $\overline{CAS}$ output	0: 16 BIT 1: 8 BIT	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT	
BEXCS	External CS / WAIT control register	6CH (Prohibit RMW)				–	–	BEXBUS	BEXW1	BEXW0
						–	–	W	W	
						–	–	0	0	0
						–	–	0: 16 BIT 1: 8 BIT	00: 2WAIT 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 Controller (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	RS0	RW2	RW1	RW0	RC
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Prohibit 1: Execute	Refresh cycle insertion interval 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0: Not inserted 1: inserted
DMEMCR	Memory Access Control Reg.	5BH (Prohibit RMW)	SRFC		BRM	MACM	MUXE	MUXW1	MUXW0	MAC
			W	–	R/W					
			1	–	0	0	0	0	0	0
			Self refresh 0: Execute 1: Release	–	DRAM pin Bus Release 0: Release 1: Not release	0: Normal access 1: Slow access	Address multiplex 0: Disable 1: Enable	Multiplexed address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		Memory access control 0: Disable 1: Enable

6. Port Section Equivalent Circuit Diagram

- Reading The Circuit Diagram

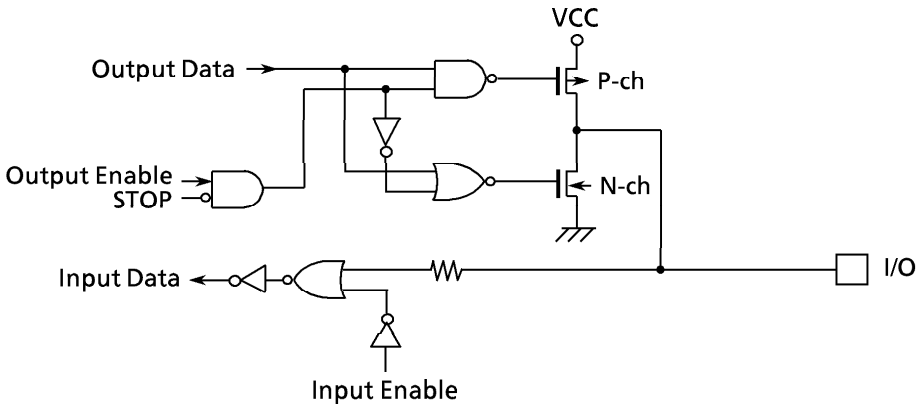
  Basically, the gate symbols written are the same as those used for the standard COMS logic IC [74HCXX] series.

  The dedicated signal is described below.

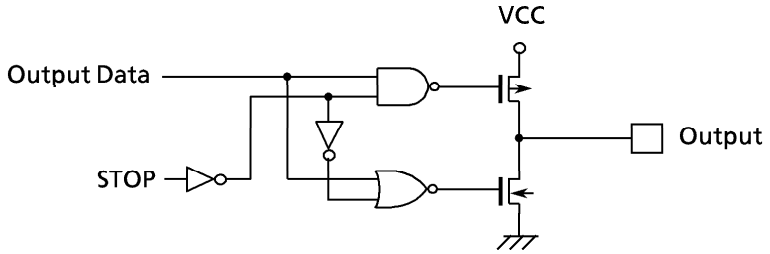
  STOP : This signal becomes active "1" when the halt mode setting register is set to the STOP mode (WDMOD<HALTM1, 0>=0,1) and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however, STOP remains at "0".

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

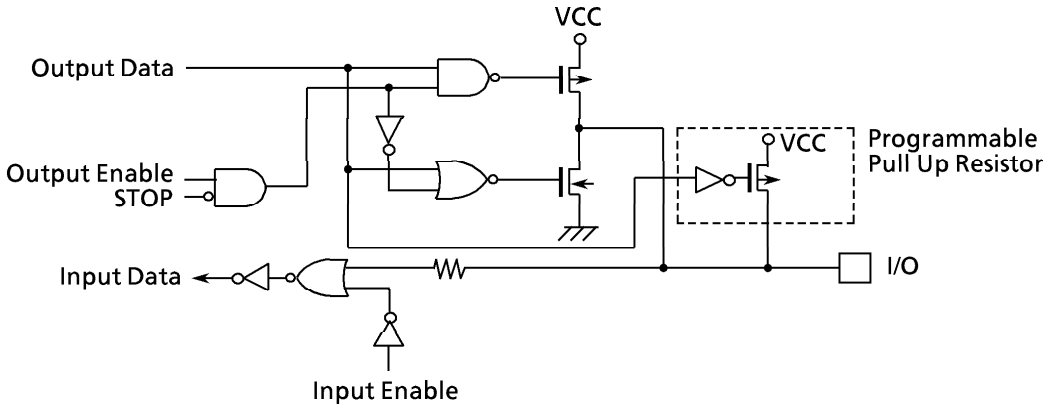
■ D0 to D7, P1 (D8 to 15)



