CMOS 16-bit Microcontrollers

TMP96C031N/TMP96C031F

1. Outline and Device Characteristics

The TMP96C031 are high-speed advanced 16-bit microcontrollers developed for controlling medium to large-scale equipment. TMP96C031N comes in a 64-pin shrink DIP; the TMP96C031F, in a 64-pin flat package.

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

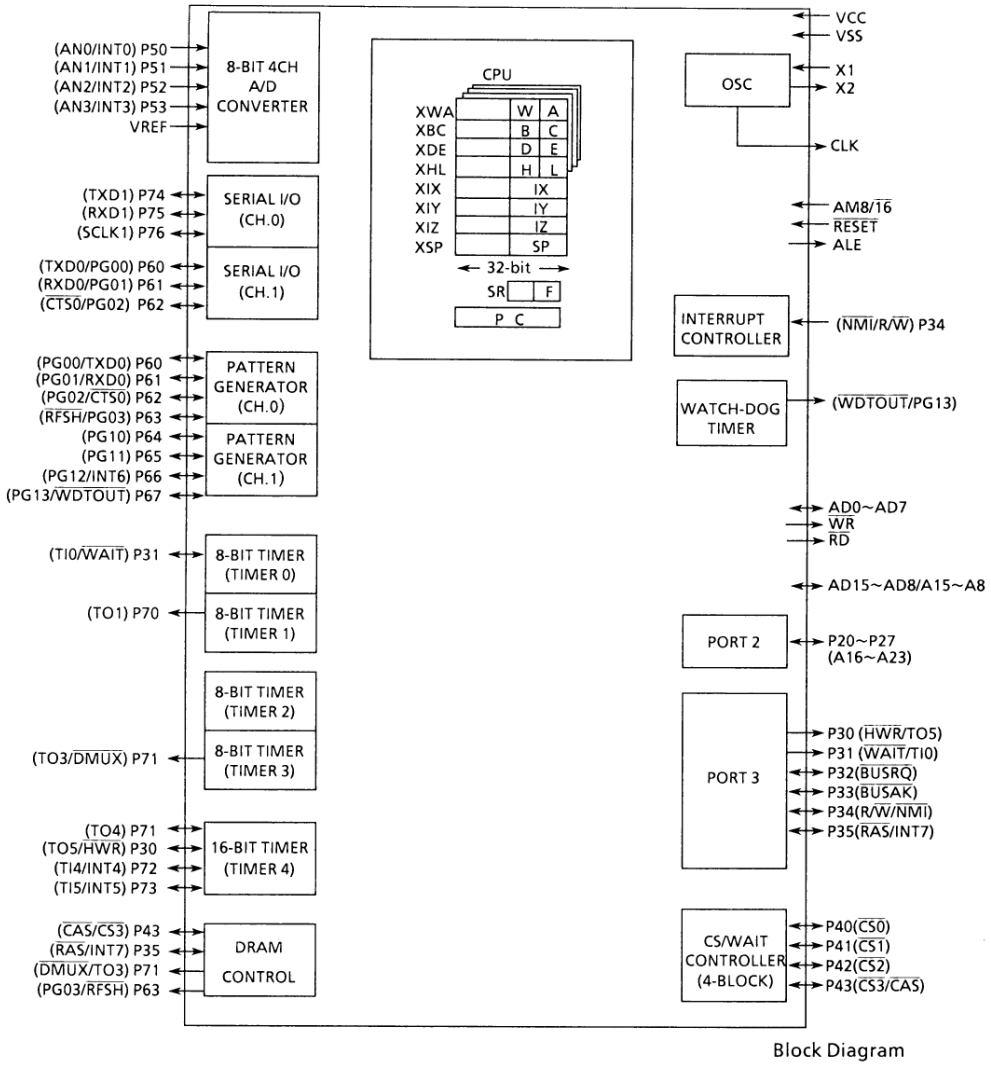
(2) Minimum instruction execution time
  - 200ns @ 20MHz

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

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

7-level priority can be set.

(13) I/O ports: 37 pins
(14) Standby function : 3 HALT modes (RUN, IDLE, STOP)



Block Diagram

Figure 1. TMP96C031F Block Diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP96C031N.

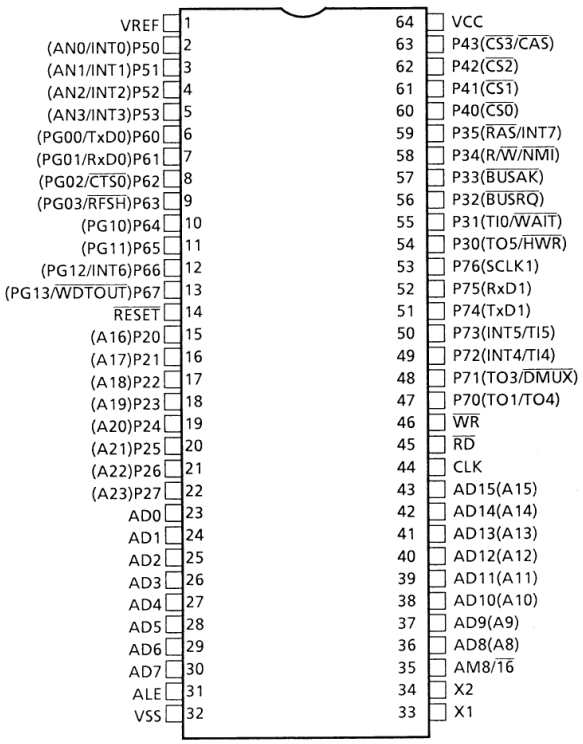


Figure 2.1 (1). Pin Assignment (64-SDIP)

Figure 2.1 (2) shows pin assignment of TMP96C031F.

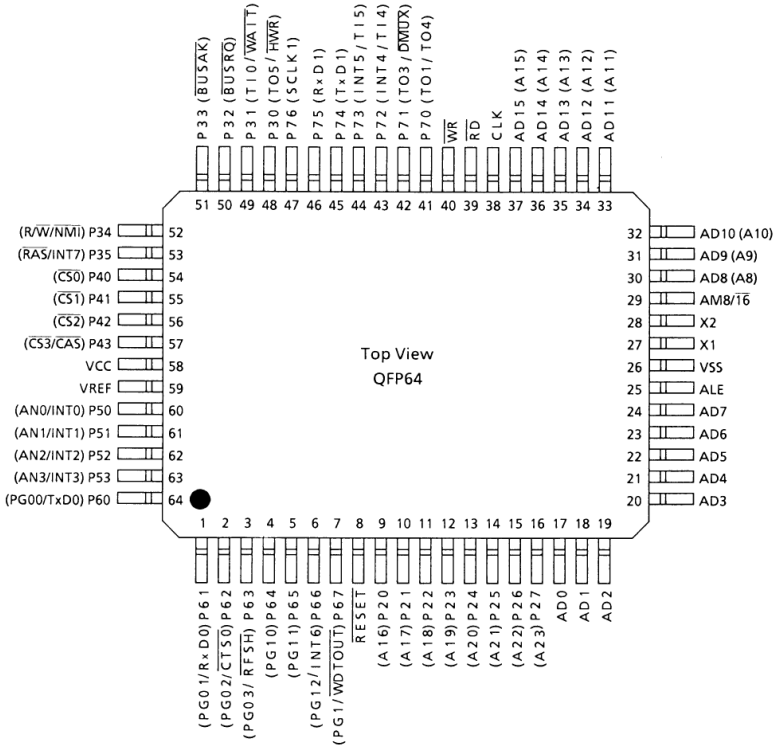


Figure 2.1 (2). Pin Assignment (64QFP)

2.2 Pin Names and Functions

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

Table 2.2. Pin Names and Functions

Pin Name	Number of Pins	I/O	Functions
AD0 ~ AD7	8	Tri-state	Address/data (lower): 0 - 7 for address/data bus
AD8 ~ AD15 A8 ~ A15	8	Tri-state Output	Address data (upper): 8 - 15 for address/data bus Address: 8 to 15 for address bus
P20 ~ P27 A0 ~ A7 A16 ~ A23	8	I/O Output Output	Port 2: I/O port that allows selection of I/O on a bit basis (with pull-down resistor) Address: 0 - 7 for address bus Address: 16 - 23 for address bus
P30 TO5 $\overline{HWR}$	1	I/O Output Output	Port 30: Output port (with pull-up register) Timer output 5: Timer 4 output pin High write: Strobe signal for writing data on pins AD8 - 15
P31 TI0 $\overline{WAIT}$	1	I/O Output Output	Port 31: Output port (with pull-up register) Timer output 0: Timer 0 input Write: Pin used to request CPU bus wait
P32 $\overline{BUSRQ}$	1	I/O Input	Port 32: I/O port (with pull-up register) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins. (For external DMAC)
P33 $\overline{BUSAK}$	1	I/O Input	Port 33: I/O port (with pull-up register) Bus acknowledge: Strobe indicating that AD0 - 15, A0 - 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins are at high impedance after receiving BUSRQ.
P34 R/$\overline{W}$ $\overline{NMI}$	1	I/O Output Input	Port 34: I/O (with pull-up register) Read/write: 1 represents read or dummy cycle 0, write cycle. Non-maskable interrupt request pin; Interrupt request pin with falling edge. Can also be operated at rising edge by program.
P35 $\overline{RAS}$ INT7	1	I/O Output Input	Port 35: I/O (with pull-up register) Row address strobe: Outputs RAS strobe for DRAM. Interrupt request pin 7: Interrupt request pin with rising edge.
P40 $\overline{CS0}$	1	Output Output	Port 40: I/O port Chip select 0: Outputs 0 when address is within specified address area.
P41 $\overline{CS1}$	1	Output Output	Port 41: Output port Chip select 1: Outputs 0 if address is within specified address area.
P42 $\overline{CS2}$	1	Output Output	Port 42: Output port Chip select 2: Outputs 0 if address is within specified address area.
P43 $\overline{CS3}$ $\overline{CAS}$	1	Output Output Output	Port 43: Output port Chip select 3: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P50 ~ P53 AN0 ~ AN3 INT1 ~ INT3	4	Input Input Input	Port 50 ~ 53: Input port Analog input: Input to A/D converter Interrupt request pin 0: Interrupt request pin with programmable level/rising edge. Interrupt request pin 1: Interrupt request pin with programmable rising/falling edge. Interrupt request pin 2 ~ 3: Interrupt request pin with rising edge.
P60 TxD0 PG00	1	I/O Output Output	Port 60: I/O port Serial send data 0 Pattern generator port 00
P61 RxD0 PG01	1	I/O Output Output	Port 61: I/O port Serial receive data 0 Pattern generator port 01

Note: The internal I/O of this device cannot be accessed using the external DMA controller.

Pin Name	Number of Pins	I/O	Functions
P62 $\overline{CTS0}$ PG02	1	I/O Output Output	Port 62: I/O port Serial data send enable 0 (Clear to Send) Pattern generator port 02
P63 $\overline{RFSH}$ PG03	1	I/O Output Output	Port 63: I/O port Refresh out: This is a state signal output pin which indicates that the DRAM controller is in refresh cycle. Pattern generator port 03
P64 PG10	1	I/O Output	Port 64: I/O port Pattern generator port 10
P65 PG11	1	I/O Output	Port 65: I/O port Pattern generator port 11
P66 INT6 PG12	1	I/O Input Output	Port 66: I/O port Interrupt request pin 6: Interrupt request pin with rising edge. Pattern generator port 12
P67 $\overline{WDTOUT}$ PG13	1	I/O Output Output	Port 71: I/O port Watchdog timer output pin Pattern generator port 13
P70 TO1 TO4	1	I/O Output Output	Port 70: I/O port Timer output 1: Timer 0 or 1 output pin Timer output 4: Timer 4 output pin
P71 TO3 $\overline{DMUX}$	1	I/O Output Output	Port 71: I/O port Timer output 3: Timer 2 or Timer 3 output pin DRAM address multiplexor: This pin outputs row address, column address, and selector select signal.
P72 INT4 TI4	1	I/O Input Input	Port 72: I/O port Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge. Timer input 4: Timer 4 count/capture trigger signal input
P73 INT5 TI5	1	I/O Input Input	Port 73: I/O port Interrupt request pin 5: Interrupt request pin with programmable rising edge. Timer input 5: Timer 4 count/capture trigger signal input
P74 TxD1	1	I/O Output	Port 74: I/O port Serial send data 1
P75 RxD1	1	I/O Input	Port 75: I/O port Serial receive data 1
P76 SCLK1	1	I/O I/O	Port 76: I/O port Serial clock I/O 1
CLK	1	Output	Clock output: Outputs ⌈X1÷ 4 ⌋clock. Pulled-up during reset.
$\overline{RD}$	1	Output	Read: Strobe signal for reading external memory.
$\overline{WR}$	1	Output	Write: Strobe signal for writing data on pins AD0 - 7.
AM8/$\overline{16}$	1	Input	Address mode: External data bus width selection pin. Set to “0” for fixed external 16-bit bus or for mixed external 8/16 bit bus and to “1” for fixed external 8-bit bus.
$\overline{RESET}$	1	Input	Reset: Initializes LSI. (With pull-up resistor)
ALE	1	Output	Address latch enable
X1/X2	1	I/O	Oscillator connecting pin
VCC	1		Power supply pin (-5V)
VSS	1		GND pin (0V)

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

3. Operation

This section describes in blocks the functions and basic operations of the TMP96C031F device.

Check the chapter Guidelines and Restrictions for proper care of the device.

3.1 CPU

The TMP96C031F device has a built-in high-performance 16-bit CPU. (For CPU operation, see TLCS-900 CPU in the book Core Manual Architecture User Manual.)

This section describes CPU functions unique to TMP96C031F that are not described in the previous section.

3.1.1 Reset

To reset the TMP96C031F, the $\overline{\text{RESET}}$ input must be kept at 0 for at least 10 system clocks (10 states: 1μs with a 20MHz system clock) within an operating voltage range and with a stable oscillation.

When reset is accepted, the CPU sets as follows:

- Program counter (PC) to 8000H.
- Stack pointer (XSP) for system mode to 100H.
- SYSM bit of status register (SR) to 1. (Sets to system mode.)
- IFF2 to 0 bits of status register to 111. (Sets mask register to interrupt level 7.)
- MAX bit of status register to 0. (Sets to minimum mode.)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

When reset is released, instruction execution starts from address 8000H. 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 I/O ports to input ports).
- Sets the $\overline{\text{WDTOUT}}$ pin to 0. (Watchdog timer is set to enable after reset.)
- Pulls up the CLK pin to 1.
- Sets the ALE pin to 0.

3.1.2 External Data Bus Width Selection Pin (AM8/$\overline{16}$)

The TMP96C031F automatically operates in 8-bit bus/16-bit bus mode after reset depending on how the AM8/$\overline{16}$ pin is set.

The TMP96C031F have altogether the following 23 interrupt sources:

- For mixed external 8/16-bit data bus or fixed 16-bit data bus

  Set this pin to “0”. Then the AD8 to 15/A8 to 15 pins are fixed to functions AD8 to 15.

  The external data bus width is set by the chip select/wait control register described in section 3.6.1.

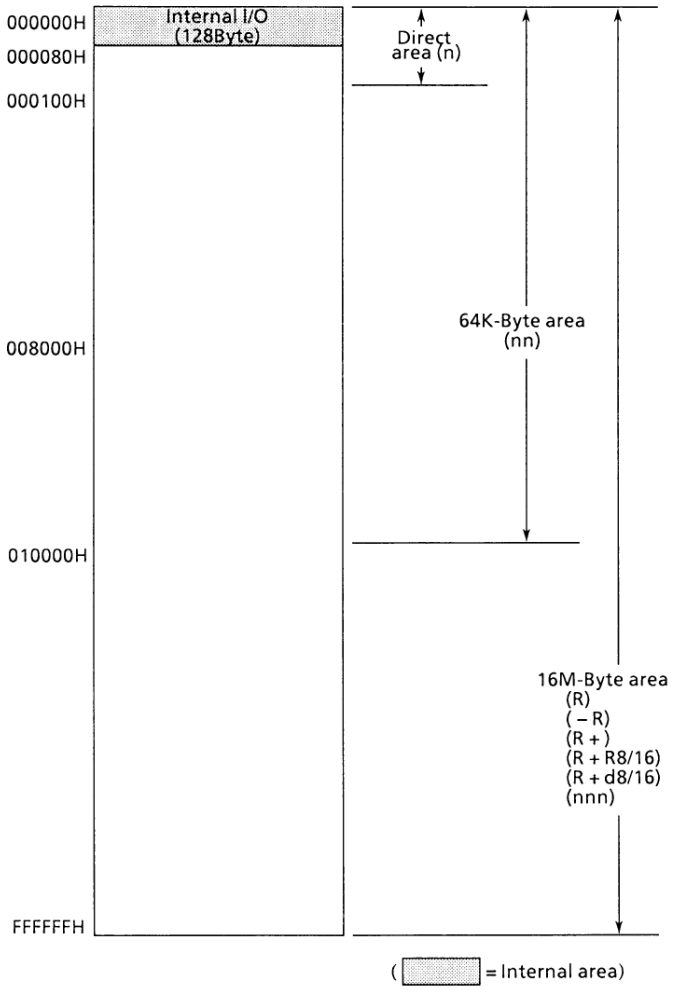
- For fixed external 8-bit data bus

  Set this pin to “1”. Then the AD8 to 15/A8 to 15 pins are fixed to functions A8 to 15.

  The value of chip select/wait control register bit 4 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS>) described in Section 3.6.1 is ignored and the bus is fixed external 8-bit data bus.

3.2 Memory Map

Figure 3.2 is a memory map of the TMP96C031F.



Note: The start address after reset is 8000H. Resetting sets the stack pointer (XSP) on the system mode side to 100H.

Figure 3.2. Memory Map

3.3 Interrupts

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

TMP96C031F have altogether the following 24 interrupt sources:

- Interrupts from the CPU···3
  (Software interrupts, privileged violations, and Illegal (undefined) instruction execution)
- Interrupts from external pins (NMI, INT0, and INT0 to 7)···9
- Interrupts from built-in I/Os···12

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

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

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

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

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

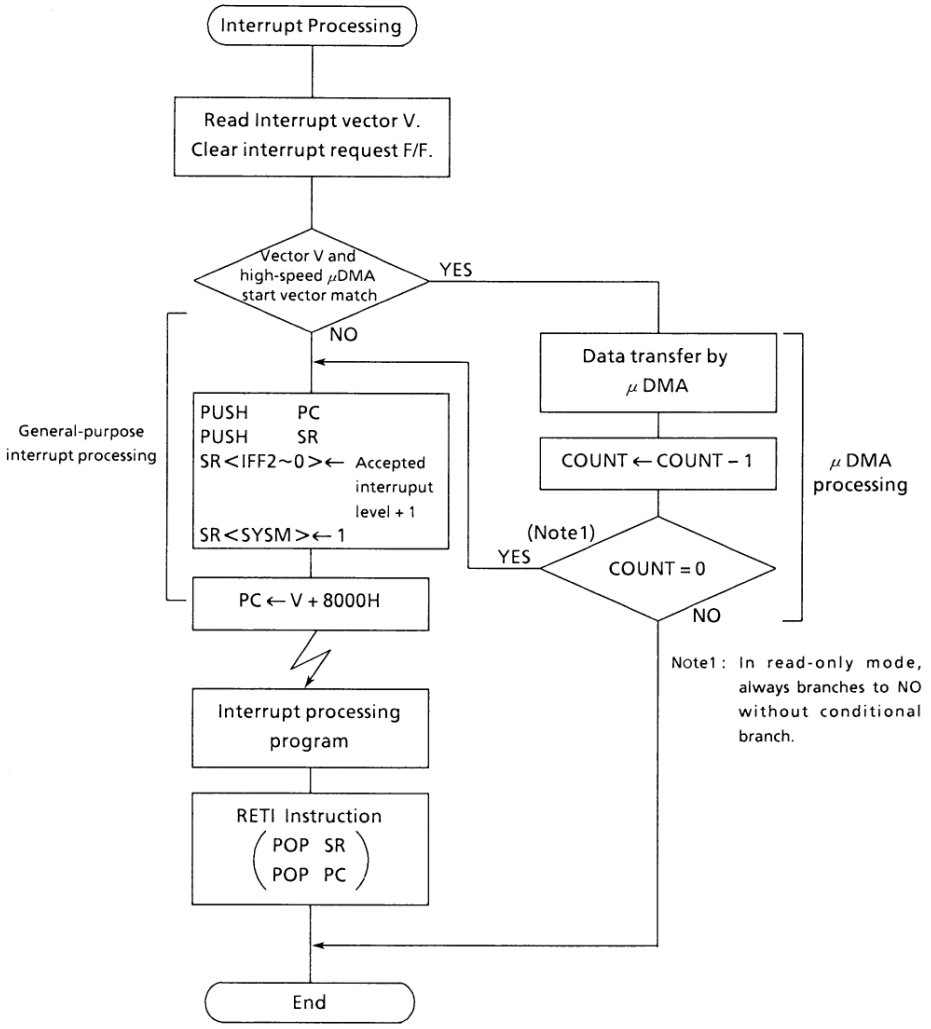


Figure 3.3 (1). Interrupt Processing Flowchart

3.3.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows:

(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).

(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 sets the <SYSM> flag of the status register to 1 and enters the system mode.

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

The table below shows the number of execution states for the above processing times.

Bus Width of Stack Area	Interrupt Processing State Number	
	MAX mode	Min mode
8-bit	23	19
16-bit	17	15

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

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

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

The addresses 008000H to 0081FFH (512 bytes) of the TLCS-900 are assigned for interrupt processing entry area.

Table 3.3 (1) TMP96C031F Interrupt Table

Default Priority	Type	Interrupt Source		Vector Value "V"	Start Address	High-Speed Micro DMA Start to Vector
1	Non-Maskable	Reset	, or SWI0 instruction	0 0 0 0 H	8 0 0 0 H	–
2		INTPREV:	Privileged violation, or SWI1	0 0 1 0 H	8 0 1 0 H	–
3		INTUNDEF:	Illegal instruction, or SWI2	0 0 2 0 H	8 0 2 0 H	–
4		SWI 3 Instruction		0 0 3 0 H	8 0 3 0 H	–
5		SWI 4 Instruction		0 0 4 0 H	8 0 4 0 H	–
6		SWI 5 Instruction		0 0 5 0 H	8 0 5 0 H	–
7		SWI 6 Instruction		0 0 6 0 H	8 0 6 0 H	–
8		SWI 7 Instruction		0 0 7 0 H	8 0 7 0 H	–
9		$\overline{\text{NMI}}$ Pin		0 0 8 0 H	8 0 8 0 H	08H
10		INTWD:	Watchdog timer	0 0 9 0 H	8 0 9 0 H	09H
11	Maskable	INT0 pin		0 0 A 0 H	8 0 A 0 H	0AH
12		INT4 pin		0 0 B 0 H	8 0 B 0 H	0BH
13		INT5 pin		0 0 C 0 H	8 0 C 0 H	0CH
14		INT6 pin		0 0 D 0 H	8 0 D 0 H	0DH
15		INT7 pin		0 0 E 0 H	8 0 E 0 H	0EH
-		(Reserved)		0 0 F 0 H	8 0 F 0 H	0FH
16		INTT0:	8-bit timer 0	0 1 0 0 H	8 1 0 0 H	10H
17		INTT1:	8-bit timer 1	0 1 1 0 H	8 1 1 0 H	11H
18		INTT2:	8-bit timer 2/PWM0	0 1 2 0 H	8 1 2 0 H	12H
19		INTT3:	8-bit timer 3/PWM1	0 1 3 0 H	8 1 3 0 H	13H
20		INTTR4:	16-bit timer 4 (TREG4)	0 1 4 0 H	8 1 4 0 H	14H
21		INTTR5:	16-bit timer 4 (TREG5)	0 1 5 0 H	8 1 5 0 H	15H
22		(Reserved)		0 1 6 0 H	8 1 6 0 H	16H
23		(Reserved)		0 1 7 0 H	8 1 7 0 H	17H
24		INTRX0:	Serial receive (Channel.0)	0 1 8 0 H	8 1 8 0 H	18H
25		INTTX0:	Serial send (Channel.0)	0 1 9 0 H	8 1 9 0 H	19H
26		INTRX1:	Serial receive (Channel.1)	0 1 A 0 H	8 1 A 0 H	1AH
27		INTTX1:	Serial send (Channel.1)	0 1 B 0 H	8 1 B 0 H	1BH
28		INTAD:	A/D conversion completion	0 1 C 0 H	8 1 C 0 H	1CH
29		INT1 pin		0 1 D 0 H	8 1 D 0 H	1DH
30		INT2 pin		0 1 E 0 H	8 1 E 0 H	1EH
31		INT3 pin		0 1 F 0 H	8 1 F 0 H	1FH

3.3.2 High-Speed μDMA

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

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

(1) High-Speed μDMA Operation

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

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

The 32-bit control registers are used for setting transfer source/destination addresses. However, the TLCS-900 has only 24 address pins for output. A 16M-byte space is available for the high-speed μDMA. Also in normal mode operation, the all address space (in other words, the space for system mode which is set by the CS/WAIT controller) can be accessed by high-speed μDMA processing.