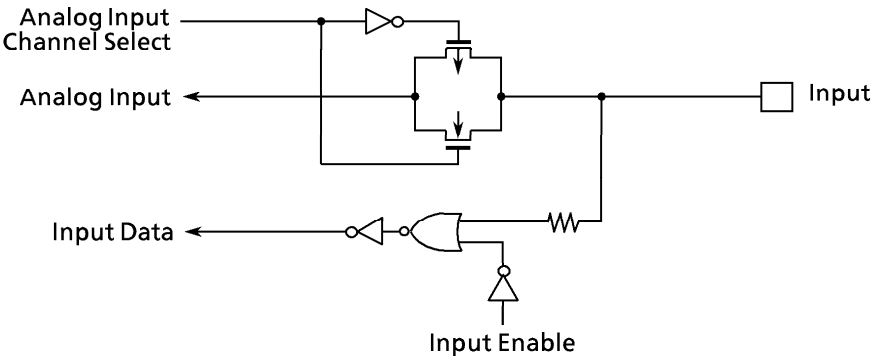
■ P2 (A16 to A23), A0 to 15, $\overline{RD}$, $\overline{WR}$, P6



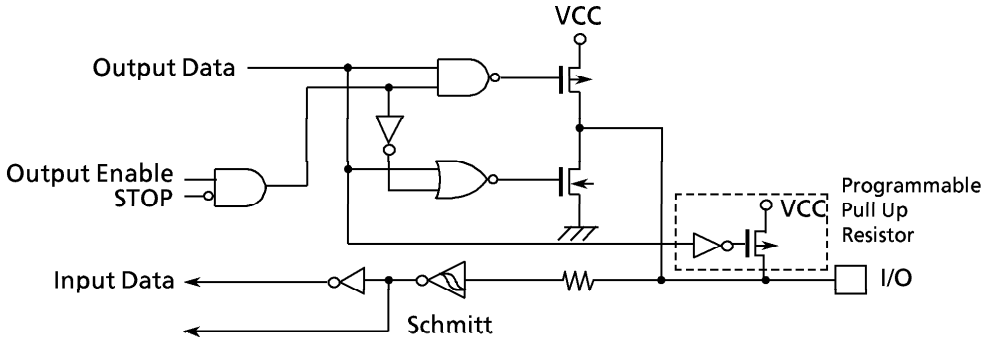
■ P52 to 55, P7, P81, P82, P84, P85, PA, PB6 to B0



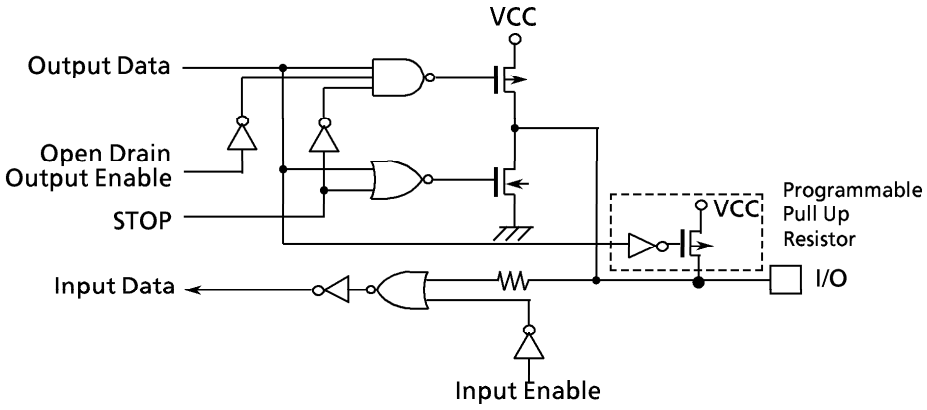
■ P9 (AN0 to 3)



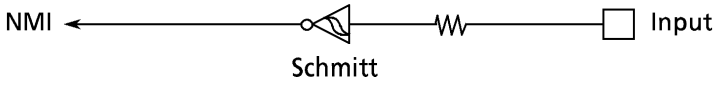
■ PB7 (INT0)



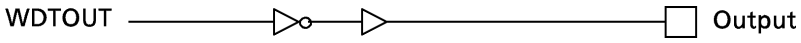
■ P80 (TXD0), P83 (TXD1)