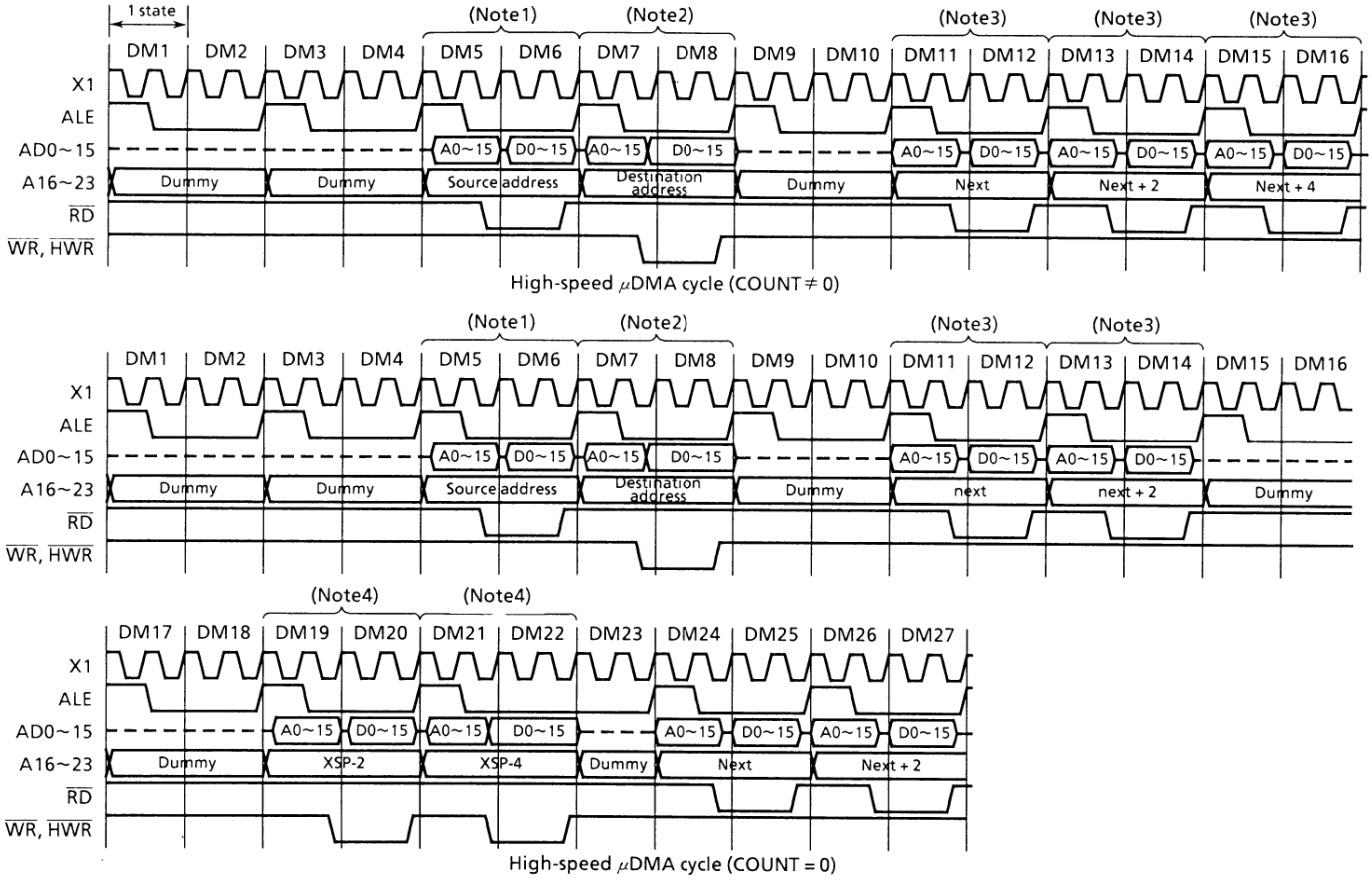
There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes,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

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

After transferring data using the high-speed μDMA and the transfer counter has been decremented to 0, the program goes to a general-purpose interrupt processing. Note that after interrupt processing, when an interrupt for the same channel is generated, if the system requires resetting the transfer counter starts from 65536.

The following section illustrates the high-speed μDMA cycle when the transfer destination address is in INC mode. (MIN mode, 16-bit bus for all address areas, 0 wait).

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



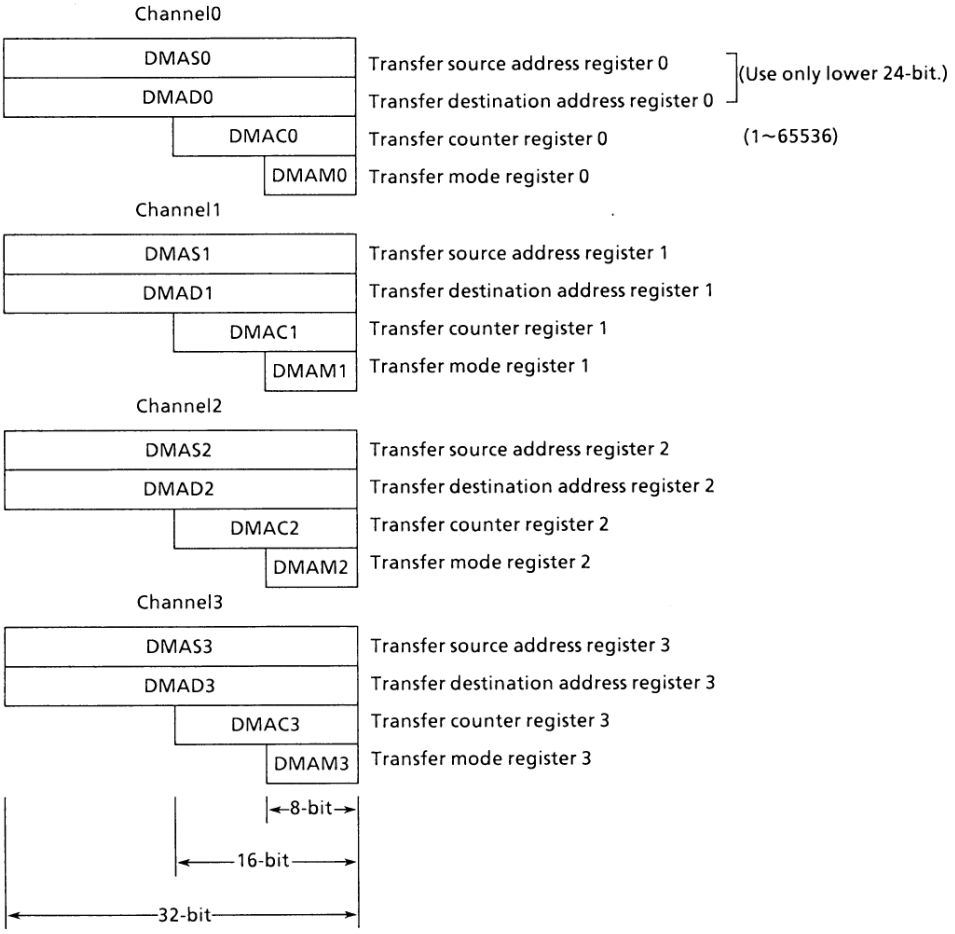
Note 1: If an 8-bit bus is used for the source address area, two states are added.

Note 2: If an 8-bit bus is used for the destination address area, two states are added.

Note 3: A dummy cycle may be generated depending on the instruction queue buffer states.

Note 4: If an 8-bit bus is used for the stack area, two states are added.

(2) Register Configuration (CPU Control Register)



Only the LDCcr.r instruction can set data in those control registers.

(3) Transfer Mode Register Details

(DMAM0~3)

0	0	0	0	Mode

Note : When specifying values for this register, set the upper 4-bit to 0.

Z: 0 = byte transfer, 1 = word transfer

					execution time (Min. 20MHz)
0	0	0	Z	Transfer destination address INC mode ..... for I/O to memory (DMADn +) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0 then INT.	16 states (2μs @16MHz)
0	0	1	Z	Transfer destination address DEC mode ..... for I/O to memory (DMADn –) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0 then INT.	16 states (2μs @16MHz)
0	1	0	Z	Transfer source address INC mode ..... for I/O to memory (DMADn) ← (DMASn +) DMACn←DMACn – 1 if DMACn = 0 then INT.	16 states (2μs @16MHz)
0	1	1	Z	Transfer source address DEC mode ..... for I/O to memory (DMADn) ← (DMASn –) DMACn←DMACn – 1 if DMACn = 0 then INT.	16 states (2μs @16MHz)
1	0	0	Z	Fixed address mode ..... I/O to I/O (DMADn) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0 then INT.	16 states (2μs @16MHz)
1	0	1	1	Counter mode ..... for interrupt counter DMASn←DMASn + 1 DMACn←DMACn – 1 if DMACn = 0 then INT.	11 states (1.375μs @16MHz)

(1 states = 100ns)

Execution time: When 16-bit bus width and 0 wait are set for the transfer destination/source address.

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

All address space (the space for system mode) can be for transfer mode control. accessed by high-speed μDMA. Do not use undefined codes

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

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

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

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

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

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

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

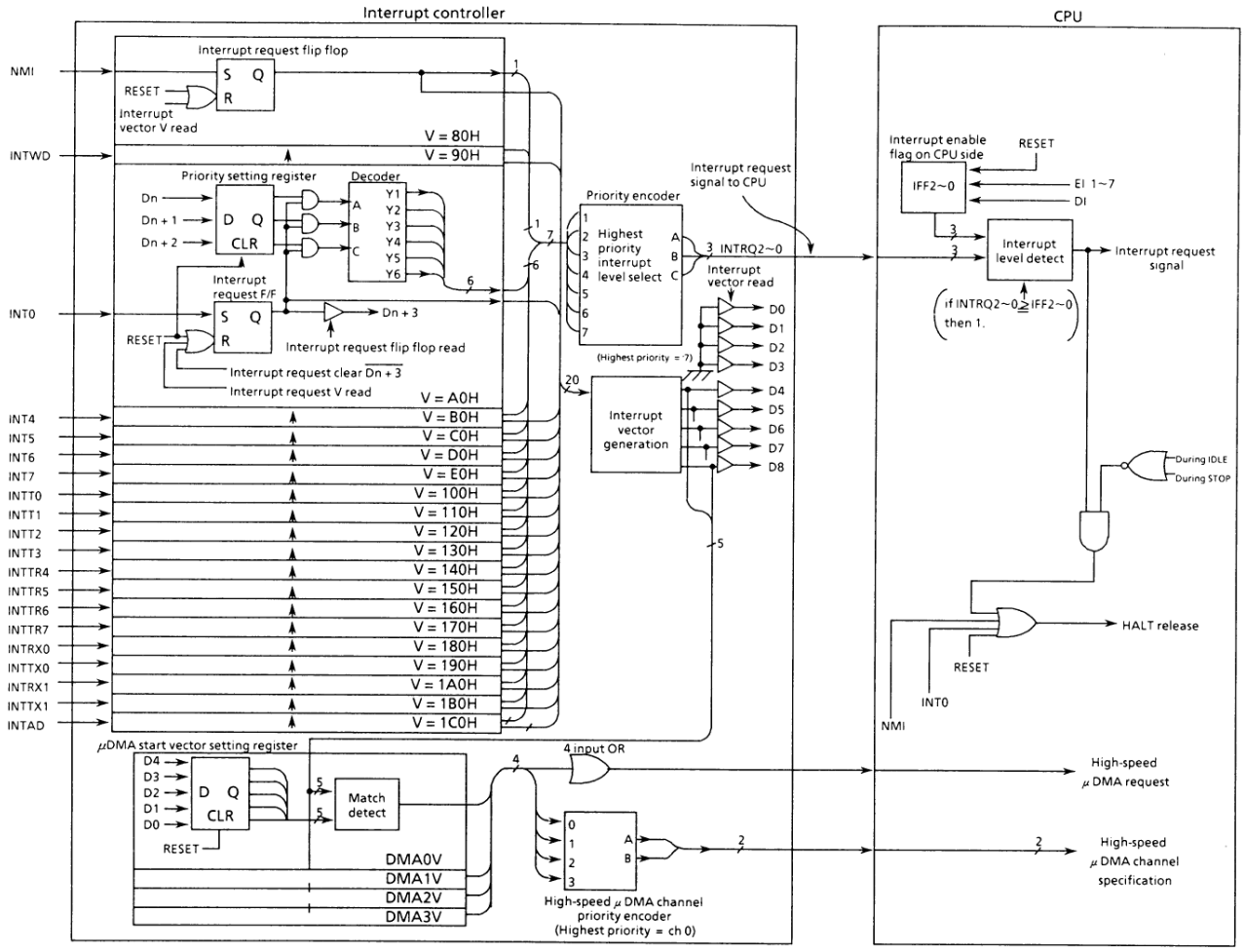


Figure 3.3.3 (1). Block Diagram of Interrupt Controller

(1) Interrupt Priority Setting Register

(Read-modify-write prohibited.)

Symbol	Address	7	6	5	4	3	2	1	0
INTE01	0070H	INT1				INT0			
		I1C	I1M2	I1M1	I1M0	I0C	I0M2	I0M1	I0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE23	0071H	INT3				INT2			
		I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE45	0072H	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	0073H	INT7				INT6			
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET10	0074H	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
INTET32	0075H	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
INTET54	0076H	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
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
INTEAD	0079H	INTAD							
		IADC	IADM2	IADM1	IADM0				
		R/W	W						
		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 0

IIMC0 (007AH)

	7	6	5	4	3	2	1	0
bit Symbol	I4IE	I3IE	I2IE	I1IE	I1EM	I0IE	I0LE	NMIREE
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	1: INT4 Input enable	1: INT3 Input enable	1: INT2 Input enable	1: INT1 Input enable	0: INT1 rising edge 1:INT1 falling edge	1: INT0 Input enable	0: INT0 edge mode 1: INT0 level mode	1: Can be operated in $\overline{NMI}$ rising edge.

Read-modify-write is prohibited.

INTn Input enable

0	INTn Input disable
1	INTn 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

INT1 level enable

0	Rising edge detect interrupt
1	Falling edge detect 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 <I0IE> to "0" maintains the port function during standby mode.

Interrupt Input Mode Control Register 1

IIMC1 (007BH)

	7	6	5	4	3	2	1	0
bit Symbol						I7IE	I6IE	I5IE
Read/Write						W		
After reset						0	0	0
Function						1: INT7 Input enable	1: INT6 Input enable	1: INT5 Input enable

Read-modify-write is prohibited.

INTn Input enable

0	INTn Input disable
1	INTn Input enable

Setting of External Interrupt Pin Functions

Interrupt	Pin Name	Mode	Setting Method
$\overline{NMI}$	P34	Falling edge	IIMC <NMIREE> = 0
		Rising and falling edge	IIMC <NMIREE> = 1
INT0	P50	Rising edge	IIMC <I0LE> = 0, <I0IE> = 1
		Level	IIMC <I0LE> = 1, <I0IE> = 1
INT1	P51	Rising edge	IIMC <I1EM> = 0
		Falling edge	IIMC <I1EM> = 1
INT2	P52	Rising edge	IIMC <I2EM> = 1
INT3	P53	Rising edge	IIMC <I3EM> = 1
INT4	P72	Rising edge	T4MOD <CAP12M1, 0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD <CAP12M1, 0> = 1, 0
INT5	P73	Rising edge	–
INT6	P66	Rising edge	IIMC <I6IIE> = 1
INT7	P35	Rising edge	IIMC <I7IIE> = 1

(3) High-Speed μDMA Start Vector

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

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

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

DMA0V (007CH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
Read/Write				W				
After reset				0	0	0	0	0
Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed μDMA channel 0 is processed.							

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

DMA1V (007DH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
Read/Write				W				
After reset				0	0	0	0	0
Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed μDMA channel 1 is processed.							

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

DMA2V (007EH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
Read/Write				W				
After reset				0	0	0	0	0
Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed μDMA channel 2 is processed.							

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

DMA3V (007FH)

	7	6	5	4	3	2	1	0
bit Symbol				DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
Read/Write				W				
After reset				0	0	0	0	0
Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed μDMA channel 3 is processed.							

(4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently of each other. Therefore, if the instruction used to clear an interrupt request flag of an interrupt is fetched before the interrupt is generated, it is possible that the CPU might execute the fetched instruction to clear the interrupt request flag while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector 00A0H and start the interrupt processing from the address 80A0H.

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

3.4 Standby Function

When the HALT instruction is executed, the TMP96C031 enters RUN, IDLE, or STOP mode depending on the contents of the HALT mode setting register.

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

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

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

The states of the port pins in STOP mode can be set as listed in Table 3.4 (1) using the I/O register WDMOD <DRVE> bit.

WDMOD (005CH)

	7	6	5	4	3	2	1	0
Bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	1 : WDT Enable	00 : 2^{16} / fc 01 : 2^{18} / fc 10 : 2^{20}/ fc 11 : 2^{22}/ fc Detection time		Warming up time 0 : 2^{16} /fc 1 : 2^{18} /fc	Standby mode 00 : RUN mode 01 : STOP mode 10 : IDLE mode 11 : Don't care		1: Connects watchdog timer output to $\overline{\text{RESET}}$ pin internally.	1: Drive pin even in STOP mode.

When STOP mode is released by other than a reset, the system clock output starts after allowing some time for warming up set by the warming-up counter fro stabilizing the built-in oscillator. To release STOP mode by reset, it is necessary to allow the oscillator to stabilize.

To release standby mode, a reset or an interrupt is used. To release IDLE or STOP mode, only an interrupt by the $\overline{\text{NMI}}$ or INT0 pin, or a reset can be used. The details are described below:

Standby Release by Interrupt

Standby Mode \ Interrupt Level	Interrupt Mask (IFF2 to 0) ≤ Interrupt Request Level	Interrupt Mask (IFF2 to 0) > Interrupt Request Level
RUN	Can be released by any interrupt. After standby mode is released, interrupt processing starts.	Can only be released by INT0 pin. Processing resumes from address next to HALT instruction.
IDLE	Can only be released by $\overline{\text{NMI}}$ or INT0 pin. After standby mode is released, interrupt processing starts.	↑
STOP	↑	↑

Note 1: When releasing standby setting INT0 to high level input mode, keep it high until interrupt processing starts. If the level drops to low, interrupt processing cannot be started correctly.

Table 3.4 (1) Pin States in STOP Mode

Pin Name	I/O	DRVE = 0	DRVE = 1
AD0 ~ AD7	AD0 ~ 7	–	–
AD8 ~ AD15	AD8 ~ 15 A8 ~ 15	– –	–
P20 ~ P27	Input mode Output mode/A16 ~ 23	PD* PD*	PD Output
P30 ~ P33	Input mode Output mode	PD* PD*	PD Output
P34 (R/$\overline{W}$/$\overline{NMI}$)	Input mode Output mode $\overline{NMI}$	PU* PU* Input	PU Output Input
P35 ($\overline{RAS}$/INT7)	Input mode Output mode $\overline{RAS}$	PU* PU* Output	PU Output Output
P40 ~ P42 ($\overline{CS0}$ ~ $\overline{CS2}$)	Output	PU*	Output
P43 ($\overline{CS3}$/$\overline{CAS}$)	Output $\overline{CAS}$	PU* Output	Output Output
P50 (AN0/INT0)	Input INT0	– Input	Input Input
P51 ~ P53	Input	–	Input
P60 ~ P66	Input mode Output mode	– –	Input Output
P67 (P13/$\overline{WDTOUT}$)	Input mode Output mode $\overline{WDTOUT}$	– – Output	Input Output Output
P70 ~ P76	Input mode Output mode	– –	Input Output
ALE	Output	“0”	“0”
CLK	Output	–	“1”
$\overline{RESET}$	Input	Input	Input
$\overline{WR}$	Output	–	“1” Output
$\overline{RD}$	Output	–	“1” Output
AM8/$\overline{16}$	Input	Input	Input
X1	Input	–	–
X2	Output	“1”	“1”

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

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

3.5 Port Functions

The input/output ports of the TMP96C031F consist of a total of 37 bits.

In addition to general purpose input/output port functions, these port pins also function as input/outputs for internal CPU and built-in I/O. Table 3.5 (1) shows the function of each port pin.

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

Table 3.5 (1) Port Function

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port2	P20 to P27	8	Input/Output	↓	Bit	A0 to A7/ A16 to A23
Port3	P30	1	Input/Output	↑	Bit	TO5/$\overline{\text{HWR}}$
	P31	1	Input/Output	↑	Bit	TI0/$\overline{\text{WAIT}}$
	P32	1	Input/Output	↑	Bit	$\overline{\text{BUSRQ}}$
	P33	1	Input/Output	↑	Bit	$\overline{\text{BUSAK}}$
	P34	1	Input/Output	↑	Bit	R/$\overline{\text{W}}$/$\overline{\text{RAS}}$
	P35	1	Input/Output	↑	Bit	$\overline{\text{RAS}}$/INT7
Port4	P40	1	Output	↑	(Fixed)	$\overline{\text{CS0}}$
	P41	1	Output	↑	(Fixed)	$\overline{\text{CS1}}$
	P42	1	Output	↑	(Fixed)	$\overline{\text{CS2}}$
	P43	1	Output	↑	(Fixed)	$\overline{\text{CS3}}$/$\overline{\text{CAS}}$
Port5	P50 to P53	4	Input	–	(Fixed)	AN0 ~ AN3
Port6	P60	1	Input/Output	–	Bit	PG00/TxD0
	P61	1	Input/Output	–	Bit	PG01/RxD0
	P62	1	Input/Output	–	Bit	PG02/$\overline{\text{CTS0}}$
	P63	1	Input/Output	–	Bit	PG03/$\overline{\text{RSFH}}$
	P64	1	Input/Output	–	Bit	PG10
	P65	1	Input/Output	–	Bit	PG11
	P66	1	Input/Output	–	Bit	PG12/INT6
	P67	1	Input/Output	–	Bit	PG13/$\overline{\text{WDTOUT}}$
Port7	P70	1	Input/Output	–	Bit	TO1/TO4
	P71	1	Input/Output	–	Bit	TO2/$\overline{\text{DMUX}}$
	P72	1	Input/Output	–	Bit	INT4/TI4
	P73	1	Input/Output	–	Bit	INT5/TI5
	P74	1	Input/Output	–	Bit	TxD1
	P75	1	Input/Output	–	Bit	RxD1
	P76	1	Input/Output	–	Bit	SCLK1

3.5.1 Programmable Pull-up/Pull-down

PORT2 has a built-in pull-down resistor and PORT3 and PORT4 have a built-in pull-up resistor. Normally, their load can be turned on or off from software by setting the value of the output latch (registers P2, P3, and P4) during input mode. They can also be set in stand-by (STOP) mode and the load can be turned on or off when the immediately preceding setting is the value of output latch in input mode or is the value of output data in output mode.

Table 3.5 (2) Pull-up/down Function Setting

	Pull-up/down	Programmable		Setting	
		Output latch Output data	ON/OFF	Normal	Standby (STOP) mode
PORT2 (I/O)	Pull-down	0	ON	Setting enabled only in input mode	Setting enabled in input/ output mode
		1	OFF		
PORT3 (I/O)	Pull-up	0	ON	↑	↑
		1	OFF		
PORT4 (Output)	Pull-up	0	ON	Setting disabled	Setting enabled only in output mode
		1	OFF		

3.5.2 Bus Release Function

The pull-up/down function explained in section 3.5.1 is also used to stabilize bus control signal at bus release.

Table 3.5 (2) shows pin states at bus release ($\overline{BUSAK}$ = 0).

Pin Name	Pin state at bus release	
	Port mode	Function mode
AD0 - AD15 AD0 - AD7 (A8 ~ A15)	–	Becomes high impedance.
P20 - P27 (A16 ~ 23)	No status change. (Does not become high impedance.)	First sets all bits to low, then sets output buffer to off. Internal pull-down is added regardless to output latch value.
$\overline{RD}$ $\overline{WR}$	–	First sets all bits to high, then sets them to high impedance.
P30 ($\overline{HWR}$) P34 (R/$\overline{W}$)	No status change. (Does not become high impedance.)	First sets all bits to high, then sets output buffer to off. Internal pull-down is added regardless to output latch value.
P40 ($\overline{CS0}$) P41 ($\overline{CS1}$) P42 ($\overline{CS2}$) P43 ($\overline{CS3}$)	No status change. (Does not become high impedance.)	First sets all bits to high, then sets output buffer to off. Internal pull-down is added regardless to output latch value.
P71 ($\overline{DMUX}$) P63 ($\overline{RFSH}$)	No status change. (Does not become high impedance.)	First sets all bits to high, then sets them to high impedance.
P35 ($\overline{RAS}$) P43 ($\overline{CAS}$)	No status change. (Does not become high impedance.)	No status change. (Does not become high impedance.)

Figure 3.5 (2) shows the external bus interface when the bus release function is in use. The internal I/O of this device cannot be accessed when the bus is released.

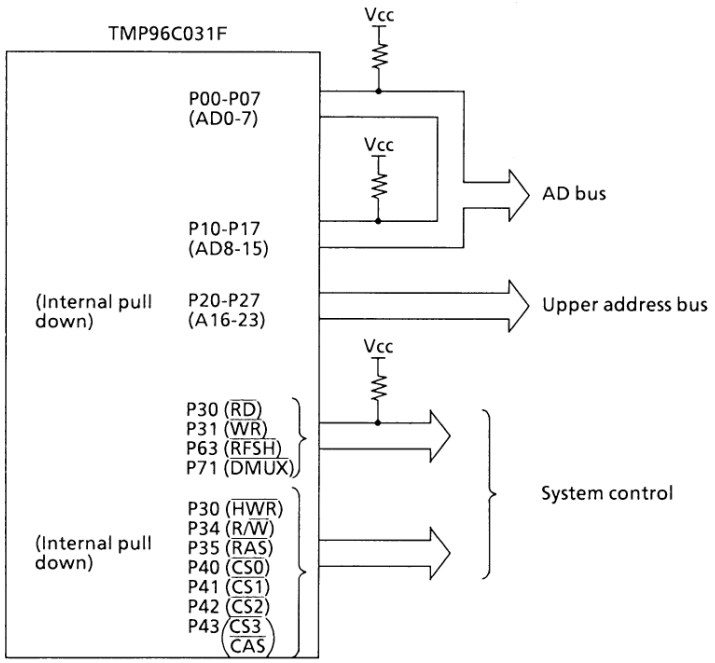


Figure 3.5 (1). External bus interface example when bus release function is in use

3.5.3 Port 2 (P20 - P27)

Port 2 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P2CR and function register P2FC. Resetting resets all bits of output latch P2, control register P2CR and function register P2FC to 0. It also sets Port 2 to input mode and connects a pull-down resistor.

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

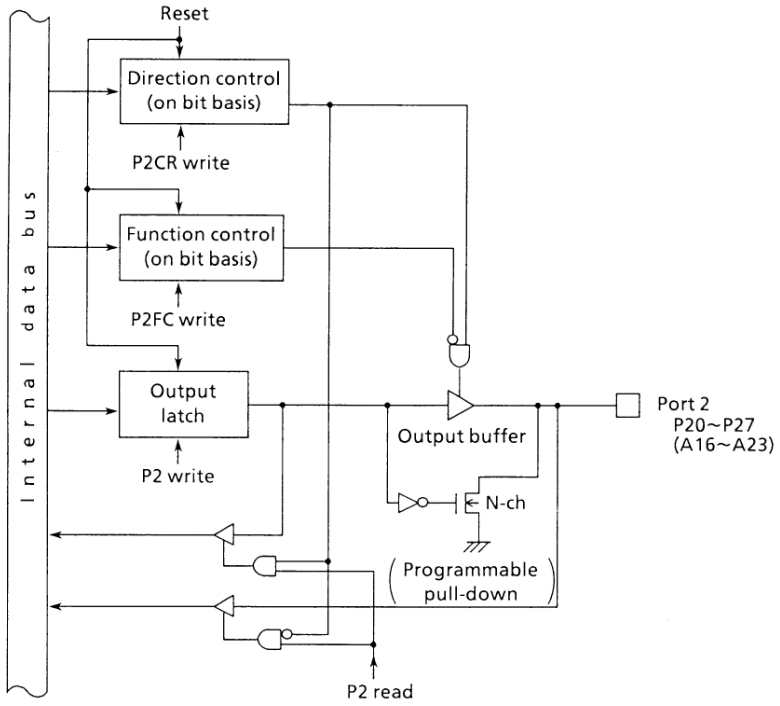


Figure 3.5 (2). Port 2

Port 2 Register

P2 (0006H)

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

Port 2 Control Register

P2CR (0008H)

	7	6	5	4	3	2	1	0
bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<< Pair P2CR with P2FC. See P2FC below. >>							

Port 2 Function Register

P2FC (0009H)

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

Note: When using as an address bus, set P2CR first, then P2FC.

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

Port 2 function setting

P2CR <P2XC> \ P2FC<P2XF>	0	1
0	Input	–
1	Output	address bus (A23-16)

Note : <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR.

Figure 3.5 (3). Registers for Port 2

3.5.4 Port 3 (P30 - P35)

Port 3 is a 6-bit general-purpose I/O port. I/O can be set bit by bit using control registers P3CRL and P3CRH. Resetting sets all bits of P3 to 0; P30 to P35 to input mode and connects a pull-up resistor. In addition to functioning as a general-purpose I/O port, port 3 is also used for CPU control/status signal interrupt input, and timer I/O.

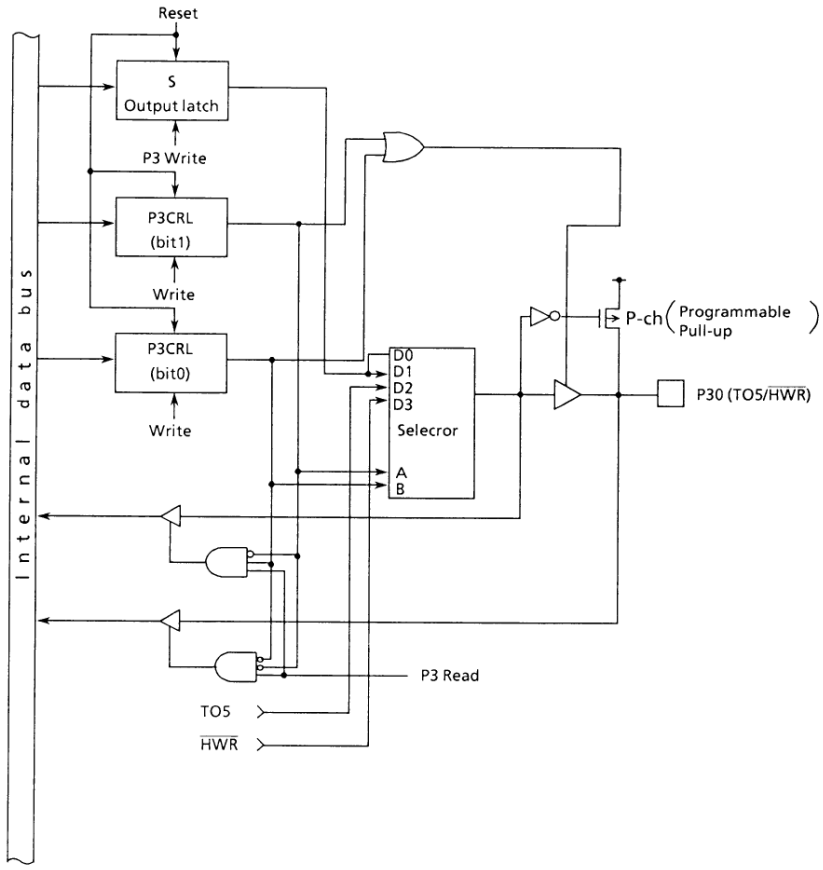


Figure 3.5 (4). Port 3 (P30)

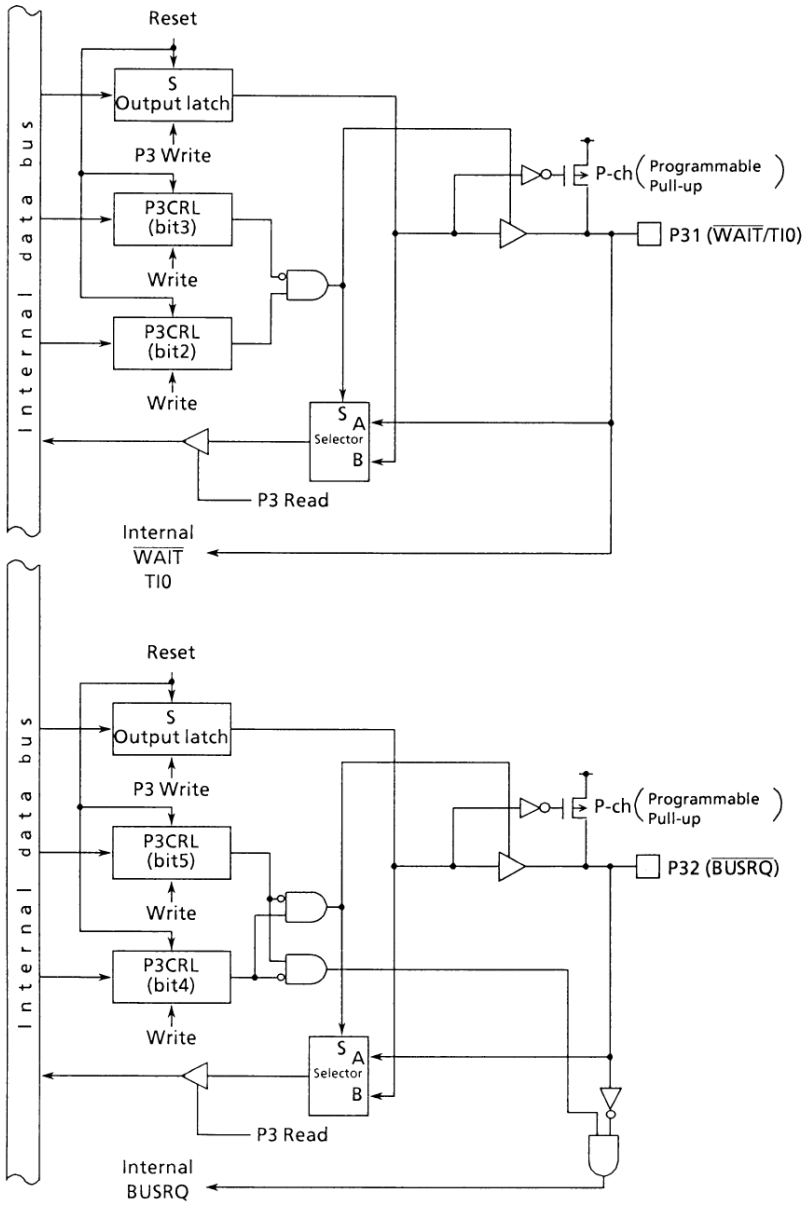


Figure 3.5 (5). Port 3 (P31, P32)

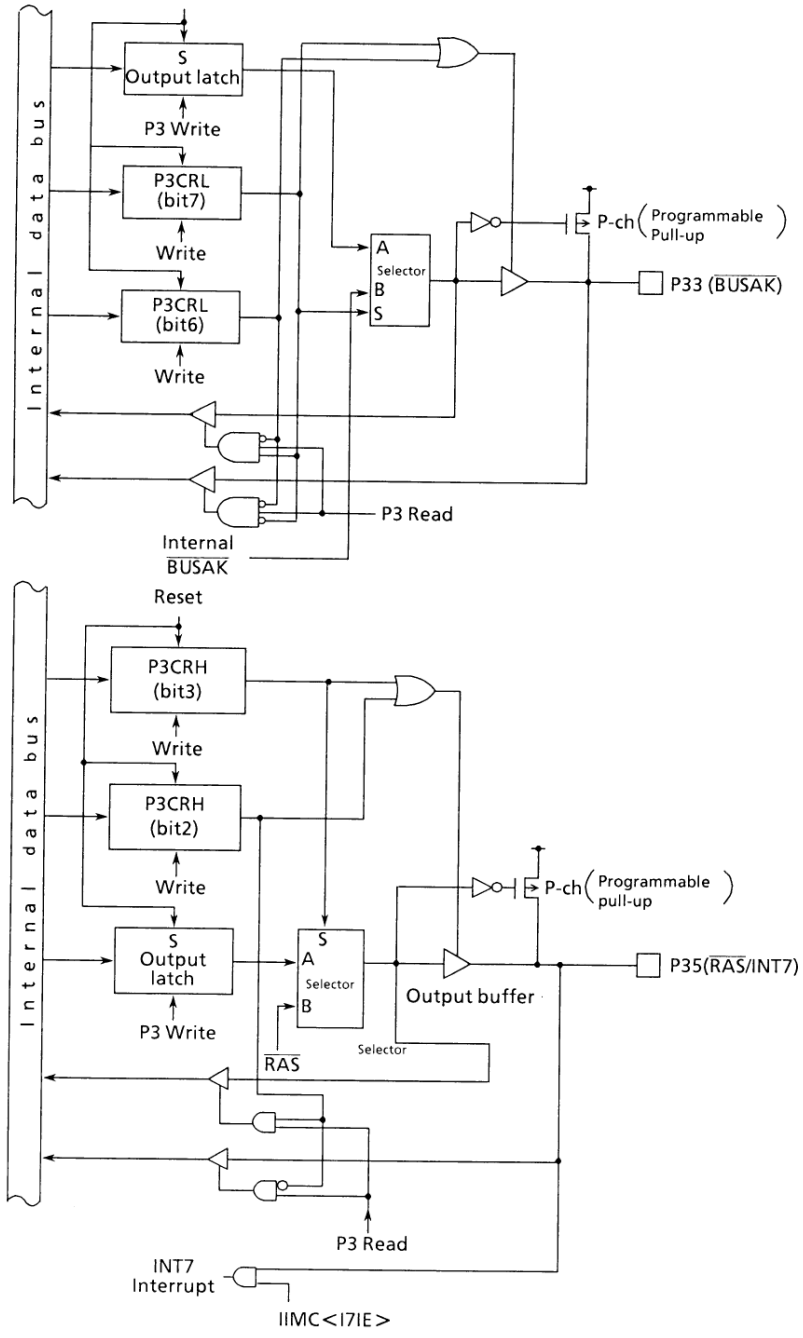


Figure 3.5 (6). Port 3 (P33, P35)

(1) P34/$\overline{\text{NMI}}$/R/$\overline{\text{W}}$

Port 34 is a general-purpose I/O port, shared with non-maskable interrupt input pin ($\overline{\text{NMI}}$). The $\overline{\text{NMI}}$ pin is selected by the control register P3CRH <P34C1,P34C0>. By setting <P34C1,P34C0> = <0,0>, it turns to the $\overline{\text{NMI}}$ input pin. Since the $\overline{\text{NMI}}$ pin is specified only once, the $\overline{\text{NMI}}$ pin cannot be switched to the general-purpose port. The <P34C1,P34C0> should be initialized to "0" by resetting in order to switch to the general-purpose I/O port mode. Port3 register (P34) is set to be "1" when the pull-up resistor is attached.

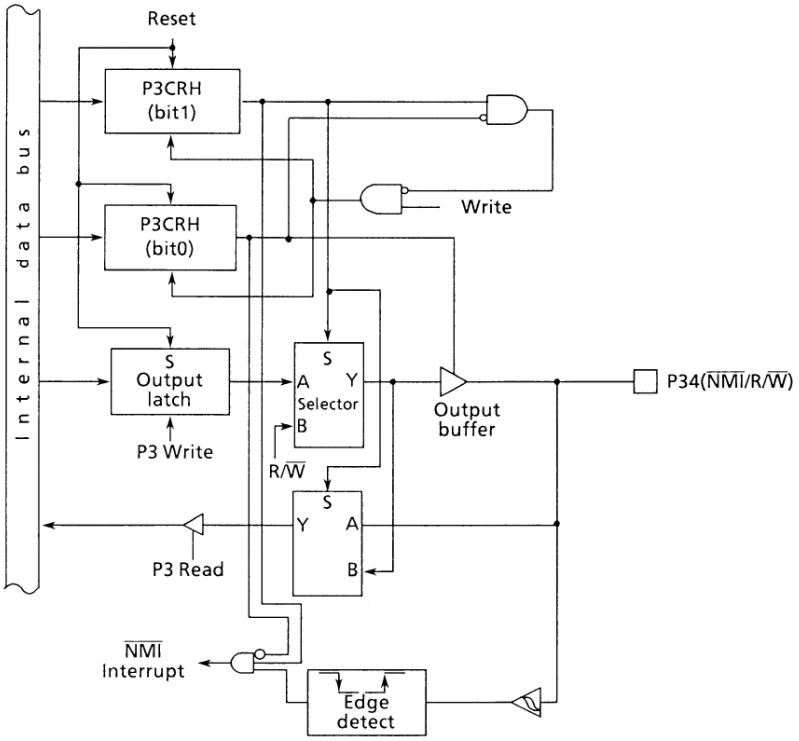
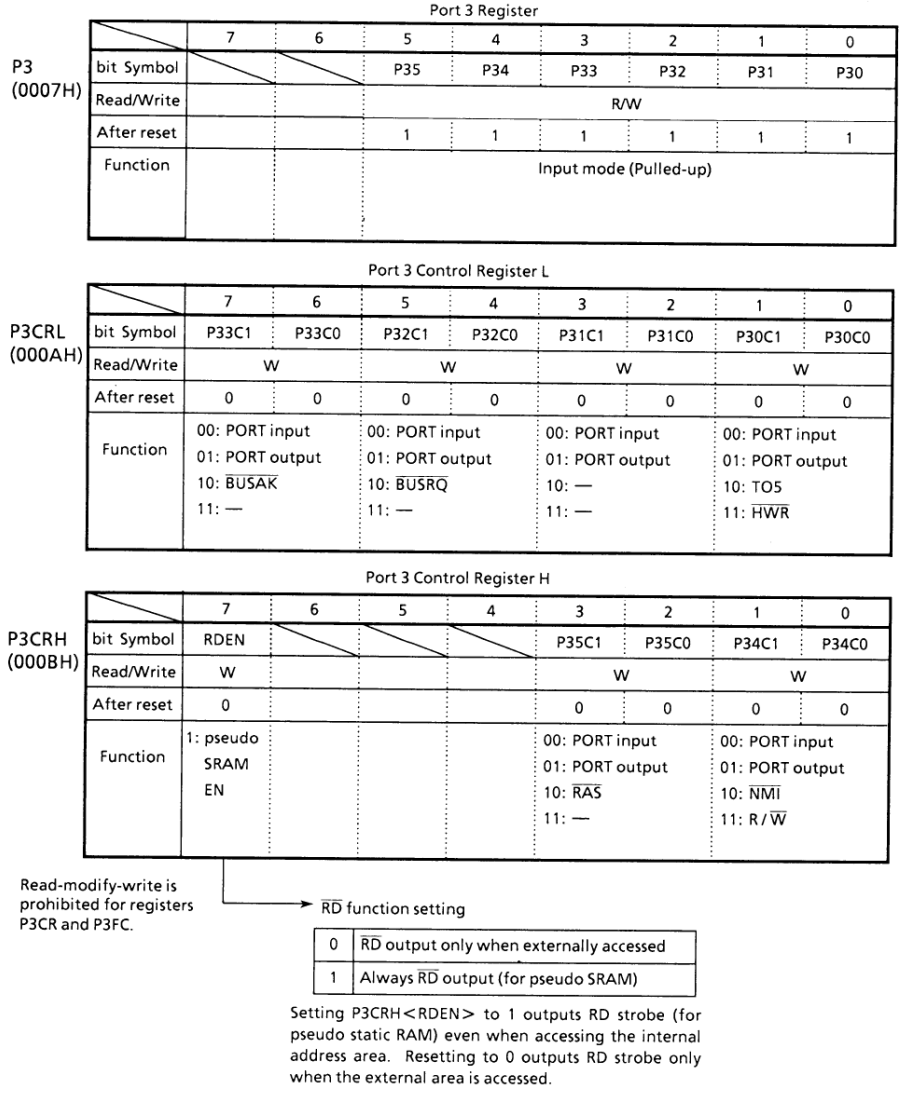


Figure 3.5 (6). Port 3 (P33, P34)



Port 3 Register

P3 (0007H)

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

Port 3 Control Register L

P3CRL (000AH)

	7	6	5	4	3	2	1	0
bit Symbol	P33C1	P33C0	P32C1	P32C0	P31C1	P31C0	P30C1	P30C0
Read/Write	W		W		W		W	
After reset	0	0	0	0	0	0	0	0
Function	00: PORT input 01: PORT output 10: $\overline{BUSAK}$ 11: —		00: PORT input 01: PORT output 10: $\overline{BUSRQ}$ 11: —		00: PORT input 01: PORT output 10: — 11: —		00: PORT input 01: PORT output 10: TO5 11: $\overline{HWR}$	

Port 3 Control Register H

P3CRH (000BH)

	7	6	5	4	3	2	1	0
bit Symbol	RDEN				P35C1	P35C0	P34C1	P34C0
Read/Write	W				W		W	
After reset	0				0	0	0	0
Function	1: pseudo SRAM EN				00: PORT input 01: PORT output 10: $\overline{RAS}$ 11: —		00: PORT input 01: PORT output 10: $\overline{NMI}$ 11: R / $\overline{W}$	

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

$\overline{RD}$ function setting

0	$\overline{RD}$ output only when externally accessed
1	Always $\overline{RD}$ output (for pseudo SRAM)

Setting P3CRH<RDEN> to 1 outputs RD strobe (for pseudo static RAM) even when accessing the internal address area. Resetting to 0 outputs RD strobe only when the external area is accessed.

Figure 3.5 (8). Port 3 Registers

Note: There is no port/function switch register for pin P31 (TIO/$\overline{WAIT}$). For example, if pin P31 is used as an input port, data are input to 8-bit timer 0. If pin P31 is used as the $\overline{WAIT}$ pin, set P3CRL <P31C1,0) > to 00, and bits 3 and 2 <BXW1,0> in the chip select/wait control register to 10.
If pin P35 ($\overline{RAS}$/INT7) is used as the INT7 pin, set P3CRH <P35C1,0) to 00 and <171E> to 1.

3.5.5 Port 4 (P40 - P43)

Port 4 is a 4-bit output dedicated port. Port 4 is also used for chip select CS0 - CS3 outputs and column address strobe $\overline{CAS}$ ($\overline{CS3}$ only) output. To select the function to be used, use function register P4FC. Resetting sets the output register for P40, P41, and P42 to 1; the output register for P42 to 0; all bits in the function register to 0. P40, P41, and P43 are set to output ports for outputting 1; P42 to output port for outputting 0.

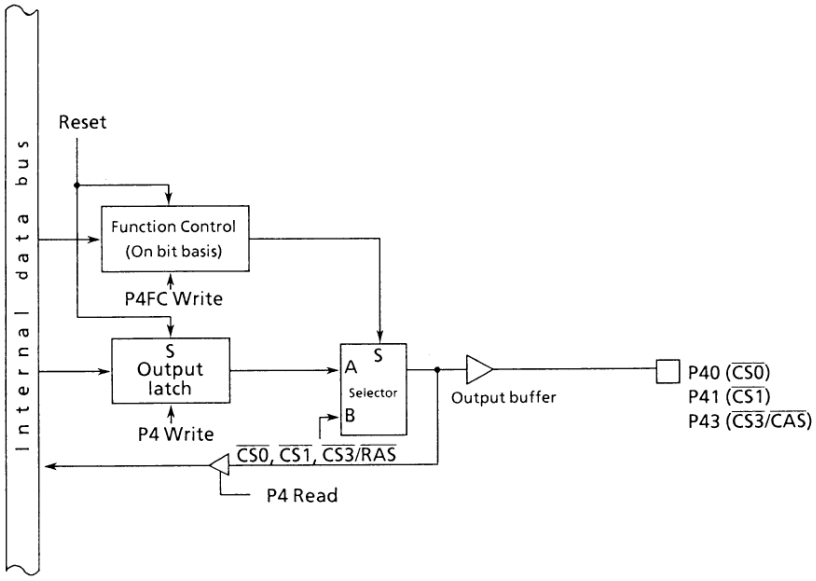


Figure 3.5 (9). Port 4 (P40, P41, P43)

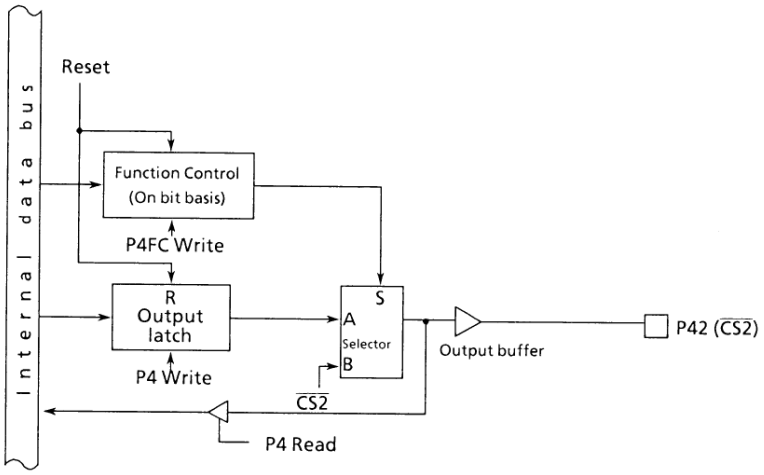


Figure 3.5 (10). Port 4 (P42)

Port 4 Register

P4 (000CH)	7	6	5	4	3	2	1	0
bit Symbol					P43	P42	P41	P40
Read/Write					R/W			
After reset					1	0	1	1
Function					Output mode			

Port 4 Function Register

P4FC (0010H)	7	6	5	4	3	2	1	0
bit Symbol	BUS WDT				P43F	P42F	P41F	P40F
Read/Write	W				W			
After reset	0				0	0	0	0
Function	0: BUSRQ DIS 1: BUSRQ EN				0: PORT 1: CS3 /CAS	0: PORT 1: CS2	0: PORT 1: CS1	0: PORT 1: CS0

→ Explained in section 3.12, Watchdog timer.

P4FC is disabled for read-modify-write.

Figure 3.5 (11). Registers for Port 4

P4FC is disabled for read-modify-write.
Note: To select the function to be used for P43, use the B3CS register for the chip select/wait controller.

3.5.6 Port 5 (P50 - P53)

Port 5 is a 4-bit input dedicated port which is also used for analog inputs or external interrupts.

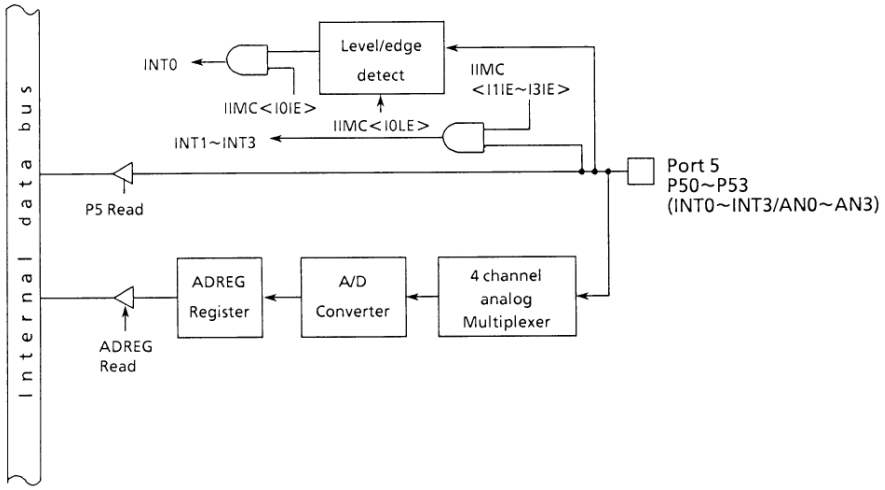


Figure 3.5 (11). Port 5 (P50, P51, P52, P53)

Port 5 Register

P5 (000DH)

	7	6	5	4	3	2	1	0
bit Symbol					P53	P52	P51	P50
Read/Write					R			
Function					Input mode			

Note : There is no input switch register for AN0-3/INT0-3; data are input to both.
When port 5 is used for INT0-5, set interrupt input mode control registers 0 and 1, IIMC0 and 1<101E-131E>, to 1.
When port 5 is used as the input channel for the A/D converter, set the A/D converter mode register, ADMOD.

Figure 3.5 (12). Register for Port 5

3.5.7 Port 6 (P60 - P67)

Port 6 is an 8-bit port. I/O can be set bit by bit. In addition to functioning as an I/O port, pins P60 to P67 function as follows:
P60 - P63/P64 - P67: pattern generate PG0/PG1 output
P60: serial channel TxD0 output pin and programmable open drain function.
P61: serial channel RxD0 output pin
P62: serial channel $\overline{\text{CTS0}}$ output pin
P63: DRAM controller refresh signal pin
P66: external interrupt request input INT6 pin
P67: watchdog timer $\overline{\text{WDT}}$ output pin. Set using port 6 control registers. P6CRL and P6CRH.

Resetting sets control registers P6CRL and P6CRH to 0; all bits to input mode.