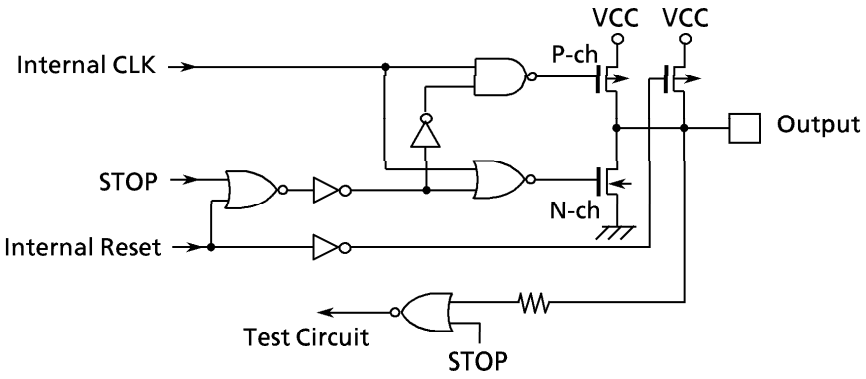
■ $\overline{\text{NMI}}$



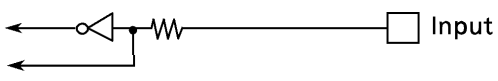
■ $\overline{\text{WDTOUT}}$



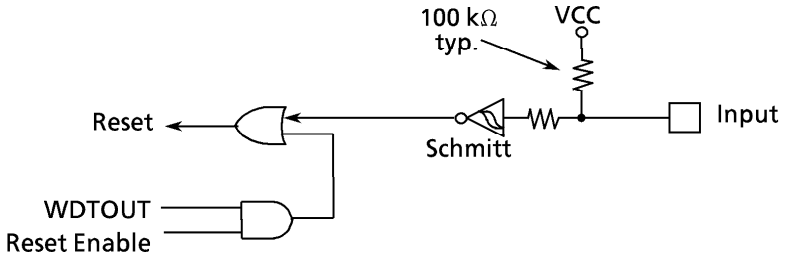
■ CLK



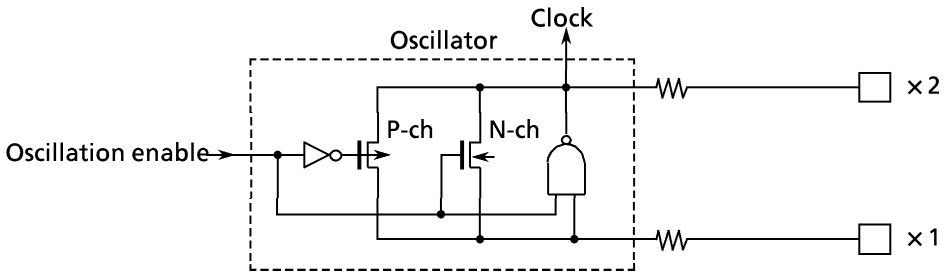
■ $\overline{\text{EA}}$, AM8/$\overline{16}$



■ $\overline{\text{RESET}}$

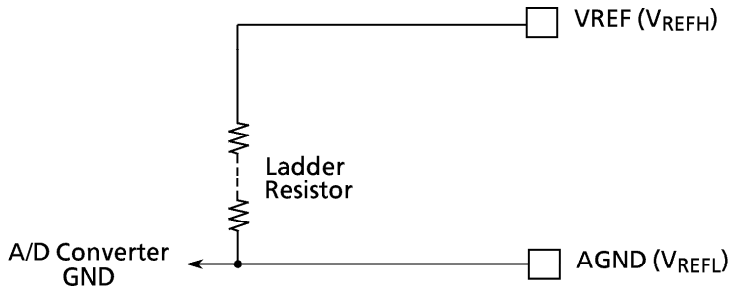


■ X1, X2



Note : The oscillation enable signal becomes nonactive "0" by execution of HALT instruction (STOP mode).

■ VREF (V_{REFH}), AGND (V_{REFL})



7. Care Points and Restriction

(1) Special Expression

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

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

② Read-modify-write Instructions

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

- The Read-modify-write Instructions in the TLCS-900

Exchange

EX	(mem), R		

Arithmetic Operations

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

Logical Operations

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

Bit Operations

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

Rotate and shift

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

③ 1 State

1 cycle clock divided by 2 oscillation frequency is called 1 state.

ex) The case of oscillation frequency is 25 MHz

2/25 MHz = 80 ns = 1 state

(2) Points of Note and Restrictions

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

This pin is connected to the VCC or the GND pin. Do not alter the level while the pin is active.

② Warm-up counter

When releasing STOP mode (by interrupt, for example) in a system that uses an external oscillator, a warm-up time is required until the system clock is output. The warm-up counter operates during the warm-up time.

③ Programmable pull-up resistor

The pull-up resistor of a port can only be set to programmable or non-programmable in input port mode. When using a port as an output port, its pull-up resistor cannot be set to programmable.

④ Watchdog timer

As the watchdog timer is enabled after a reset, disable the watchdog timer when it is not required.

Note that during bus release, the I/O block, including the watchdog timer, still operate.

⑤ CPU (Micro DMA)

Only "LDC cr, r" and "LDC r, cr" can write or read data to or from control registers (eg, transfer source register DMASx) in the CPU.

⑥ As this device does not support minimum mode, do not use the MIN instruction.

⑦ POP SR instruction

Please execute POP SR instruction during DI condition.

⑧ Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{NMI}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.