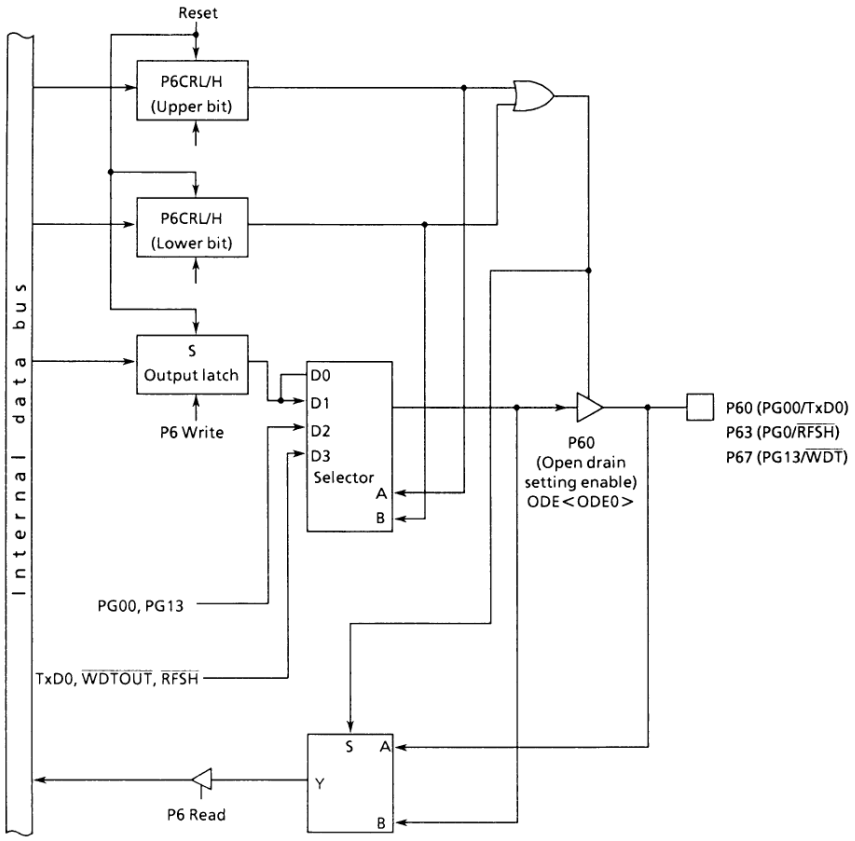


Figure 3.5 (13) Port 6 (P60, P67)

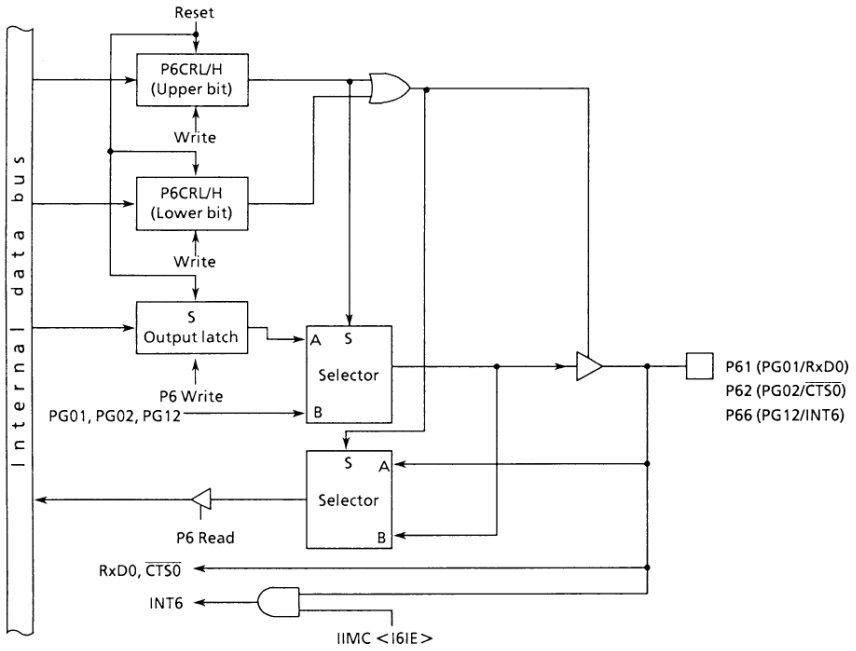


Figure 3.5 (14). Registers for Port 6 (P61, P62, P66)

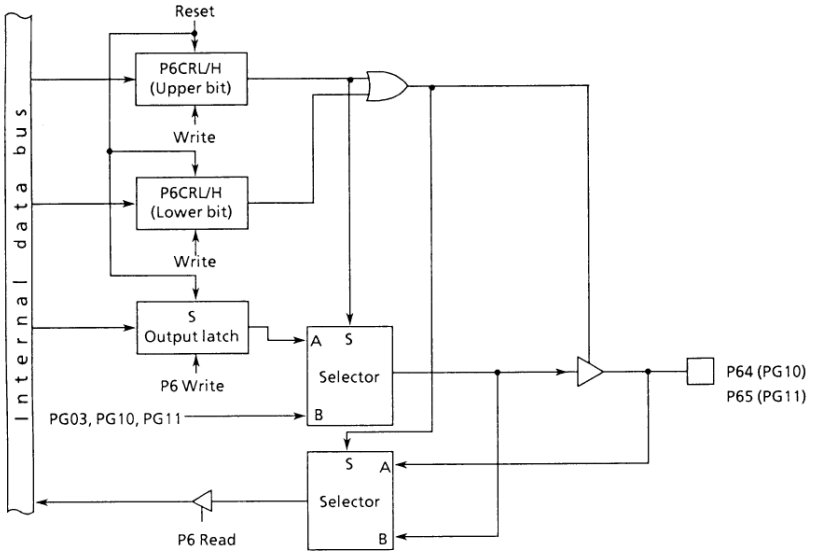


Figure 3.5 (15). Port 6 (P63, P64, P65)

Port 6 Register

P6 (0012H)

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

Port 6 Control Register L

P6CRL (0014H)

	7	6	5	4	3	2	1	0
bit Symbol	P63C1	P63C0	P62C1	P62C0	P61C1	P61C0	P60C1	P60C0
Read/Write	W		W		W		W	
After reset	0	0	0	0	0	0	0	0
Function	00: PORT input 01: PORT output 10: PG03 11: $\overline{RFSH}$		00: PORT input 01: PORT output 10: PG02 11: —		00: PORT input 01: PORT output 10: PG01 11: —		00: PORT input 01: PORT output 10: PG00 11: TXD0	

Port 6 Function Register H

P6CRH (0016H)

	7	6	5	4	3	2	1	0
bit Symbol	P67C1	P66C0	P66C1	P66C0	P65C1	P65C0	P64C1	P64C0
Read/Write	W		W		W		W	
After reset	0	0	0	0	0	0	0	0
Function	00: PORT input 01: PORT output 10: PG13 11: $\overline{WDTOUT}$		00: PORT input 01: PORT output 10: PG12 11: —		00: PORT input 01: PORT output 10: PG11 11: —		00: PORT input 01: PORT output 10: PG10 11: —	

Read-modify-write is prohibited for registers P6CR and P6FC.

Note: To set the TXD0 pin to open drain output, write 1 in bit 0<ODE0> in the ODE register. There is no port/function switch register for pin P61/RXD0. If pin P61 is used as an input port, data are input as serial receive data to SIO. When pin P66/PG12/INT6 is used for INT6, set P6CRH<P66C1,0> to 00 and IIMC1<161E> to 1.

Figure 3.5 (16). Registers for Port 6

3.5.8 Port 7 (P70 - P76)

Port 7 is a 7-bit general-purpose I/O port. I/O can be set bit by bit using control registers P7CRL and P7CRH. Resetting sets all bits in P7 to 1; control registers P7CRL and P7CRH to 0; P70 - P76 to input mode. In addition to functioning as a general-purpose I/O port, port 7 as follows: interrupt input, timer I/O, DRAM address multiplex, serial channel send/receive (TXD1 and RXD1), and transfer clock input (SCLK1) pin.

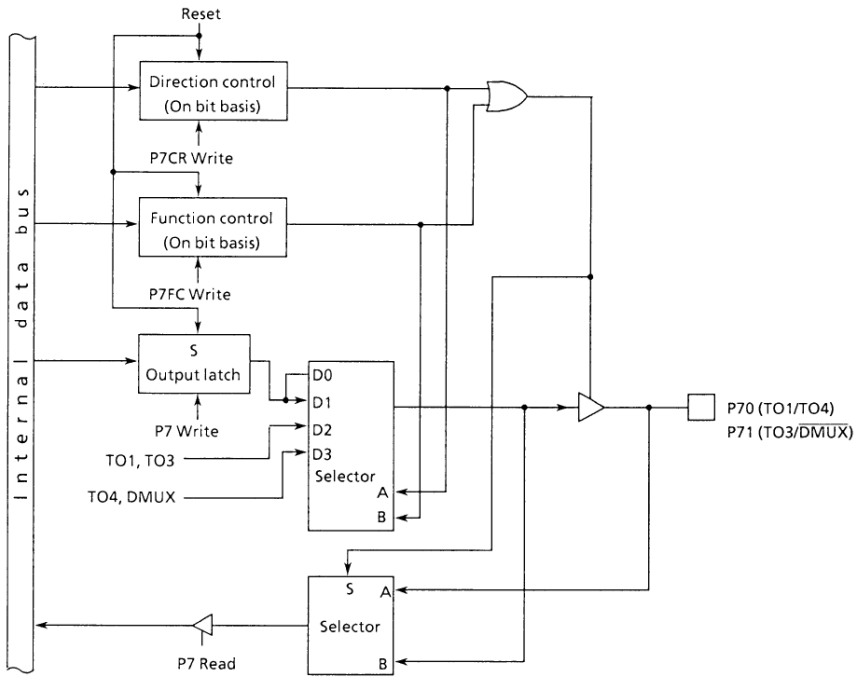


Figure 3.5 (17). Port 7 (P70, P71)

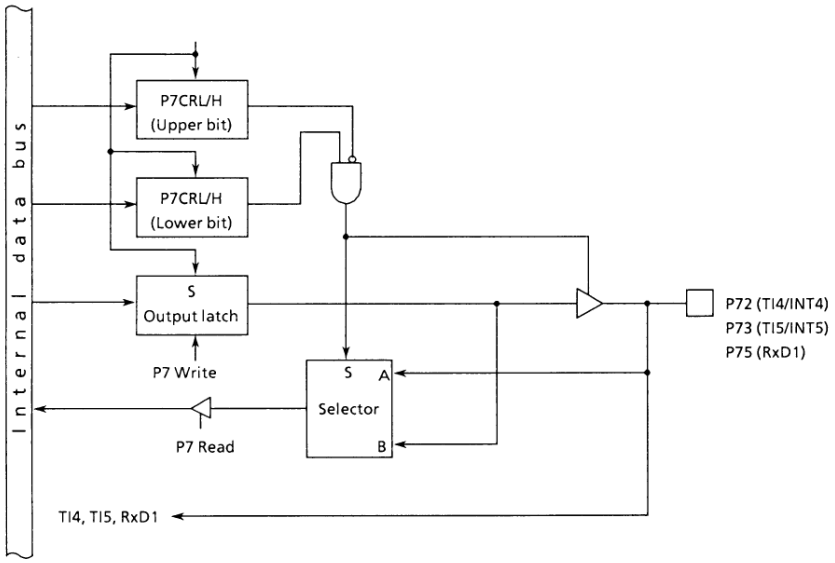


Figure 3.5 (18). Port 7 (P72, P73, P75)

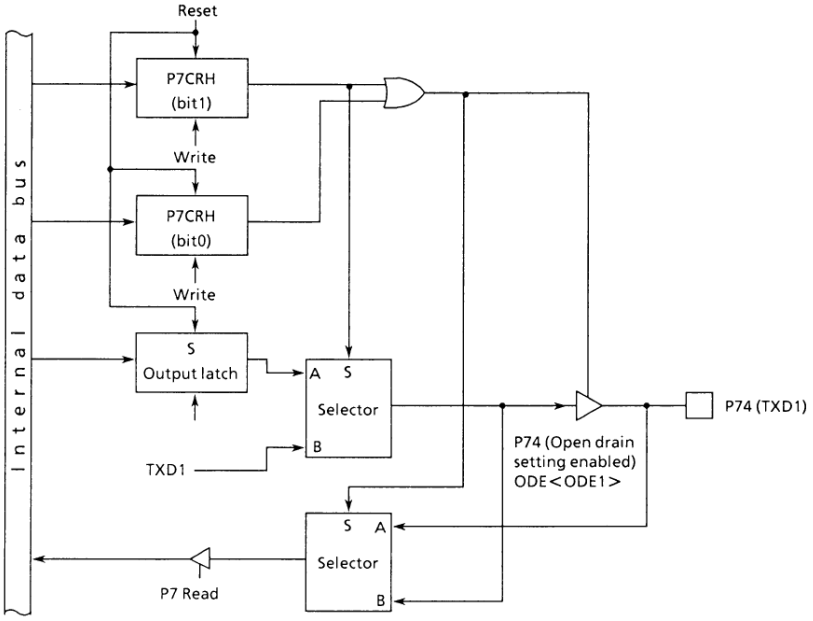


Figure 3.5 (19). Port 7 (P74)

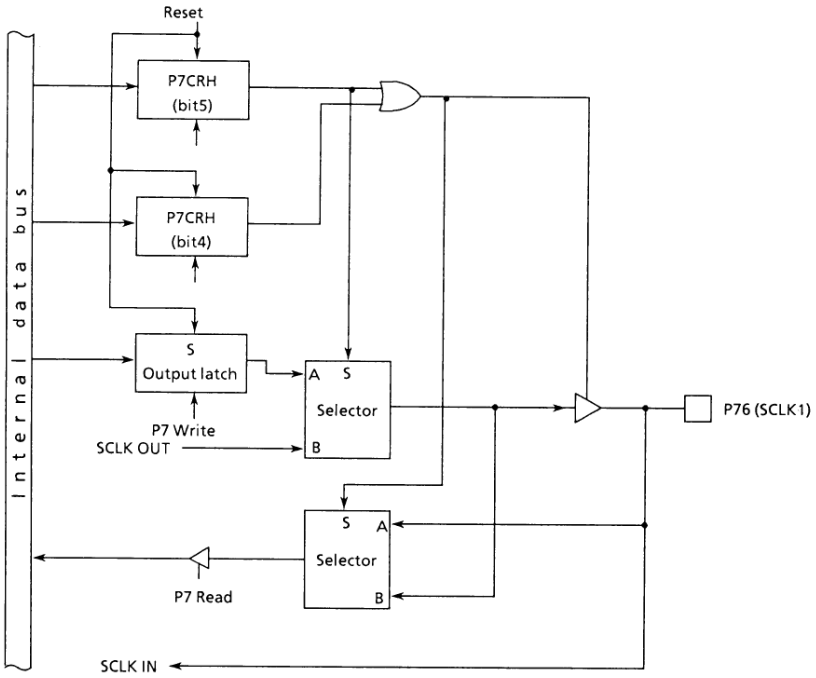


Figure 3.5 (20). Port 7 (P76)

Port 7 Register

P7 (0013H)	7	6	5	4	3	2	1	0
bit Symbol		P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W							
After reset	Input mode							
Function		1	1	1	1	1	1	1

Port 7 Control Register L

P7CRL (0015H)	7	6	5	4	3	2	1	0
bit Symbol	P73C1	P73C0	P72C1	P72C0	P71C1	P71C0	P70C1	P70C0
Read/Write	W		W		W		W	
After reset	0	0	0	0	0	0	0	0
Function	00: PORT input 01: PORT output 10: — 11: —		00: PORT input 01: PORT output 10: — 11: —		00: PORT input 01: PORT output 10: TO3 11: $\overline{DMUX}$		00: PORT input 01: PORT output 10: TO1 11: TO4	

Port 7 Control Register H

P7CRH (0017H)	7	6	5	4	3	2	1	0
bit Symbol			P76C1	P76C0	P75C1	P75C0	P74C1	P74C0
Read/Write			W		W		W	
After reset			0	0	0	0	0	0
Function			00: PORT input 01: PORT output 10: SCLK1 11: —		00: PORT input 01: PORT output 10: — 11: —		00: PORT input 01: PORT output 10: TxD1 11: —	

Read-modify-write is prohibited for registers P7CR and P7FC.

Note : To set the TxD1 pin to open drain output, write 1 in the 1<ODE1> in the ODE register.
There is no port/function switch register for pin P75/RXD1. If pin P75 is used as an input port, data are input as serial receive data to SIO. There is no port/function switch register for pin P72/T14/INT4 or pin P73/T15/INT5. If pin P72 or P73 is used as an input port, data are input to the 16-bit timer. When pin P72/P73 is used for INT4/5, set P7CRL<P73C1,0> to 00 and I1MC0<141E>/I1MC1<151E> to 1.

Figure 3.5 (21). Registers for Port 7

3.6 Chip Select/Wait Control

The TMP96C031F has a built-in chip select/wait controller used to control chip select ($\overline{CS0}$ - $\overline{CS3}$ pins), wait ($\overline{WAIT}$ pin), and data bus size (8 or 16 bits) for any of the three block address areas.

The select pin (AM8/$\overline{16}$) is used to select the width of the external data bus. (See section 3.1.2, External data width select pin.)

3.6.1 Control Registers

Table 3.6 (1) shows control registers

The block address areas is controlled by corresponding CS/wait control register (B0CS, B1CS, B2CS, B3CS) and start address register/address mask register (explained in section 3.6.2, Address area).

Registers can be written to only when the CPU is in system mode. (There are two CPU modes: system and normal.) The reason is that the settings of these registers have an important effect on the system.

Table 3.6 (1) Chip Select/Wait Control Register

		7	6	5	4	3	2	1	0
B0CS (0068H)	bit Symbol	B0E	B0SYS	B0ARE	B0BUS	B0W1	B0W0	BEXW1	BEXW0
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: $\overline{CS0}$ DIS 1: $\overline{CS0}$ EN	1: SYSTEM ONLY	0: 7F00H ~7FFFH 1: address area specification	0: 16BIT 1: 8BIT	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT		00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT	
B1CS (0069H)	bit Symbol	B1E	B1SYS	B1ARE	B1BUS	B1W1	B1W0		
	Read/Write	W							
	After reset	0	0	0	0	0	0		
	Function	0: $\overline{CS1}$ DIS 1: $\overline{CS1}$ EN	↑	0: 80H ~7FFFH 1: address area specification	↑	↑	↑	—	—
B2CS (006AH)	bit Symbol	B2E	B2SYS	B2ARE	B2BUS	B2W1	B2W0		
	Read/Write	W							
	After reset	1	0	0	Undefined	0	0		
	Function	0: $\overline{CS2}$ DIS 1: $\overline{CS2}$ EN	↑	0: 8000H ~ 3FFFFFH 1: address area specification	↑	↑	↑	—	—
B3CS (006BH)	bit Symbol	B3E	B3SYS	B3ARE	B3BUS	B3W1	B3W0	B3CAS	SRFC
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	1
	Function	0: $\overline{CS3}$/$\overline{CAS}$ DI 1: $\overline{CS3}$/$\overline{CAS}$ EN	↑	0: Undefined 1: address area specification	↑	↑	↑	0: $\overline{CS3}$ output 1: $\overline{CAS}$/ output	0: Self refresh execution 1: Release

(1) Enable

Control register bit 7 (B0E, B1E, B2E, and B3E) 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) System only specification

Control register bit 6 (B0SYS, B1SYS, B2SYS, and B3SYS) is used to specify enable/disable of the setting depending on the CPU operating mode (system or normal). Setting this bit to 0 enables setting (Address space for $\overline{CS}$, Wait state, Bus size, etc.) regardless of the CPU operating mode; setting it to 1 enables setting in system mode but disables setting in normal mode.

Resetting clears bit 6 to 0.

Bit 6 is mainly used when external memory data should not be accessed in normal mode (i.e., for system mode only memory data for the operating system).

(3) Address area specification

Control register bit 5 (B0ARE, B1ARE, B2ARE, and B3ARE) is used to specify the target address space. When this bit is set to “0” after reset, $\overline{CS0}$ is set to addresses 7F00H to 7FFFF, $\overline{CS1}$ is set to address 80H to 7FFFH, and $\overline{CS2}$ is set to addresses 8000H to 3FFFFF. $\overline{CS3}$ is undefined. (See 3.6.3 Default Address Space Specification.) When this bit is set to “1”, the target address is the address space is the address space specified by the memory start address register MSAR and memory start address mask register MAMR. (See 3.6.2 Address Space Specification.)

(4) Data bus width select

Control register bit 4 (B0BUS, B1BUS, B2BUS, B3BUS) is used to specify the data bus width. When this bit is set to “0”, memory is accessed in 16-bit data bus mode. When this bit is set to “1”, memory is accessed in 16-bit data bus mode. However, this bit is valid only in 16-bit bus mode (AM8/$\overline{16}$ pin = “0”). In 8-bit bus mode (AM8/$\overline{16}$ pin = “1”), all address space is accessed in 8-bit data bus mode regardless of the value of this bit. (See 3.1.2 External Data Bus Width Selection Pin.)

The changing data bus width according to the address to be address to be accessed is referred to dynamic bus sizing. Table 3.6 (2) shows the details of the bus operation.

Table 3.6 (2) Dynamic Bus Sizing

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

Control register bits 3 and 2 (B0W1, 0; B1W1, 0; B2W1, 0; B3W1, 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: If there ia a contention between DRAM access and refresh when using DRAM, the refresh cycle is added to the specified wait.

(6) CS/CAS waveform select

The B3CS register bit 1 <B3CAS> is used to specify the mode of the waveform output from the chip select pin ($\overline{CS3}/\overline{CAS}$) pin. When this bit is set to "0", $\overline{CS3}$ waveform is output. When it is set to "1", $\overline{CAS}$ waveform is output. This bit is cleared to zero after reset.

(7) Self refresh control

(described in section 3.13.1 Refresh Controller.)

(8) Wait control outside space $\overline{CS0}$ to $\overline{CS3}$

This bit is used to specify the number of waits when B0CS register bits 1 and 0 <BEXW1, 0> or space outside $\overline{CS0}$ to $\overline{CS3}$ is accessed.

3.6.2 Address Space Specification (B0CS to B3CS < B0ARE to B3ARE> = “1”)

The address space is specified with the start address register (MSAR0, MSAR1, MSAR2, and MSAR3) and address mask register (MAMR0, MAMR1, MAMR2, and MAMR3). For each bus cycle, the chip select controller compares the address on the bus and value of this start address register. The value of the address mask register is used to ignore result of this address comparison. When there is a match, the specified space is assumed to be accessed and a low strobe signal is output from the corresponding chip select pin ($\overline{CS0}$ to $\overline{CS3}$) if it is enabled (B0E to B3E = “1”).

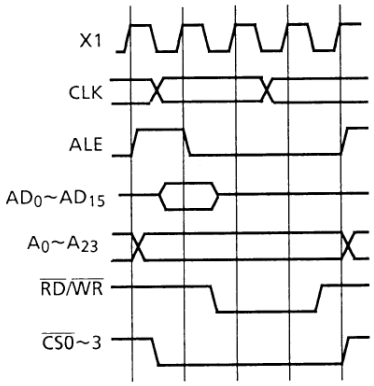


Figure 3.6 (1). Chip Select ($\overline{CS0}$ to $\overline{CS3}$) Operation Timing

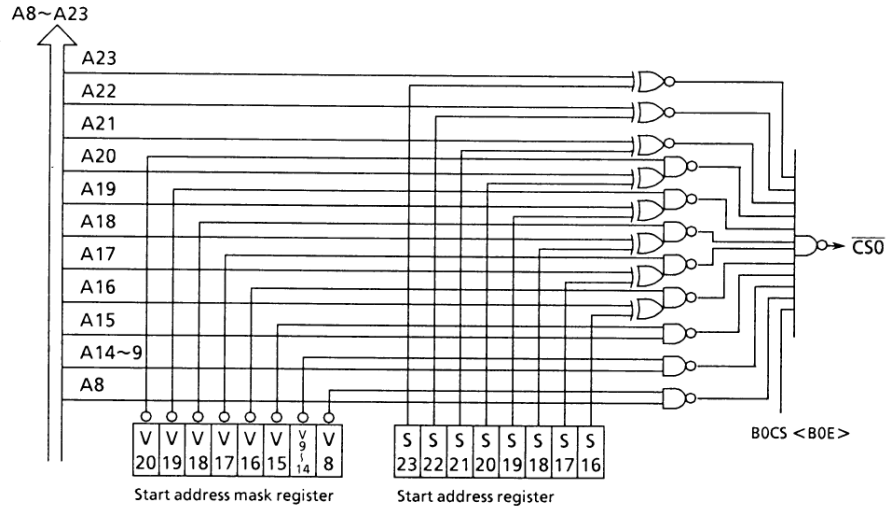


Figure 3.6 (2). $\overline{CS0}$ Address Decode Block Diagram

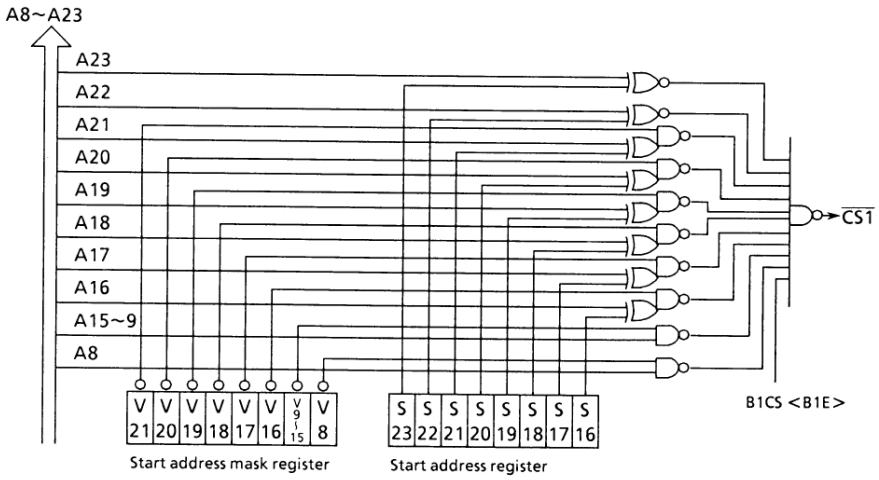


Figure 3.6 (3). $\overline{CS1}$ Address Decode Block Diagram

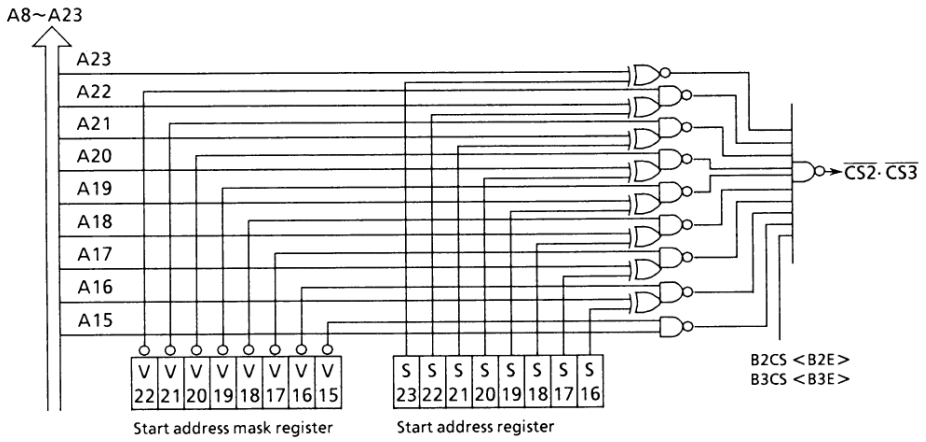


Figure 3.6 (4). $\overline{CS2}$, $\overline{CS3}$ Address Decode Block Diagram

(1) Memory start address register Memory start address register

Table 3.6 (3) Memory Start Address Register

Memory start address register ($\overline{CS0}$ to $\overline{CS3}$)

		7	6	5	4	3	2	1	0
MSAR0 (0040H) / MSAR1 (0042H)	bit symbol	S23	S22	S21	S20	S19	S18	S17	S16
	Read/Write	R/W							
MSAR2 (0044H) / MSAR3 (0046H)	After reset	1	1	1	1	1	1	1	1
	Function	Set start addresses A23 to A16							

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

Table 3.6 (4) Memory Start Address Mask Registers

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

MAMR0 (0041H)

	7	6	5	4	3	2	1	0
bit symbol	V20	V19	V18	V17	V16	V15	V14~9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0 : Compare enabled 1 : Compare disabled							

→ Control comparison of $\overline{CS0}$ addresses A8 to A20

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

MAMR1 (0043H)

	7	6	5	4	3	2	1	0
bit symbol	V21	V20	V19	V18	V17	V16	V15~9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0 : Compare enabled 1 : Compare disabled							

→ Control comparison of $\overline{CS1}$ addresses A8 to A21

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

MAMR2 (0045H) / MAMR3 (0047H)

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

→ Control comparison of $\overline{CS2}$ to $\overline{CS3}$ addresses A15 to A22

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

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

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

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

$\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$ can be used in the same manner.

(2) How to 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 due to the value of the MAMR.

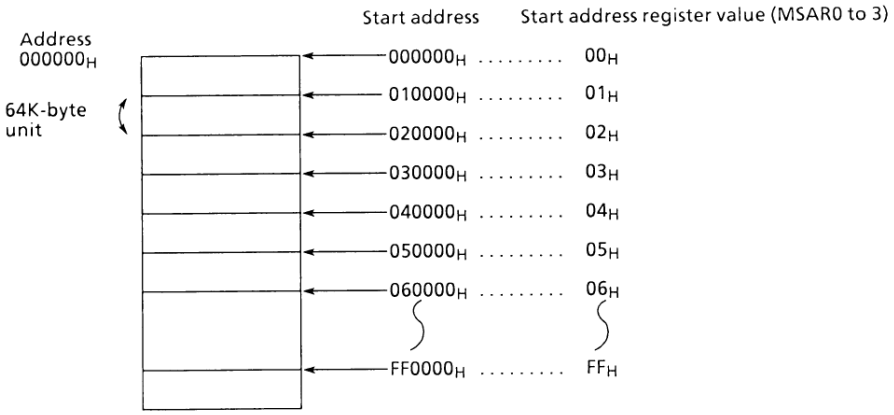


Figure 3.6 (5). Where to Set Start Address

(3) How to Set the Address Space

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

As shown in the address decoder block diagram (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 address A8 to A20, A8 to A21, or A15 to A22 respectively.

$\overline{CS}$ \ SIZE	256	512	32K	64K	128K	256K	512K	1M	2M	4M	8M
$\overline{CS0}$	○	○	○	○	○	○	○	○	○		
$\overline{CS1}$	○	○		○	○	○	○	○	○	○	
$\overline{CS2}$			○	○	○	○	○	○	○	○	○
$\overline{CS3}$			○	○	○	○	○	○	○	○	○

Figure 3.6 (6). Chip Select and Space Size

(4) Start Address/Address Space Setting Procedure

① Set memory start address mask register (MAMR)
(Set address space)

② Set memory start address register (MSAR)
(Set area start address)

③ Check the identical address bit of MAMR and MSAR

Example: Check the value of ($\overline{CS0}$) MAMR0 <V16> and MSAR0 <S16>

④ If the bits at identical address are "1" and "1", MSAR bit is treated as "0". <-The start address changes.

Example: If ($\overline{CS0}$) MAMR <V16> = 1 and MSAR <S16> = 1, comparison of address A16 and <S16> is disabled and address A16 is selected regardless of whether the value is "1" or "0" and the start address is replaced by the value in MSAR.

⑤ If it is OK for the start address to change, end the setting procedure. If not, change the value to MSAR.

⑥ Reset MSAR and re-verify (return to step 3).

(Setting Example)

When address space is 128K-byte and start address is 30000H (area 30000H to4FFFFH).

Set

MAMR = 0FH address space 128K-byte
MSAR = 03H start address 30000H

MAMR <V16> and MSAR <S16> are "1" and "1" and the start address changes to 2000H. (space 20000H to 3FFFFH).
If this is not desired, change the start address.
Change the start address to 4000H. (space 40000H to 5FFFFH).

MAMR = 0FH
MSAR = 04H

The bits at identical address of MAMR and MSAR are not "1" and "1" and the start address remains unchanged. Therefore, a 128K-byte space starting at address 40000H can be decoded.

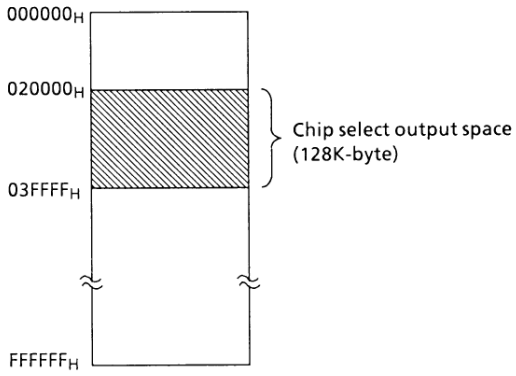
(Setting example 1) ($\overline{CS0}$) chip select output is as shown in the following memory map.

When MSAR is set to 02H and MAMR is set to 0FH, the

	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14~9	S8
MSAR setting	0	0	0	0	0	0	1	0	0	0	0
									default "0"		

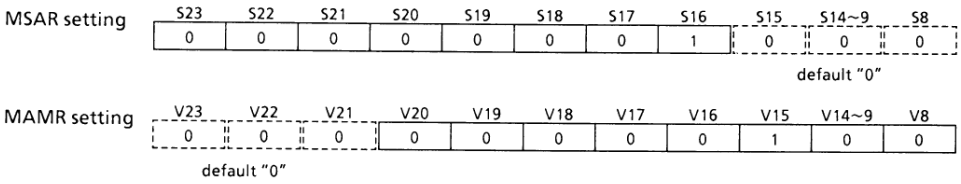
	V23	V22	V21	V20	V19	V18	V17	V16	V15	V14~9	V8
MAMR setting	0	0	0	0	0	0	0	1	1	1	1
	default "0"										

S23 to S17 are valid because V23 to V17 = "0" and S16 are invalid because V16 to V8 = "1".



(Setting example 2) ($\overline{CS0}$)

When MSAR is set to 01H and MSAR is set to 04H, the chip select output is as shown in the following memory map.



The values of S23 - S16, and S14-S8 become valid because V23 to V16 = 0 and V14 - V8 = 0.The value of S15 becomes invalid because V15 = 1.

(Setting example 3) ($\overline{CS0}$)

Space where chip select is output by values set in MSAR and MAMR (excerpt).

MSAR \ MAMR	00	01	03	04
00	0000 ≀ 00FF (256-byte)	0000 ≀ 01FF (512-byte)	0000 ≀ 7FFF (32K-byte)	0000 ≀ 00FF (256-byte × 2) and 8000 ≀ 80FF (256-byte × 2)
01	10000 ≀ 100FF (256-byte)	10000 ≀ 101FF (512-byte)	10000 ≀ 17FFF (32K-byte)	10000 ≀ 100FF (256-byte × 2) and 18000 ≀ 180FF (256-byte × 2)
02	20000 ≀ 200FF (256-byte)	20000 ≀ 201FF (512-byte)	20000 ≀ 27FFF (32K-byte)	20000 ≀ 200FF (256-byte × 2) and 28000 ≀ 280FF (256-byte × 2)
03	30000 ≀ 300FF (256-byte)	30000 ≀ 301FF (512-byte)	30000 ≀ 37FFF (32K-byte)	30000 ≀ 300FF (256-byte × 2) and 38000 ≀ 380FF (256-byte × 2)

MAMR \ MAMR	03	07	0F	1F	3F	7F	FF
04	40000 ≀ 47FFF (32K-byte)	40000 ≀ 4FFFF (64K-byte)	40000 ≀ 5FFFF (128K-byte)	40000 ≀ 7FFFF (256K-byte)	00000 ≀ 7FFFF (512K-byte)	00000 ≀ FFFFF (1M-byte)	000000 ≀ 1FFFFF (2M-byte)
08	80000 ≀ 87FFF (32K-byte)	80000 ≀ 8FFFF (64K-byte)	80000 ≀ 9FFFF (128K-byte)	80000 ≀ BFFFF (256K-byte)	80000 ≀ FFFFF (512K-byte)		
10	100000 ≀ 107FFF (32K-byte)	100000 ≀ 10FFFF (64K-byte)	100000 ≀ 11FFFF (128K-byte)	100000 ≀ 13FFFF (256K-byte)	100000 ≀ 17FFFF (512K-byte)	100000 ≀ 1FFFFF (1M-byte)	
20	200000 ≀ 207FFF (32K-byte)	200000 ≀ 20FFFF (64K-byte)	200000 ≀ 21FFFF (128K- byte)	200000 ≀ 23FFFF (256K-byte)	200000 ≀ 27FFFF (512K-byte)	200000 ≀ 2FFFFF (1M-byte)	200000 ≀ 3FFFFF (2M-byte)
40	400000 ≀ 407FFF (32K-byte)	400000 ≀ 40FFFF (64K-byte)	400000 ≀ 41FFFF (128K-byte)	400000 ≀ 43FFFF (256K-byte)	400000 ≀ 47FFFF (512K-byte)	400000 ≀ 4FFFFF (1M-byte)	400000 ≀ 5FFFFF (2M-byte)
80	800000 ≀ 807FFF (32K-byte)	800000 ≀ 80FFFF (64K-byte)	800000 ≀ 81FFFF (128K-byte)	800000 ≀ 83FFFF (256K-byte)	800000 ≀ 87FFFF (512K-byte)	800000 ≀ 8FFFFF (1M-byte)	800000 ≀ 9FFFFF (2M-byte)

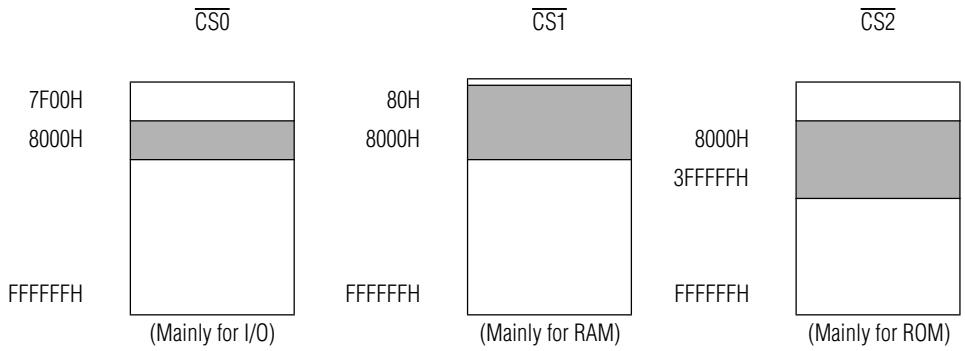
3.6.3 Default Address Space Specification (B0CS to B2CS < B0ARE to B2ARE> = “0”)

The following figures show the actual chip select image. $\overline{CS0}$ can specify 7F00H to 7FFFH, $\overline{CS1}$ can specify 80H to 7FFFH, and $\overline{CS2}$ can specify 8000H to 3FFFFFH. This is because external connection of devices (such as RAM or I/O) other than ROM is considered.

The area 7F00 to 7FFFH (256-byte space) for $\overline{CS0}$ is mapped in this space mainly due to external I/O expansion consideration.

The area 80H to 7FFFH (approximately 32K-byte space) for $\overline{CS1}$ is mapped in this space mainly due to external RAM expansion consideration.

The area 8000H to 3FFFFFH (approximately 4M-byte space) for $\overline{CS2}$ is mapped in this space mainly due to external ROM expansion consideration.



Supplement 1: The access priority is in the order of built-in I/O and chip select/wait controller.

Supplement 2: Wait for spaces other than $\overline{CS0}$ to $\overline{CS3}$ is set with B0CS register <BEXW1,0> and the data bus width is fixed to 16-bit if the AM8/$\overline{16}$ pin is “0” and to 8 bits if it is “1”.

Note: When using the chip select/wait controller, do not assign multiple definitions to the same address area. (However, if $\overline{CS0}$ is set to 7F000H to 7FFFH and $\overline{CS1}$ is set to 80H to 7FFFH, only the $\overline{CS0}$ setting/pin is active in the overlapped address space 7F00H to 7FFFH.)
When the bus is opened ($\overline{BUSAK}$ = “0”), $\overline{CS0}$ to $\overline{CS3}$ pins are also opened (output buffer OFF). Refer to the note on bus open in section “3.5 Port Functions” for the pin status at this point.

3.6.4 Example of Usage

(1) Connection example 1

Figure 3.6 (6) is an example (1) in which an external memory is connected to the TMP96C031F. In this example, a ROM is connected using 16-bit Bus; a RAM is connected using 8-bit Bus.

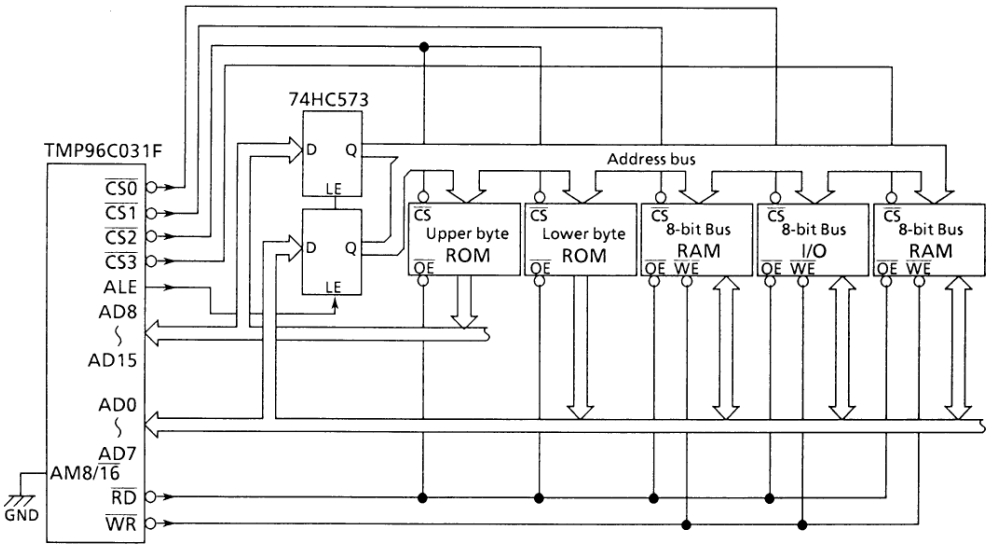


Figure 3.6 (7). Example of External Memory Connection (ROM = 16-bit, RAM and I/O = 8-bit)

After a reset, the $\overline{CS0}$ - $\overline{CS3}$ pins are set to output port mode; 1 is output from $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS3}$; 0 from $\overline{CS2}$.

The program used to set these pins is as follows:

```
P4FC    EQU     10H
B0CS    EQU     68H
B1CS    EQU     69H
B2CS    EQU     6AH
B3CS    EQU     6BH
MSAR3   EQU     46H
MAMR3   EQU     47H
LD      (B0CS),     10010000B   ; CS0 =  8-bit, 2WAIT, 7F00H ~ 7FFFH, 2WAIT other than in CS0 ~ CS3 areas
LD      (B1CS),     100111XXB   ; CS1 =  8-bit, 0WAIT, 80H ~ 7FFFH
LD      (B2CS),     100001XXB   ; CS2 = 16-bit, 1WAIT, 8000H ~ 3FFFFFH
LD      (B3CS),     10111100B   ; CS3 = 8-bit, 0WAIT, address area specification (400000H ~ 407FFFH)
LD      (MSAR3),    01000000B   ; CS3 start address: 400000H
LD      (MAMR3),    00000000B   ; CS3 area = 32K-byte
LD      (P4FC),     XXXX1111B   ; CS0 ~ CS3 output mode
Note:   X: don't care
```

(2) Connection example 2

Figure 3.6 (7) is an example (2) in which an external memory is connected to the TMP96C031. In this example, the ROM, RAM, and I/O are connected with 8-bit width.

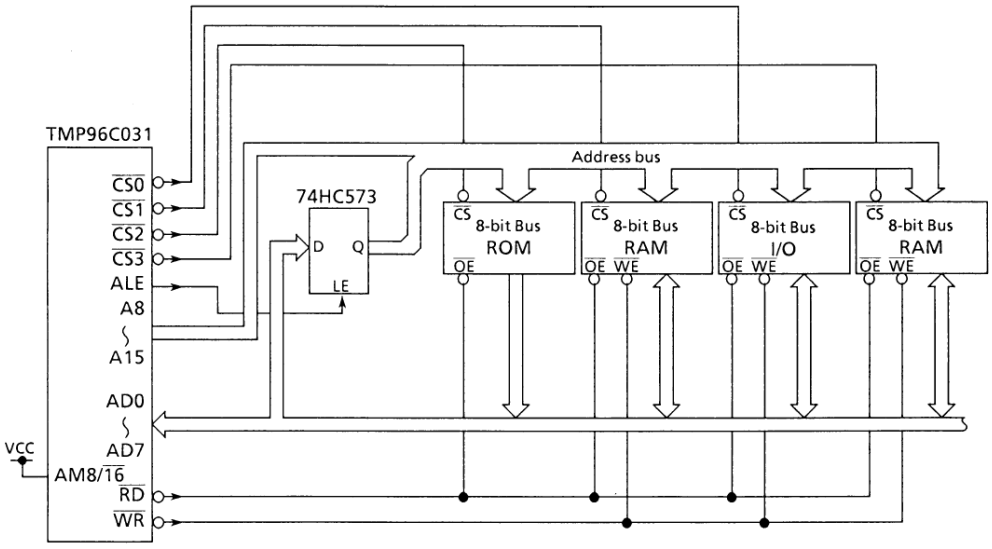


Figure 3.6 (8). Example of External Memory Connection (ROM = 16-bit, RAM and I/O = 8-bit)

After a reset, the $\overline{CS0}$ - $\overline{CS3}$ pins are set to output port mode; 1 is output from $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS3}$; 0 from $\overline{CS2}$.

The program used to set these pins is as follows:

```
P4FC    EQU     10H
B0CS    EQU     68H
B1CS    EQU     69H
B2CS    EQU     6AH
B3CS    EQU     6BH
MSAR3   EQU     46H
MAMR3   EQU     47H
LD      (B0CS),     10010000B   ; CS0 = 8-bit, 2WAIT, 7F00H ~ 7FFFH, 2WAIT other than in CS0 ~ CS3 areas
LD      (B1CS),     100111XXB   ; CS1 = 8-bit, 0WAIT, 80H ~ 7FFFH
LD      (B2CS),     100001XXB   ; CS2 = 16-bit, 1WAIT, 8000H ~ 3FFFFFH
LD      (B3CS),     10111100B   ; CS3 = 8-bit, 0WAIT, address area specification (400000H ~ 407FFFH)
LD      (MSAR3),    01000000B   ; CS3 start address: 400000H
LD      (MAMR3),    00000000B   ; CS3 area = 32K-byte
LD      (P4FC),     XXXX1111B   ; CS0 ~ CS3 output mode
Note:   X: don't care
```

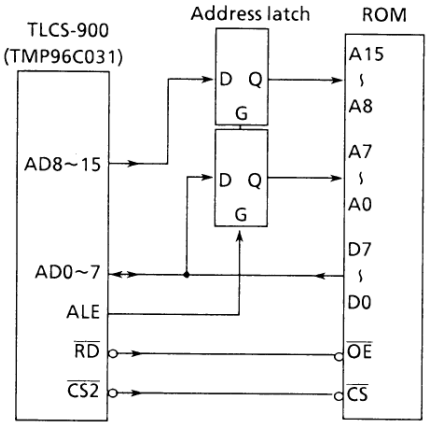
3.6.5 How to Start with an 8-Bit Data Bus (with AM8/$\overline{16}$ = "0")

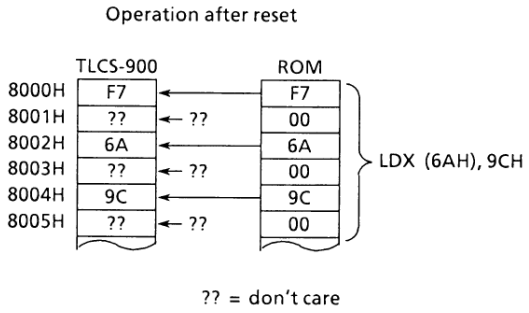
Resetting sets the $\overline{CS2}$ pin low due to an internal pull-down resistor; memory access starts in 16-bit data bus (2-wait) mode. To start in 8-bit data bus mode, a special operation is required. Operation is as described in the example below:

```
B2CS    EQU     6AH             ; CS2 register address
        ORG     8000H           ; RESET address
        LDX     (B2CS), 9CH     ; CS2 8-bit, 0WAIT, 8000H ~
```

After reset, the program reads the LDX (B2CS), 9CH instruction in 16-bit data bus mode. LDX is a 6-byte instruction: the 2nd, 4th and 6th bytes are handled as dummies (i.e., only codes in the 1st, 3rd and 5th bytes are actually used). Even if starting in 8-bit data bus mode, it is possible to program so that the LDX instruction is executed and the block 2 area (8000H - 3FFFFFH) is accessed in 8-bit data bus mode without any problem.

The above program does not include setting the P42/$\overline{CS2}$ pin to output; add a program to set the P4CR and P4FC registers as required.





3.7 8-bit Timers

TMP96C031F 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 two 16-bit timers. The following four operating modes are provided for the 8-bit timers.

- 8-bit interval timer mode (4 timers)
- 16-bit interval timer mode (2 timers)

) Either two 8-bit buses or one 16-bit bus can be used.

- 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 (1 timer)

Figure 3.7 (1) shows the block diagram of 8-bit timer (timer 0 and timer 1).

Timers 2 and 3 have the same circuit configuration as timers 0 and 1. However, timer 0 has an external clock, pin TI0, whereas timer 2 does not.

Each interval timer consists of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Timer flip-flop (TFE1) is provided for timers 0 and 1; TFE3 timer 2 and 3.

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

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

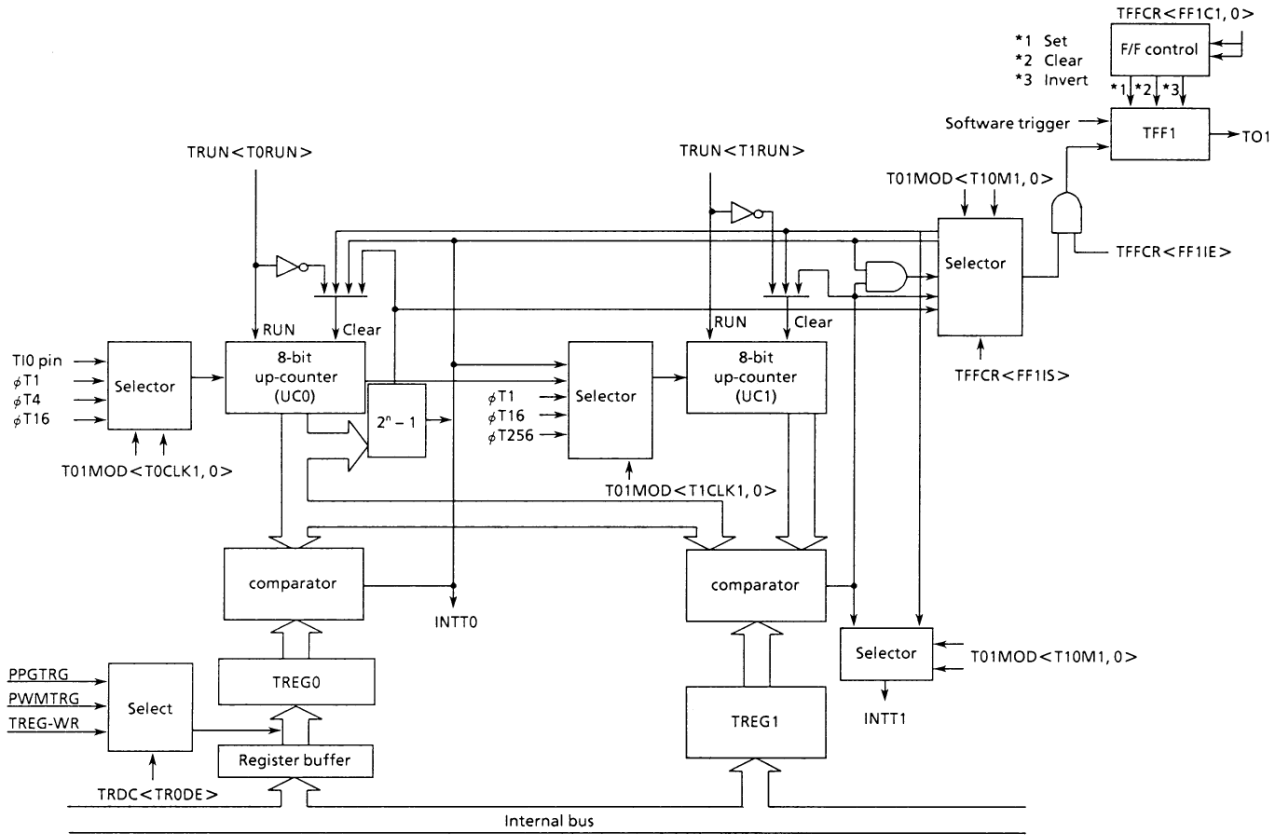


Figure 3.7 (1). Block Diagram of 8-Bit Timers (Timers 0 and 1)

① Prescaler

This 9-bit prescaler generates the clock input to the 8-bit timers, 16-bit timer/event counters, and baud rate generators by further dividing the fundamental clock (fc) after it has been divided by 4 (fc/4).

Among them, 8-bit timer uses 4 types of clock: ϕT1, ϕT4, ϕT16, and ϕT256.

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

Cycle

Input clock \ fc	20MHz
ϕT1 (8/fc)	400ns
ϕT4 (32/fc)	1.6μs
ϕT16 (128/fc)	6.4μs
ϕT256 (2048/fc)	102μs

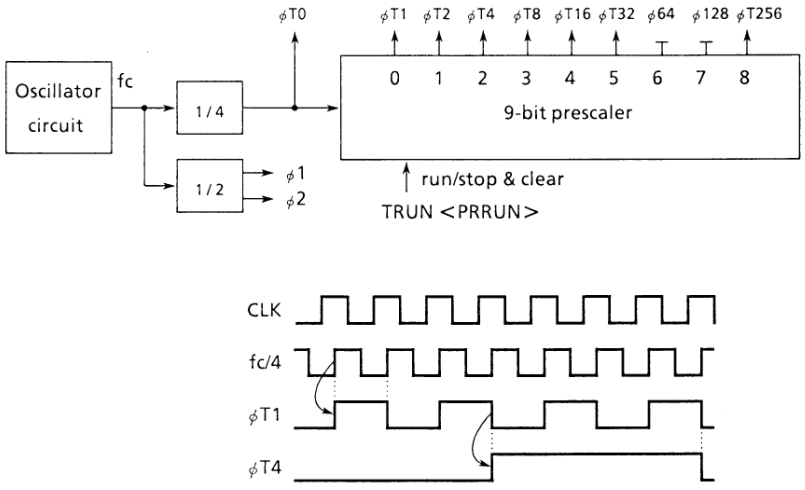


Figure 3.7 (2). Prescaler

② Up-counter

This is an 8-bit binary counter counted by an input clock specified by mode register T01MOD for timers 0 and 1, or mode register T23MOD for timers 2 and 3. Input clocks for timer 0 or 2 can be selected from internal clocks ϕT1, ϕT4, and ϕT16 depending to the value set in the TI0 pin can also be selected.

The input clock of timer 1 or 3 depends on the operation mode; in 16-bit timer mode, timer 0/2 overflow output is used as the output clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (match detection signal) of timer 0 according to the set value of T01MOD register or T23MOD.

Example: When T01MOD <T01M1,0> = 01, the over flow output of timer 0 becomes the input clock of timer 1 (16-bit timer).

When T01MOD7, 6 = 00, T01MOD3, 2 = 01, ϕT1 (8/fc) becomes the input of timer 1 (8-bit timer).

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

③ Timer register

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

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

TREG0/TREG2 is used to control enable/disable of the double buffers according to the timer register double buffer control register, TRDC <TR0DE, TR2DE>. 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 transferred from the register buffer to the timer register when the 2^n - 1 overflow occurs in PWM mode, or at the PPG cycle in PPG mode.

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

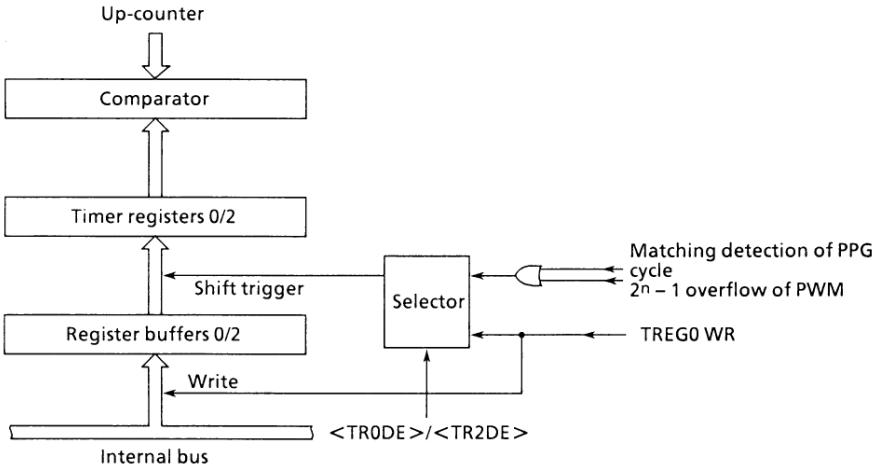


Figure 3.7 (3). Configuration of Timer Register 0/2

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 registers are write-only and cannot be read.

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: $2^6 - 1$ PWM 10: $2^7 - 1$ cycle 11: $2^8 - 1$		00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 0

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

Input clock of timer 1

	T01MOD7, 6 ≠ 01	T01MOD7, 6 = 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 in 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 3.7 (4). Timer 0, 1 Mode 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

	T01MOD7, 6 ≠ 01	T01MOD7, 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 PWM2 cycle
("Don't care" except in PWM mode)

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

Set the operation mode of timer 2 and 3

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

Figure 3.7 (5). Timer 2,3 Mode Register (T23MOD)

TFFCR (0025H)

	TFF3				TFF1			
	7	6	5	4	3	2	1	0
bit Symbol	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read/Write	W		R/W		W		R/W	
After reset	–		0	0	–		0	0
Function	00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		1: TFF3 Invert Enable	0: Inversion of 8-bit timer 2 1: Inversion of timer 3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		1: TFF1 Invert Enable	0: Inversion of 8-bit timer 0 1: Inversion of timer 1

Select inverse signal of timer F/F (TFF1)
("Don't care" except in 8-bit timer mode)

0	Inversion of timer 0
1	Inversion of timer 1

Inversion of Timer F/F (TFF1)

0	Disable
1	Enable

Control of Timer F/F (TFF1)

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

Select inverse signal of timer F/F (TFF3)
("Don't care" except in 8-bit timer mode)

0	Inversion of timer 2
1	Inversion of timer 3

Inversion of Timer F/F (TFF3)

0	Disable
1	Enable

Control of Timer F/F (TFF3)

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

Figure 3.7 (6). 8-Bit Timer Flip-Flop Control Register (TFFCR)

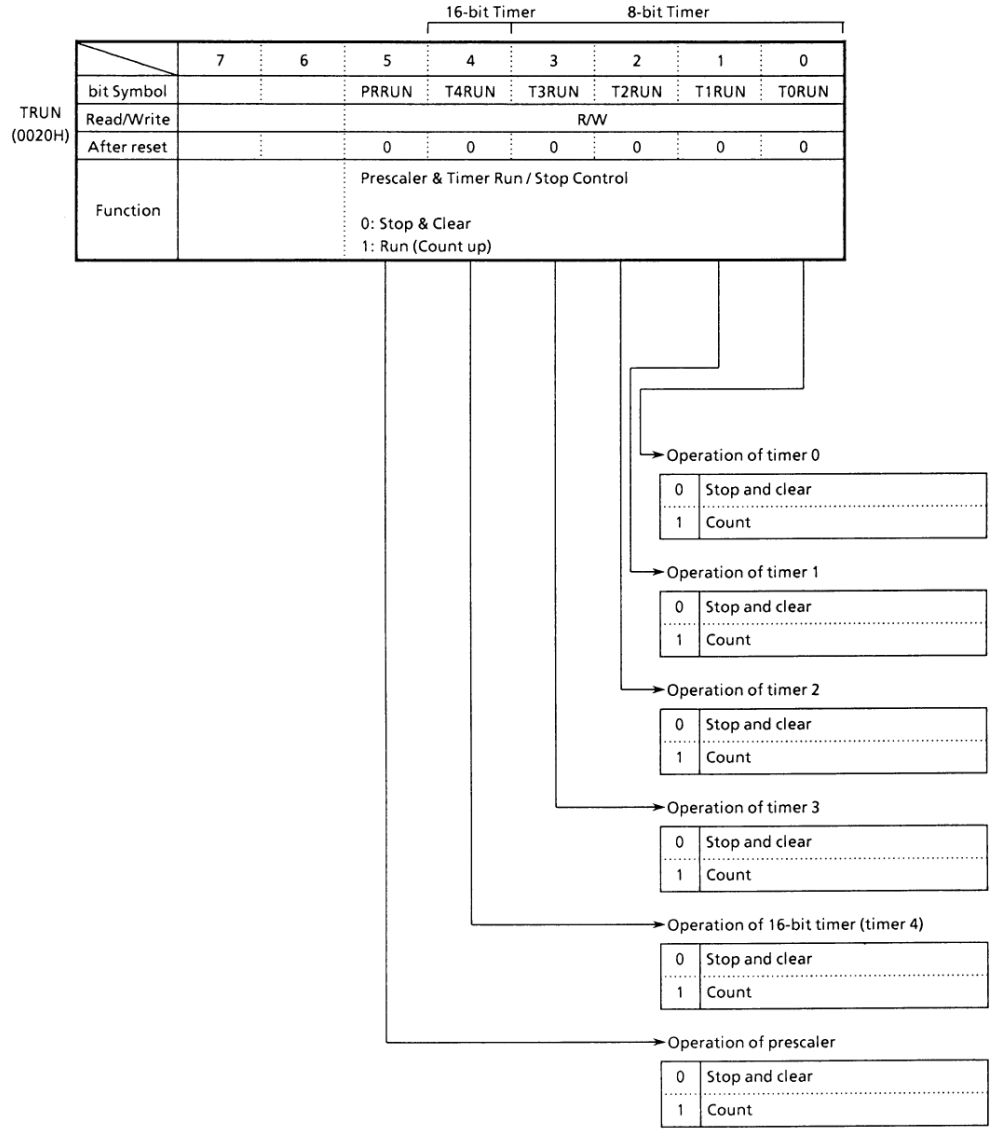


Figure 3.7 (7). Timer Operation Control Register (TRUN)

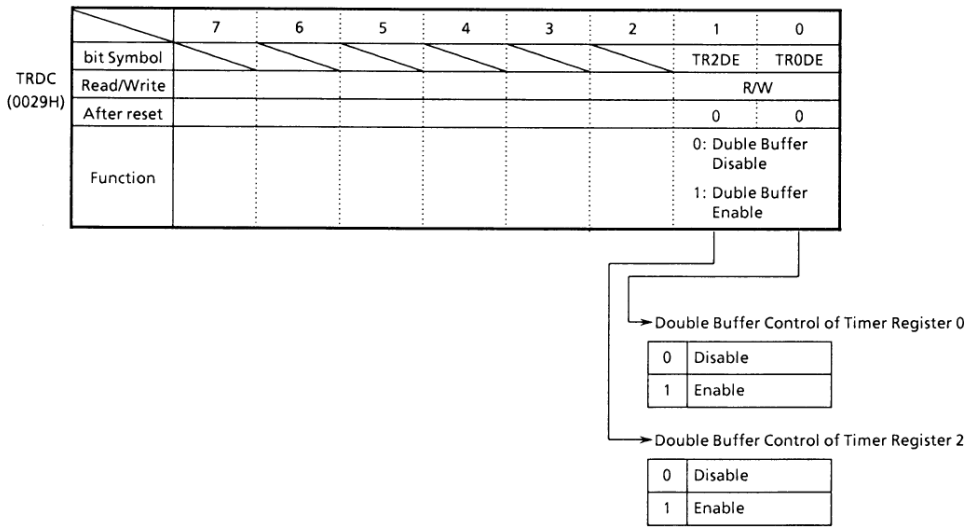


Figure 3.7 (8). Timer Register Double Buffer Control Register (TRDC)

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, INTT1, INTT2, INTT3) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

⑤ Timer flip-flops (timer F/F)

The timer flip-flops are inverted according to the interval timer match detect signal (comparator output). The signal can output a value to the timer output pins TO1 (also used as P70) and TO3 (also used as P71).

There are two timer flip-flops: TFF1 for timers 0 and 1; TFF3 for timers 2 and 3. TFF1 is output to the TO1 pin; TFF3 to the TO3 pin.

TO3 (also used as P71) is multiplexed using the $\overline{\text{DMUX}}$ pin; setting must be done using the port 7 control registers (P7CRL and P7CRH).

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 40 microseconds at fc = 16MHz, set each register in the following manner.

		MSB 7	6	5	4	3	2	1	LSB 0	
TRUN	←	–	–	–	–	–	–	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 (0.5μs @ fc = 16MHz) as the input clock.
TREG1	←	0	1	1	0	1	0	0	0	Set the timer register at 40μs ϕT1 = 50H.
INTET10	←	1	1	0	1	–	–	–	–	Enable INTT1, and set it to “Level 5”.
TRUN	←	x	x	1	–	–	–	1	–	Start timer 1 counting.

Note: x; don’t care –; no change

Use the following table for selecting the input clock.

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

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

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

		MSB 7	6	5	4	3	2	1	LSB 0	
TRUN	←	–	–	–	–	–	–	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 as the input clock.
TREG1	←	0	0	0	0	0	0	1	1	Set the timer register at 2.4μs ÷ ϕT1 ÷ 2 = 3.
TFFCR	←	–	–	–	–	1	0	1	1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P7CRL	←	–	–	–	–	–	–	1	0	Select P71 as TO1 pin.
TRUN	←	x	x	1	–	–	–	1	–	Start timer 1 counting.

Note: x; don't care –; no change

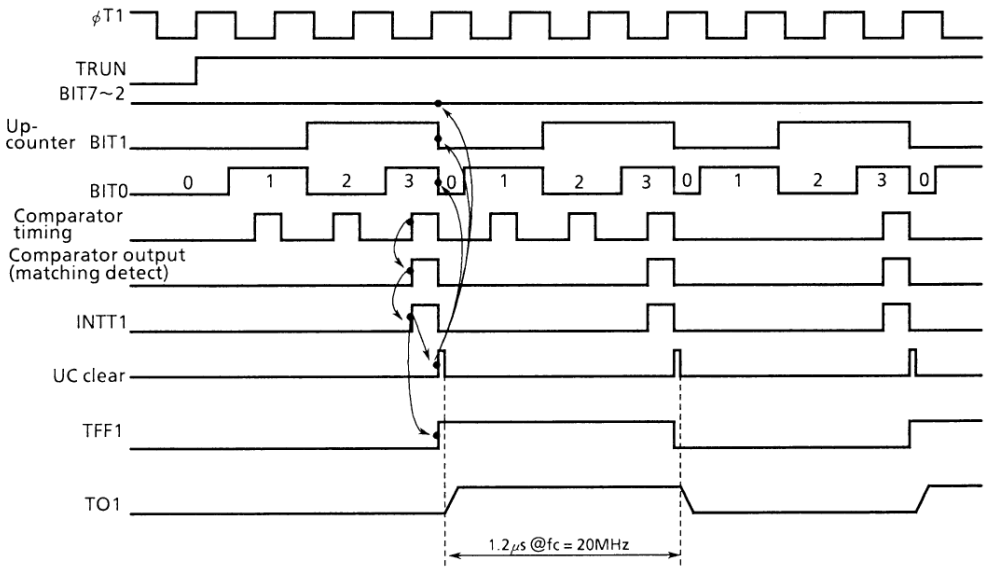


Figure 3.7 (9). Square Wave (50% Duty) Output Timing Chart

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.

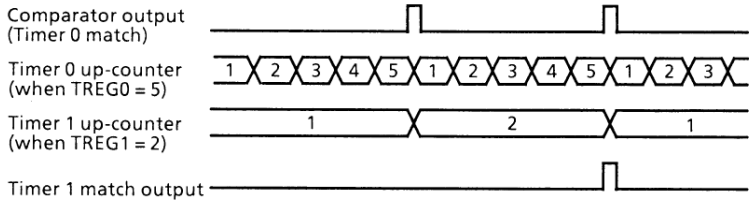


Figure 3.7 (10). Timer 1 Count Up by Timer 0

④ Output inversion with software

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

Writing “00” into TFFCR <TFF1C1, 0> inverts the value of TFF1, writing “00” into TFFCR <FF3C1, 0> inverts the value of TFF3.

⑤ Initial setting of timer flip-flop (Timer F/F)

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

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

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

(2) 16-bit timer mode

A 16-bit interval timer is configured by combining timers 0 and 1, or timers 2 and 3.

Timers 0 and 1 combined function the same as timers 2 and 3. A combination of timers 0 and 1 is used for explanation here.

To configure a 16-bit timer by cascade-connecting timers 0 and 1, set the mode register, T01MOD <T10M1,0>, to 00.

Setting 16-bit timer mode the input clock for timer 1 to timer 0 overflow output regardless of the value set in the clock control register, TCLK.

Table 3.7 (2) 16-Bit Timer (Interrupt) and Input Clock

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

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

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

When counting with input clock of ϕT16 (8μs @ 16MHz) 0.4 sec ÷ 4μs = 62500 = F424H

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

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

INT0 is not to be generated at this time, either.

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.

	Timer 0			Timer 1		
	INT T0	TO1	Match value	INT T1	TO1	Match value
16-bit timer mode (counts up timer 1 by timer 0 overflow.)	Interrupt generation	Output enable	TREG0 (counts up even at match)	Interrupt generation	Output enable	TREG1•2^8 + TREG0 (full 16-bit)
8-bit timer mode (counts up timer 1 by timer 0 match)	Interrupt generation	Output enabled (either timer 0 or 1)	TREG0 (clears at match)	Interrupt generation	Output enabled (either timer 0 or 1)	TREG1• TREG0 (multiplied value)

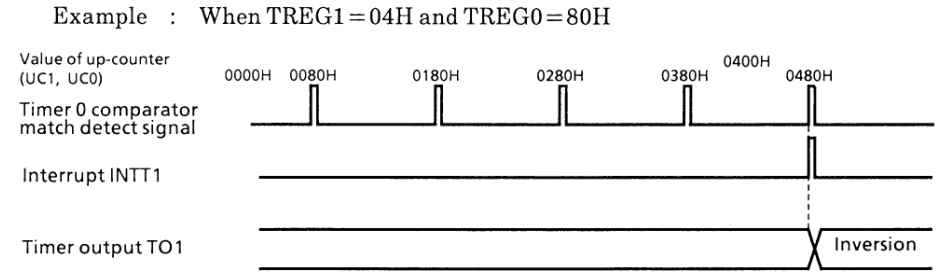
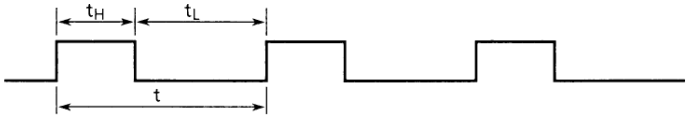


Figure 3.7 (11). Output Timer by 16-Bit Timer Mode

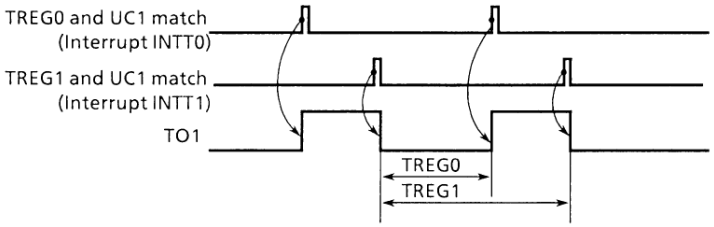
(3) 8-bit PPG (Programmable Pulse Generation) Output mode

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

With timer 0, data are output to the TO1 pin (also used as P70); with timer 2, to the TO3 pin (also used as P71).



Timer 0 is explained here because operation is the same as timer 2.



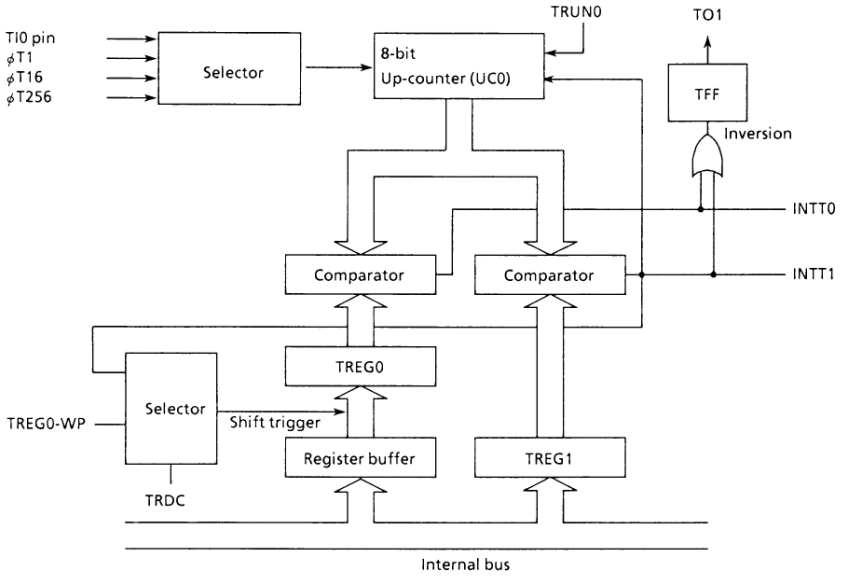
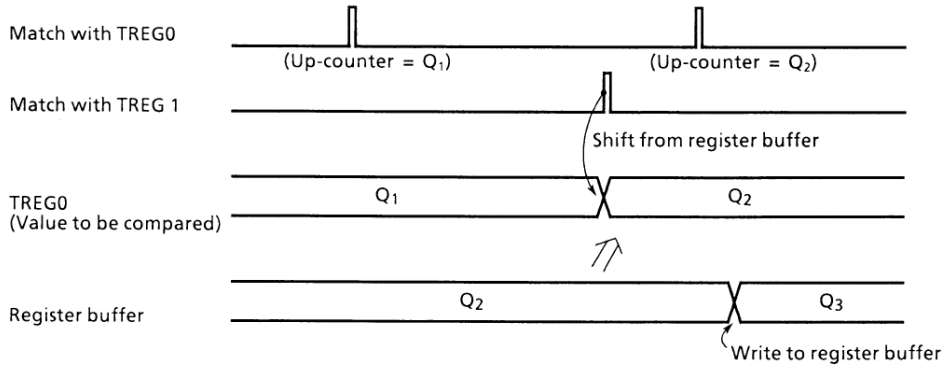


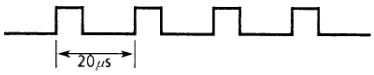
Figure 3.7 (12). Block Diagram of 8-Bit PPG Output Mode

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 handling of low duty waves (when duty is varied).



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



- Calculate the value to be set for timer register.
  To obtain the frequency 62.5kHz, the pulse cycle t should be: t = 1/62.5kHz = 16μs.
  Given ϕT1 = 0.4μs @ 20MHz),
  16μs ÷ 0.4μs = 40

Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H and then duty to 1/4, t x 1/4 = 16μs x 1/4 = 4μs

4μs ÷ 0.4μs = 10

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

		MSB 7	6	5	4	3	2	1	LSB 0	
TRUN	←	x	x	–	–	–	–	0	0	Stop timer 0, and clear it to "0".
T01MOD	←	1	0	x	x	x	x	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TFFCR	←	–	–	–	–	0	1	1	x	Sets TFF1 and enables the inversion and double buffer enable.
										Writing "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".
P7CRL	←	–	–	–	–	–	–	1	0	Set P70 as the T01 pin.
TRUN	←	x	x	1	–	–	–	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)

Mode used for timers 1 and 3. Up to 2 PWMs with a resolution of 8-bit (PWM1 and PWM3) pulse can be output.

With timer 1, PWM is output to the TO1 pin (also used as P70); with timer 3, to the TO3 pin (also used as P71).

Timer 0 or 2 is used as an 8-bit timer.

Timer 1 (PWM1) is explained here because the operation is the same as timer 3.

Timer output is inverted when up-counter (UC1) matches the set value of timer register TREG or when 2n - 1 (n = 6, 7, or 8; specified by T01MOD) counter overflow occurs. Up-counter UC1 is cleared when 2n - 1 counter overflow occurs. For example, when n = 6, 6-bit PWM will be output, while when n = 7, 7-bit PWM will be output.

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

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

(Set value of timer register ≠ 0)

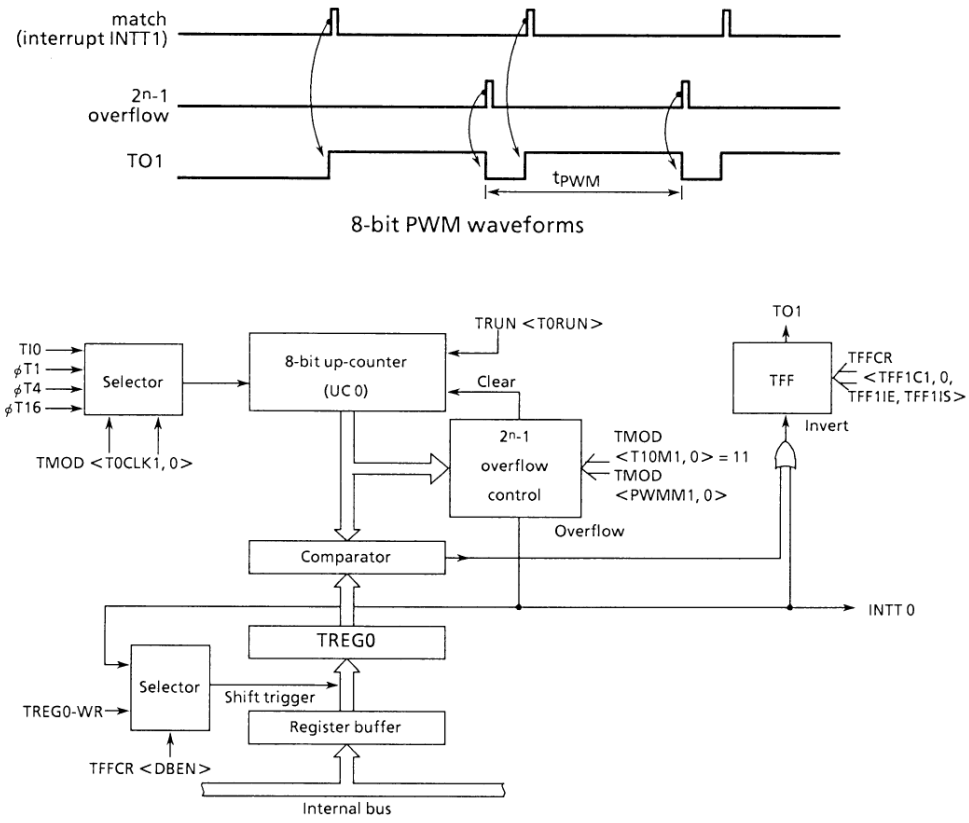
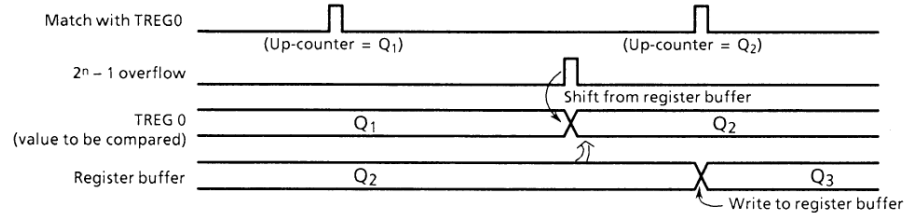


Figure 3.7 (13). Block Diagram of 8-Bit PWM Waveforms

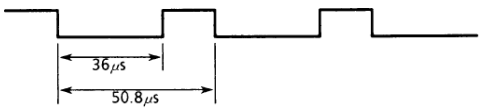
In this mode, the value of register buffer will be shifted in TREG0 if 2^n - 1 overflow is detected when the double buffer of TREG0 is enabled.

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



Operation of Register buffer

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



To realize 50.8μs of PWM cycle by ϕT1 = 0.4μs (@ fc = 20MHz),

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

Consequently, n should be set to 7.

As the period of low level is 36μs, for ϕT1 = 0.4μs, set the following value for TREG0:

$$36\mu s \div 0.4\mu s = 90 = 5AH$$

		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, enables the inversion and double buffer.
TREG0	←	0	1	0	1	1	0	1	0	Write "5AH".
P7CRL	←	–	–	–	–	–	–	1	0	Set P70 as the TO1 pin.
TRUN	←	x	x	1	–	–	–	–	1	Start timer 0 counting.

Note : x; don't care –; no change

Table 3.7 (3) PWM Cycle and the Setting of 2^n - 1 Counter

	PWM Cycle (@ fc = 20 MHz)		
	ϕT1	**ϕT4**	**ϕT16**
2^6 - 1	25.2μsec (39.0kHz)	100μsec (10.0kHz)	4.03msec (2.4kHz)
2^7 - 1	50.8μsec (19.7kHz)	203μsec (4.9kHz)	812msec (1.2kHz)
2^8 - 1	102μsec (9.80kHz)	408μsec (2.4kHz)	1.63msec (0.61kHz)

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

Table 3.7 (4) Timer Mode Setting Registers

Timer Mode (8-bit timer x 2channel)	Mode T01M (T23M)	PWM0 (PWM2)	Upper Input T1CLK (T3CLK)	Lower Input T0CLK (T2CLK)	Invert Select FF1IS (FF3IS)
16-bit timer (Full 16-bit) x 1channel	01	–	–	(External clock, ϕT1, 4, 16)	–
8-bit timer (8-bit x 8-bit mode x 1channel) (Comparator output from the lower timer is input to the upper timer.)	00	–	00	(External clock, ϕT1, 4, 16)	0 : Lower timer 1 : Upper timer
8-bit timer x 2channel	00	–	(ϕT1, T16, T256)	(External clock, ϕT1, 4, 16)	0 : Lower timer 1 : Upper timer
8-bit PPG x 1channel	10	–	–	(External clock, ϕT1, 4, 16)	–
8-bit PWM x 1channel (Lower) 8-bit timer x 1channel (Upper)	11	PWM cycle	(ϕT1, T16, T256)	(External clock, ϕT1, 4, 16)	–

Note: –: don't care

3.8 16-Bit PWM Timer

The TMP96C031F contains one (timer 4) 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 mode
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

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

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

Figure 3.8 (1) shows the block diagram of 16-bit timer/event counter (timer 4).

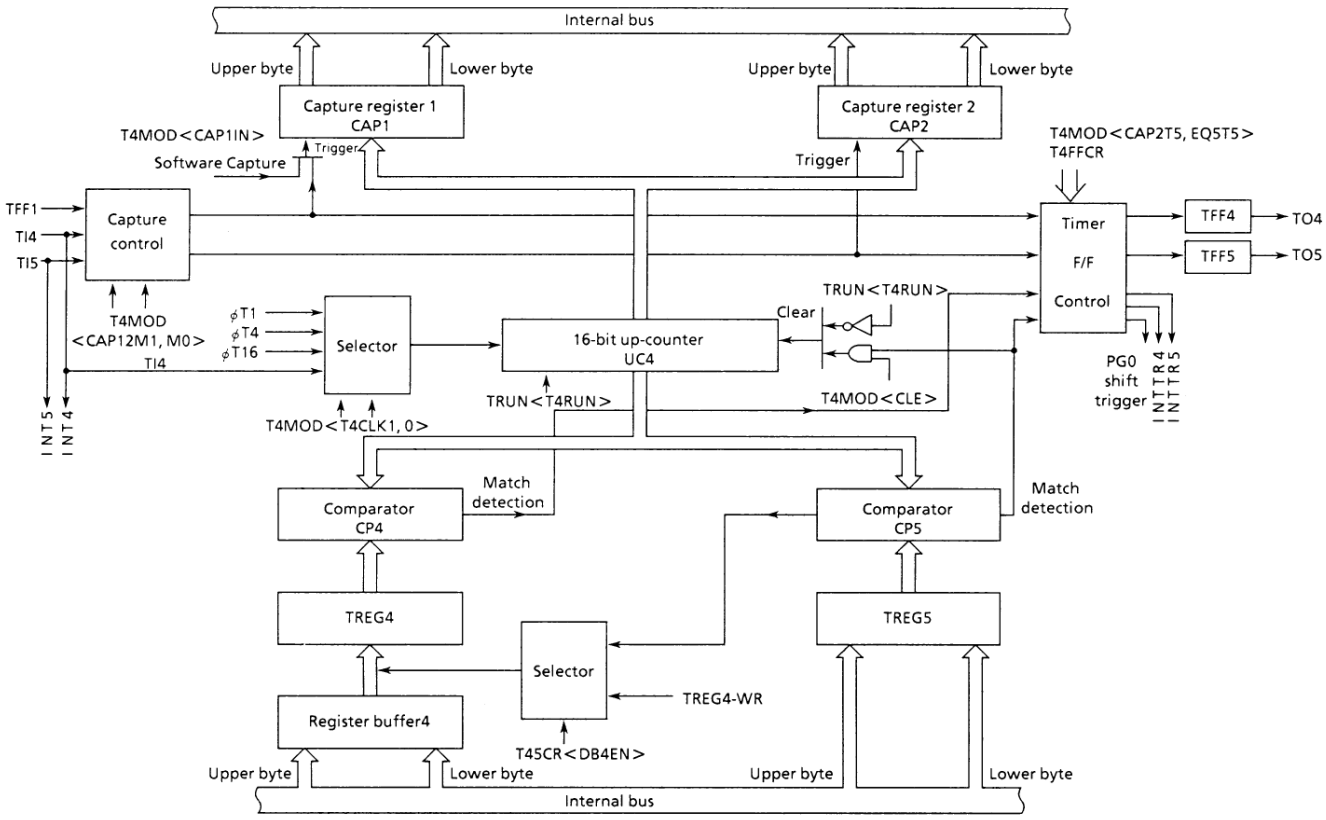


Figure 3.8 (1). Block Diagram of 16-Bit Timer (Timer 4)

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 3.8 (2). 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 3.8 (3). 16-Bit Timer Mode Controller Register (T4MOD) (2/2)

T4FFCR (0039H)

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

Timer flip-flop 4 (TFF4) control

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

Timer flip-flop 4 (TFF4) invert trigger

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

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

Timer flip-flop 5 (TFF5) control

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

Figure 3.8 (4). 16-Bit Timer 4 F/F Control (T4FFCR)

T45CR (003AH)

	7	6	5	4	3	2	1	0
bit Symbol	—				PG1T	PG0T		DB4EN
Read/Write	R/W				R/W			R/W
After reset	0				0	0		0
Function	Fix at "0"				PG1 shift Trigger 0: 8-bit Timer Trigger (Timer 0, 1) 1: 16-bit Timer Trigger (Timer 5)	PG0 shift Trigger 0: 8-bit Timer Trigger (Timer 0, 1) 1: 16-bit Timer Trigger (Timer 4)		Double buffer of TREG4 0: Disable 1: Enable

Double buffer countrol

0	Disable
1	Enable

DB4EN : Double buffer of TREG4

Figure 3.8 (5). 16-Bit Timer (Timer 4) Control Register (T45CR)

TRUN (0020H)

16-bit timer 8-bit timer

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

Operation of 16-bit timer (timer4)

0	Stop and clear
1	Count

Operation of prescaler

0	STOP and clear
1	Count

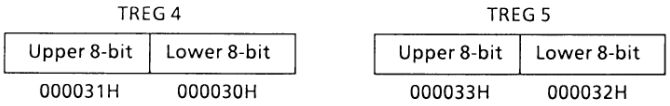
Figure 3.8 (6). Timer Operation Control Register (TRUN)

① Up-counter

UC4 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD <T4CLK1,0> register.
As the input clock, one of the internal clocks ϕT1(8/fc), ϕT4 (32/fc), and ϕT16 (128/fc) from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P72/INT4 pin) can be selected.
When reset, it will be initialized to <T4CLK1,0> = 00 to select TI4 input pin mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN> .
When clearing is enabled, up-counter UC4 will be cleared to zero each time it coincides matches the timer register TREG5. The "clear enable/disable" is set by T4MOD <CLE>.
If clearing is disabled, the counter operates as a free-running counter.

② Timer registers

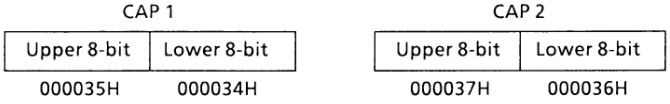
These two 16-bit registers are used to set the interval time. When the value of up-counter UC4 matches the set value of this timer register, the comparator match detect signal will be active.
Setting data for timer register (TREG4 and TREG5) is executed using 2-byte data transfer instruction or using 1-byte data transfer instruction twice for lower 8-bit and upper 1-bit in order.



TREG4 timer register is of double buffer structure, which is paired with register buffer. The timer control register T45CR <DB4EN> controls whether the double buffer should be enabled or disabled. : disabled when <DB4EN> = 0, while enabled when <DB4EN> = 1.
When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up-counter (UC4) and timer register TREG5.
When reset, it will be initialized to <DB4EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN> = 1, and then write the following data in the register buffer.
TREG4 and register buffer are allocated to the same memory addresses 000030H/000031H. When <DB4EN> = 0, same value will be written into only the register buffer.

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-bit followed by the upper 8-bit.



④ Capture Input Control

This circuit controls the timing to latch the value of up-counter UC4 into (CAP1, CAP2). The latch timing of capture register is controlled by register T4MOD <CAP12M 1, 0>/T5MOD <CAP34M1,0>.

- When T4MOD <CAP12M 1, 0> = 00
  Capture function is disabled. Disable is the default on reset.

- When T4MOD <CAP12M1, 0> = 01
  Data is loaded to CAP1 at the rise edge of TI4 pin (also used P80/INT4) input, while data is loaded to CAP2 at the rise edge of TI5 pin (also used as P81/INT5) and input. (Time difference measurement)

- When T4MOD <CAP12M1, 0> = 10
  Data is loaded to CAP1 at the rise edge of TI4 pin input, while to CAP2 at the fall edge. Only in this setting, interrupt INT4 occurs at fall edge. (Pulse width measurement)

- When T4MOD <CAP12M1, 0> = 11
  Data is loaded to CAP1 at the rise edge of timer flip-flop TFF1, while to CAP2 at the fall edge.

  Besides, the value of up-counter can be loaded to capture registers by software. Whenever “0” is written in T4MOD <CAP1IN> the current value of up-counter will be loaded to capture register CAP1. It is necessary to keep the prescaler in RUN mode (TRUN <PRRUN> to be “1”).

⑤ Comparator

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

⑥ Timer flip-flop (TFF4)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4>. TFF4 will be inverted when “00” is written in T4FFCR <TFF4C1,0>. Also it is set to “1” when “10” is written, and cleared to “0” when “10” is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P70).

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>/T6FFCR <TFF6C1,0>. 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).
TO5 (also used as P30) is multiplexed using the HWR pin; setting must be done using the port 3 control register, P3CRL.

Note: TO5 (also used as P30) is multiplexed with HWR; setting must be done using the P3SR.

(1) 16-Bit Timer Mode

Generating interrupts at fixed intervals

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

		7	6	5	4	3	2	1	0	
TRUN	←	x	x	–	0	–	–	–	–	Stop timer 4.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4. Disable INTTR4.
T4FFCR	←	1	1	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	*	*	Select internal clock for input and disable the capture function.
							(** = 01, 10, 11)			
TREG5	←	0	0	1	0	1	0	0	0	Set the interval time (16-bit).
		*	*	*	*	*	*	*	*	
TRUN	←	1	x	1	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 be selecting the external clock (TI4 pin input) as the input clock. To read the value of the counter, first perform the "software capture" once and read the captured value.
The counter counts at the rise edge of TI4 pin input.
TI4 can also be used as P72/INT4.

		7	6	5	4	3	2	1	0	
TRUN	←	x	x	–	0	–	–	–	–	Stop timer 4.
P7CR	←	–	–	0	0	–	–	–	–	Set P72 to input mode.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4, while disable 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-bit).
TRUN	←	x	x	1	1	–	–	–	–	Start timer 4.

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

(3) 16-Bit Programmable Pulse Generation (PPG) Mode

The PPG mode is obtained by inversion of the timer flip-flop TFF4 that is to be enabled by the match of the up-counter UC4 with the timer register TREG4 or 5 and to be output to TO4 (also used as P70). 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	x	–	0	–	–	–	–	Stop timer 4.
TREG4	←	*	*	*	*	*	*	*	*	Set the duty. (16-bit).
TREG5	←	*	*	*	*	*	*	*	*	Set the cycle. (16-bit).
T45CR	←	*	*	*	*	*	*	*	*	Double Buffer of TREG4 enable (Change the duty and cycle at the interrupt INTTR5).
T4FFCR	←	1	1	0	0	0	0	1	1	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	0	0	Select the internal clock for the input, and disable the capture function.
							(** = 01, 10, 11)			
P7CR	←	–	–	–	–	–	–	1	1	Assign P70 as TO4.
TRUN	←	x	x	1	1	–	–	–	–	Start timer 4.

Note : x; don't care –; no change

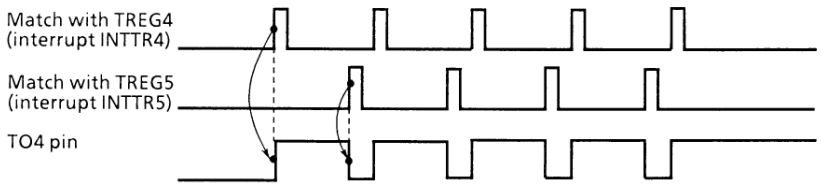


Figure 3.8 (7). Programmable Pulse Generation (PPG) Output Waveforms

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

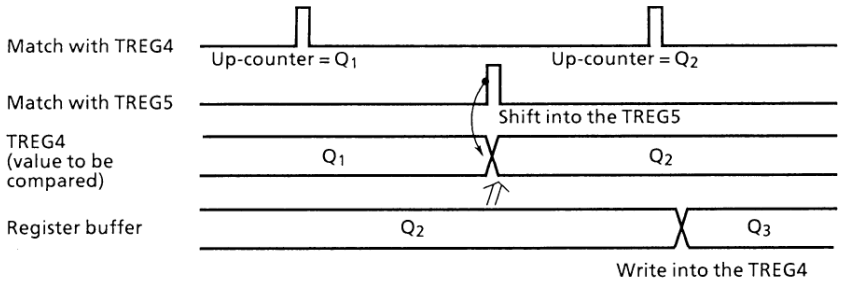


Figure 3.8 (8). Operation of Register Buffer

Shows the block diagram of this mode.

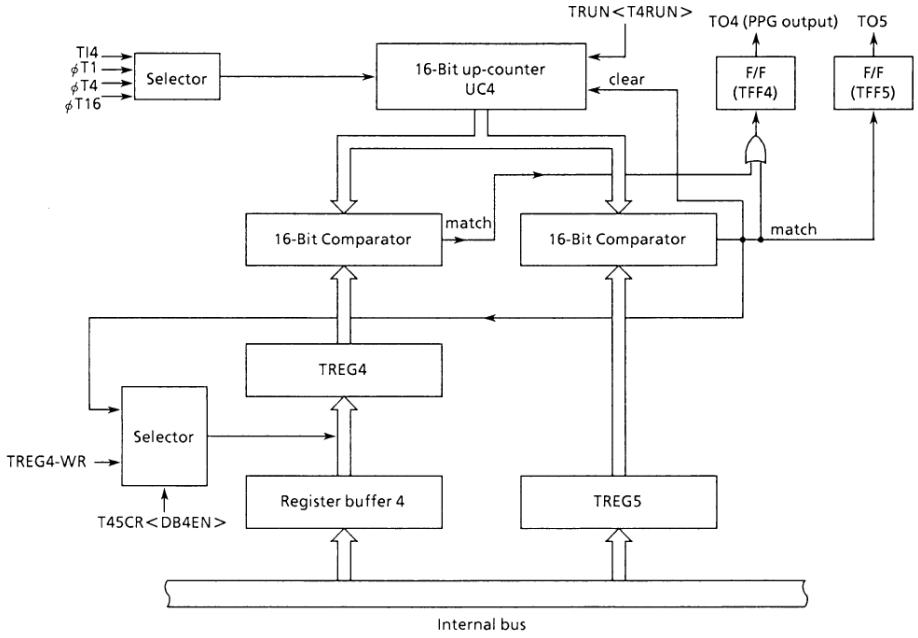


Figure 3.8 (9). Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

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

① One-shot pulse output from the external trigger pulse
② Frequency measurement
③ Pulse width measurement
④ Time difference measurement

① One-Shot Pulse Output from the 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 T4MOD <CAP12M1, 0> = 01.

When the interrupt INT4 is generated at the rise edge of the TI4 pin, set the CAP1 value (c) plus a delay time (d) to TREG5 (= c + d), and set the above set value (c + d) plus a one-shot pulse width (p) to TREG5 (= c + d + p). When interrupt INT4 occurs the T4FFCR <EQ5T4, EQ4T4> register sgould 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.

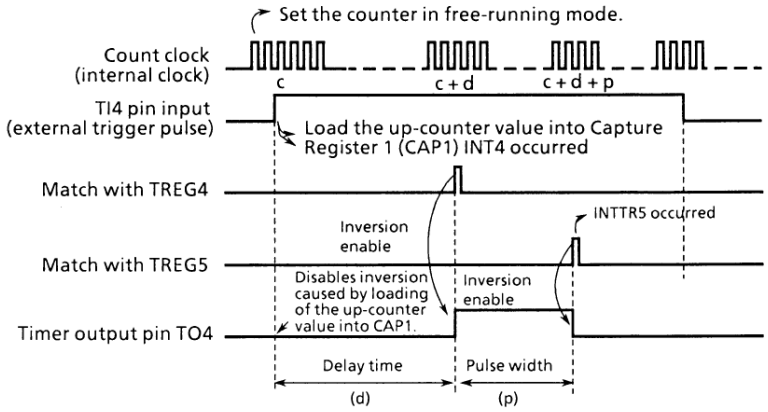


Figure 3.8 (10). One-Shot Pulse Output (with Delay)

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

When delay 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

Main setting → Keep

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

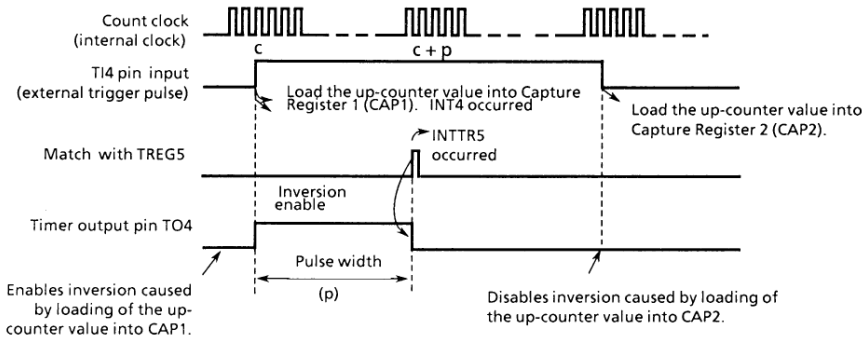


Figure 3.8 (11). One-Shot Pulse Output (without Delay)

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.

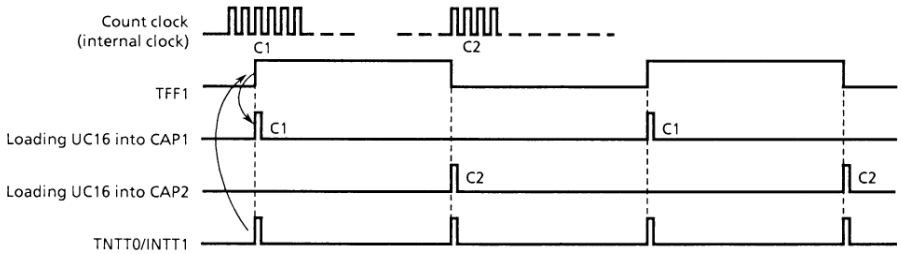


Figure 3.8 (12). Frequency Measurement

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

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

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

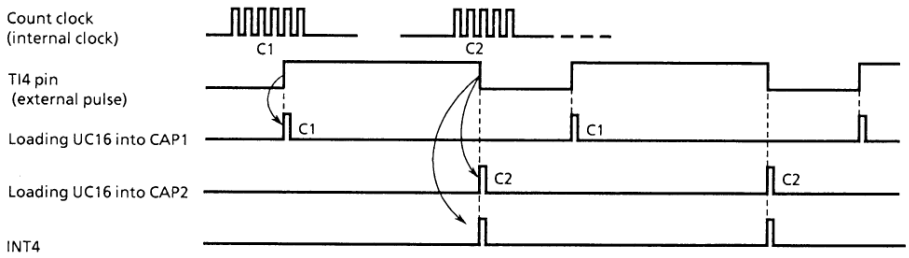


Figure 3.8 (13). Pulse Width Measurement

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.

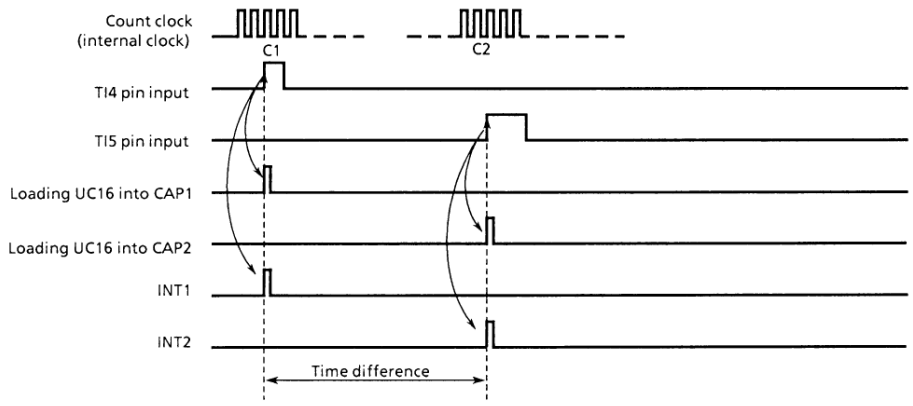


Figure 3.8 (14). Time Difference Measurement

(5) Different Phased Pulses Output Mode

In this output 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 only be used by 16-bit timer 4.

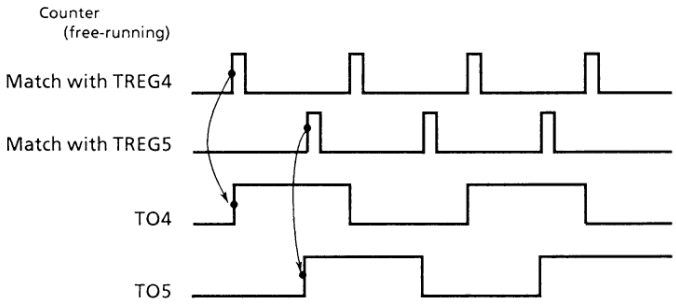


Figure 3.8 (15). Phase Output

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

	16MHz	20MHz
ϕT1	32.768ms	26.214ms
ϕT4	131.072ms	104.856ms
ϕT16	524.288ms	419.424ms

3.9 Stepping Motor Control/Pattern Generation Port

The TMP96C031F 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 P6.

Channel 0 (PG0) is synchronous with 8-bit timer 0 or timer 1, 16-bit timer 4, channel 1 (PG1) is synchronous with 8-bit timer2 or timer3, 16-bit timer4, to update the output.

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

Channel 0 (PG0) and channel 1 (PG1) operate independently.

Except in the following case, both channels operate the same. Thus, channel 0 (PG0) is explained here.

Difference between PG0 and PG1

	PG0	PG1
Timer trigger signal	8-bit timer0, 1 16-bit timer4	8-bit timer2, 3 16-bit timer4

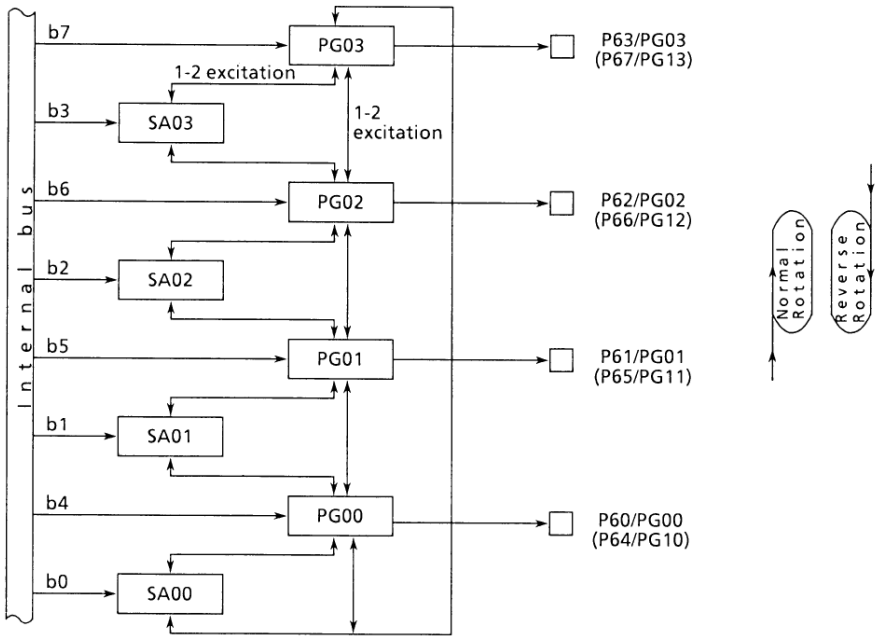


Figure 3.9 (1). Pattern Generator/Stepping Motor Control Block Diagram

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: 8-bit write 1: 4-bit 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: 8-bit write 1: 4-bit 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 3.9 (2a). Pattern Generation Control Register (PG01CR)

PG01CR (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R / W				R / W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8-bit write 1: 4-bit 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: 8-bit write 1: 4-bit 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 3.9 (2b). Pattern Generation Control Register (PG01CR)

PG0REG (004CH)

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

Prohibit Read modify write

Figure 3.9 (3). Pattern Generation 0 Register (PG0REG)

PG1REG (004DH)

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

Prohibit Read modify write

Figure 3.9 (4). Pattern Generation 1 Register (PG1REG)

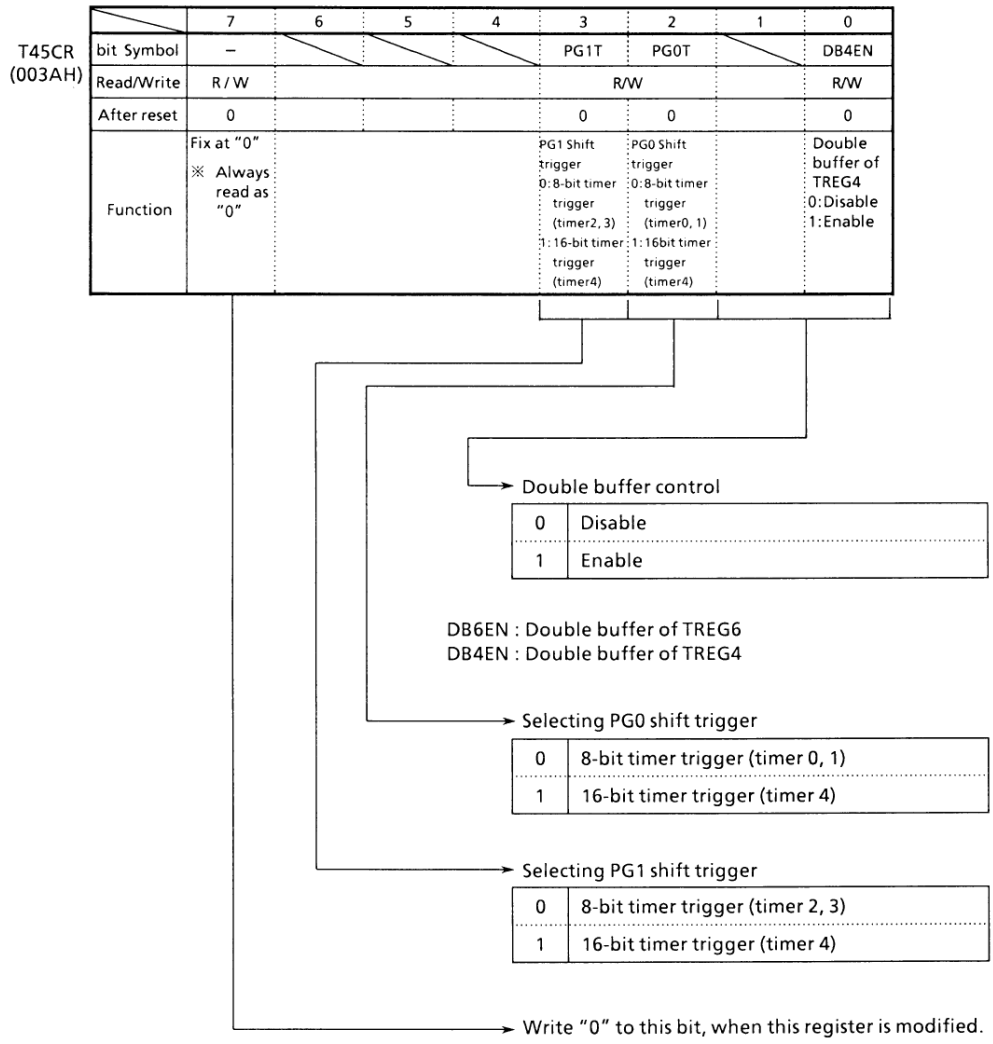


Figure 3.9 (5). 16-bit Timer Trigger Control Register (T45CR)

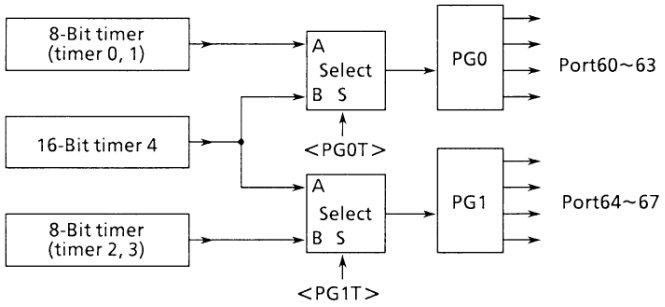


Figure 3.9 (6). Connection of Timer and Pattern Generator

(1) Pattern Generation Mode

PG functions as a pattern generation according to the setting of PG01CR <PAT1>/<PAT0>. In this mode, writing from CPU is executed only on the shifter alternate register. Writing a new data should be done during the interrupt operation of the timer for shift trigger, and a pattern can be output synchronous with the timer.

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

The output of this pattern generator is output to port 6; since port and functions can be switched on a bit basis using port function control register P6CRL/P6CRH, any port pin can be assigned to pattern generator output.

Figure 3.9 (7) shows the block diagram of this mode.

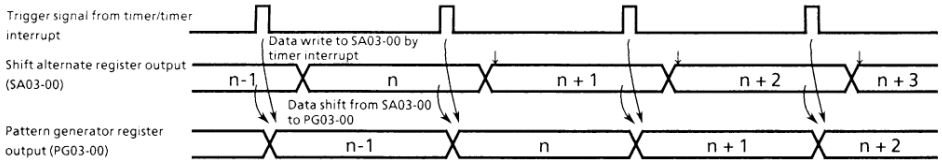


Figure 3.9 (7). Pattern Generation Mode Timing Example

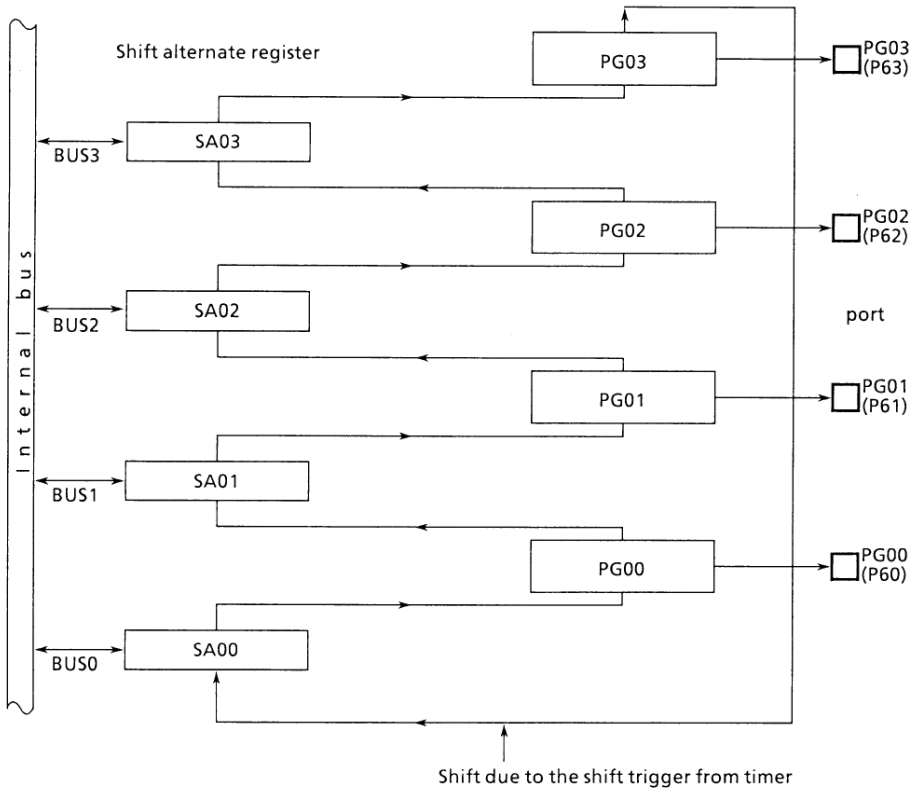


Figure 3.9 (7). Pattern Generation Mode Block Diagram (PG0)

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

(2) Stepping Motor Control Mode

① 4-phase 1-Step/2-Step Excitation

Figure 3.9 (8) and Figure 3.9 (9) show the output waveforms of 4-phase 1 excitation and 4-phase 2 excitation, respectively when channel 0 (PG0) is selected.

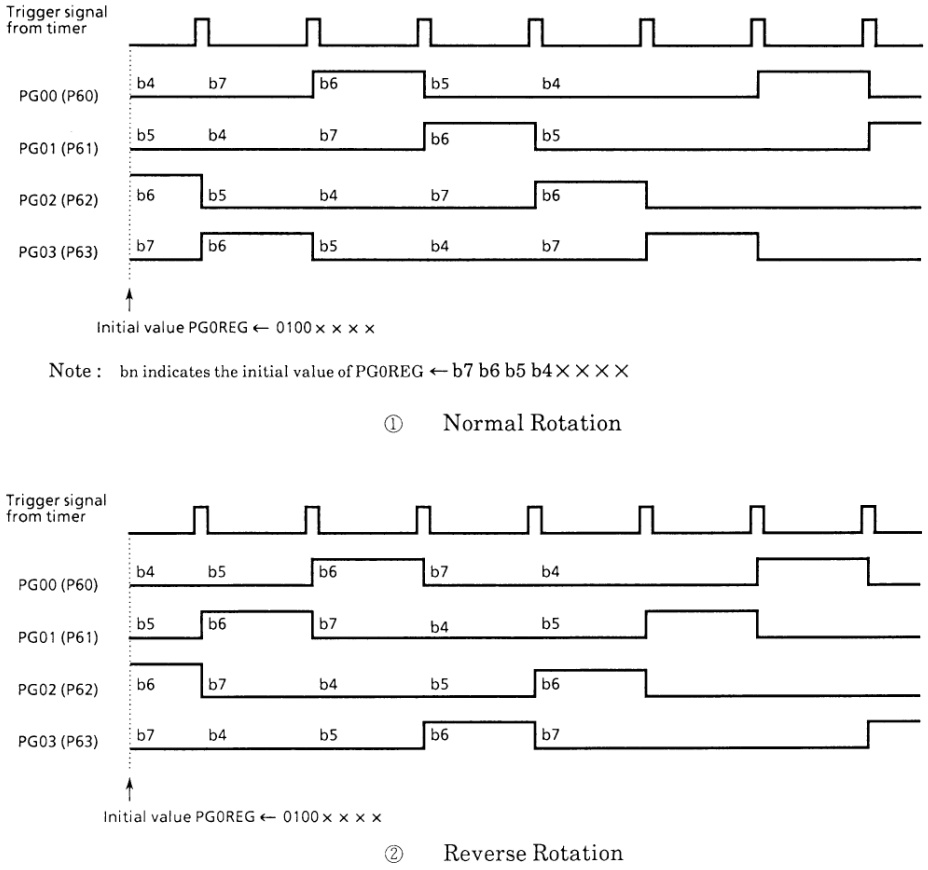


Figure 3.9 (8). Output Waveforms of 4-Phase 1-Step Excitation (Normal Rotation and Reverse Rotation)

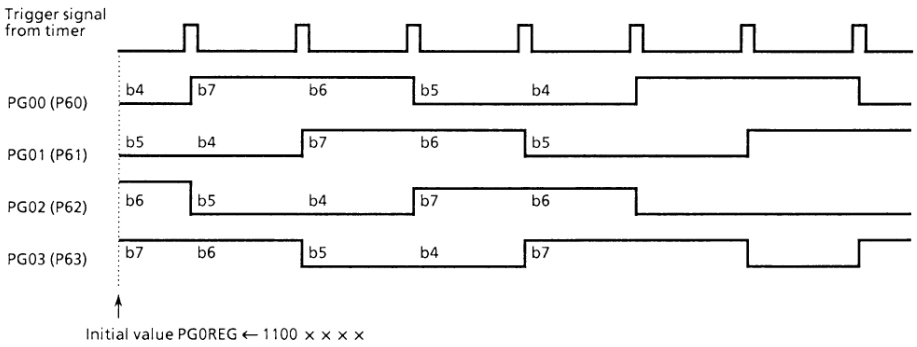


Figure 3.9 (9). Output Waveforms of 4-Phase 2-Step Excitation (Normal Rotation)

The operation when channel 0 is selected is explained below.

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

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

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

Figure 3.9 (10) shows the block diagram.

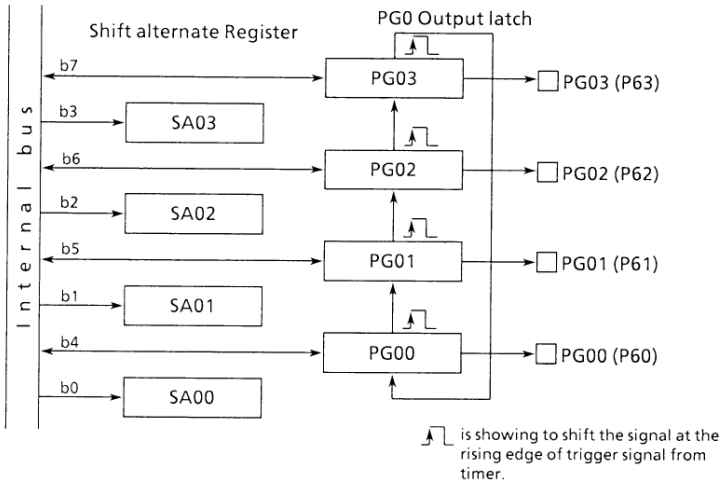
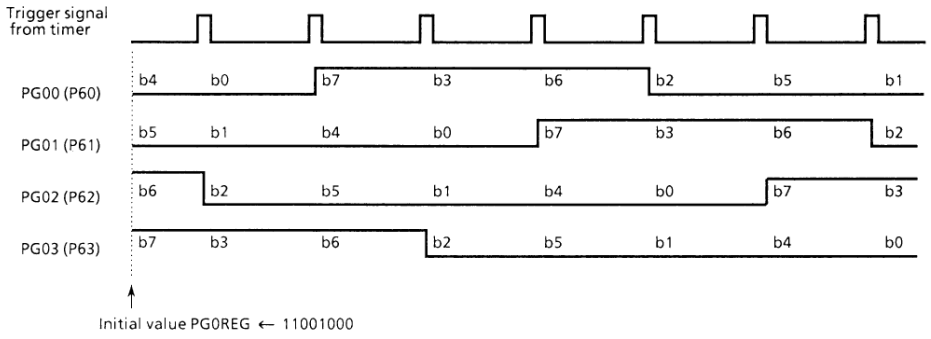


Figure 3.9 (10). Block Diagram of 4-Phase 1-Step Excitation/2-Step Excitation (Normal Rotation)

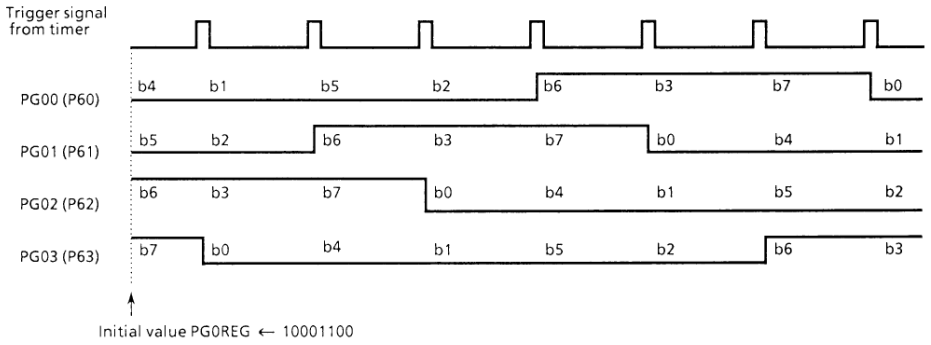
② 4-Phase 1-2 Step Excitation

Figure 3.9 (11) shows the output waveforms of 4-phase 1 -2 step 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 3.9 (11). Output Waveforms of 4-Phase 1-2 Step Excitation (Normal Rotation and Reverse Rotation)

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

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

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

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

The operation will be explained below for channel 0.

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

Figure 3.9 (12) shows the block diagram.

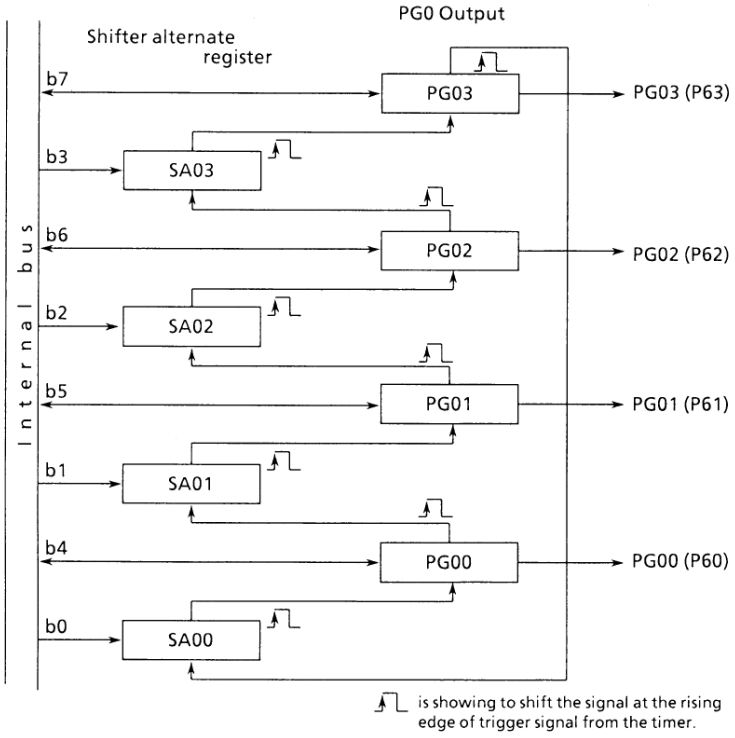


Figure 3.9 (12). Block Diagram of 4-Phase 1-2 Step Excitation (Normal Rotation)

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

		7	6	5	4	3	2	1	0	
TRUN	←	–	x	–	–	–	–	–	0	Stop timer 0, and clears it to zero.
TMOD	←	0	0	x	x	–	–	0	1	Set 8-bit timer mode and selects ϕT1 as the input clock of timer 0.
TFFCR	←	x	x	x	0	1	0	1	0	Clear TFF1 to zero and enables the inversion trigger by timer 0.
TREG0	←	*	*	*	*	*	*	*	*	Set the cycle in timer register.
P6CRL	←	1	0	1	0	1	0	1	0	Set P60 ~ P63 bits to PG output.
PG01CR	←	–	–	–	–	0	0	1	1	Select PG0 4-phase 1 - 2 step excitation mode and normal rotation.
PG0REG	←	1	1	0	0	1	0	0	0	Set an initial value.
TRUN	←	1	x	–	–	–	–	–	1	Start timer 0.

Note: x; don't care –; no change

(3) Trigger Signal From Timer

The trigger signal from the timer which is used by PG is not equal to the trigger signal of timer flip-flop (TFF1, TFF4, TFF5, and TFF6) and differs as shown in Table 3.9 (1) depending on the operation mode of the timer.

Table 3.9 (1) Select of Trigger Signal

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

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

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

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

(4) Application of PG and Timer Output

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

To drive a stepping motor, in addition to the value of each phase (PG output), synchronizing signal is often required at the timing when excitation is changed over. In this application, port 6 is used as a stepping motor control port to output a synchronizing signal to the TO1 pin (shared by P70).

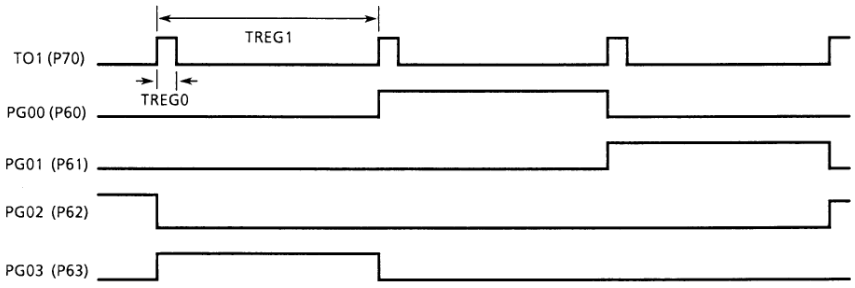


Figure 3.9 (13). Output Waveforms of 4-Phase 1-Step Excitation

Setting example:

		7	6	5	4	3	2	1	0	
TRUN	←	–	x	–	–	–	–	0	0	Stop timer 0, and clears it to zero.
TMOD	←	1	0	x	x	x	x	0	1	Set timer 0 and timer 1 in PPG output mode and selects ϕT1 as the input clock.
TFFCR	←	x	x	x	0	0	1	1	x	Enable TFF1 inversion and sets TFF1 to “1”.
TREG0	←	*	*	*	*	*	*	*	*	Set the duty of TO1 to TREG0.
TREG1	←	*	*	*	*	*	*	*	*	Set the cycle of TO1 to TREG1.
P7CR	←	–	–	–	–	–	–	1	0	Assign P70 as TO1.
P6CRL	←	1	0	1	0	1	0	1	1	Assign P60 ~ 63 as PG0.
PG01CR	←	–	–	–	–	0	0	0	1	Set PG0 in 4-phase 1-step excitation mode.
PG0REG	←	*	*	*	*	*	*	*	*	Set an initial value.
TRUN	←	1	x	–	–	–	–	1	1	Start timer 0 and timer 1.

Note: x; don’t care –; no change

3.10 Serial Channel

The TMP96C031F contains 2 serial I/O channels for full duplex asynchronous transmission (UART) as well as for I/O extension.

The serial channel has the following operation modes.

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

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

Figure 3.10 (1) shows the data format (for one frame) in each mode.

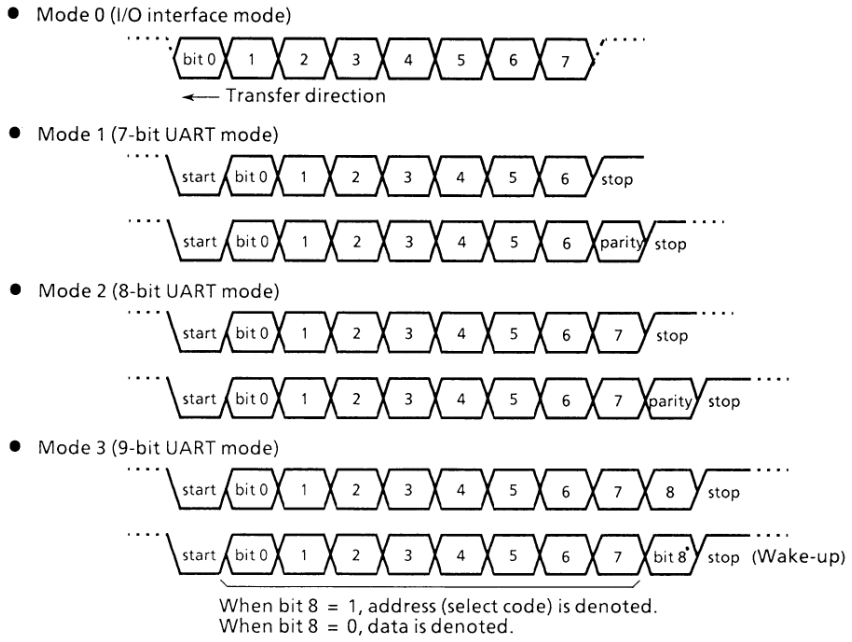


Figure 3.10 (1). Data Formats

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

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

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

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

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

3.10.1 Control Registers

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

SC0MOD (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data Bit 8	Hand shake fanction control 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: Can not be used 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 Trigger 01: baud rate generator 10: Internal clock Φ1 11: don't care	

Serial transmission clock (UART)

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

Serial transmission mode

00	Can not be used	
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

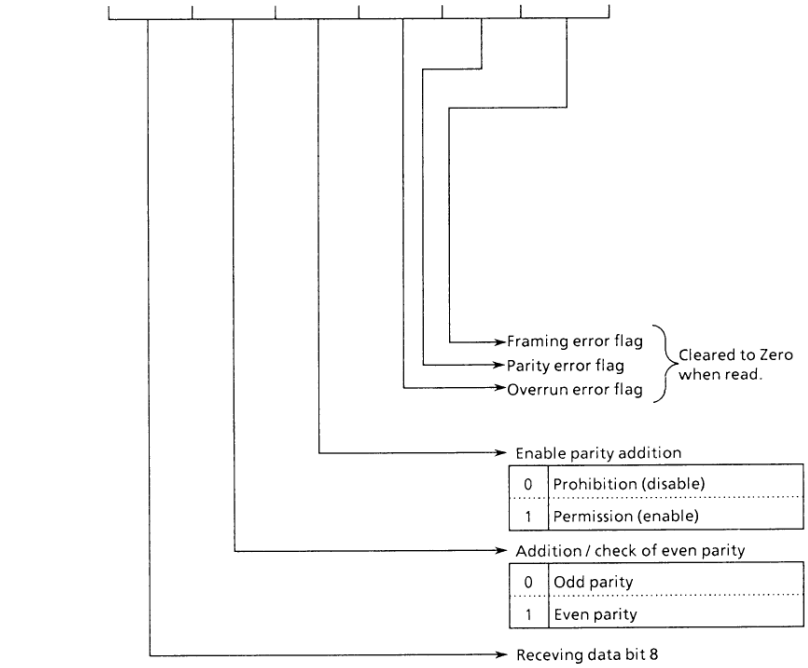
Transmission data bit 8

Note : There is SC1MOD (56H) in Channel1

Figure 3.10 (2). Serial Mode Control Register (Channel 0, SC0MOD)

SC0CR (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	—	—
Read/Write	R	R/W		R (cleared to Zero when read)			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	Fix at "0"	Fix at "0"



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 3.10 (3). Serial Control Register (Channel, SC0CR)

BR0CR (0053H)

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

Setting of the divided frequency of baud rate generator

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

Selecting the input clock of baud rate generator

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

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

Figure 3.10 (4). Serial Channel Control (Channel 0, BR0CR)

SC0BUF (50H)

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

Figure 3.10 (5). Serial Transmission/Receiving Buffer Registers (Channel 0, SC0BUF)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	—	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfered data Bit 8	Fix at "0"	Receiving Function 0: Receive disable 1: Receive Enable	Wake up Function 0: disable 1: enable	Serial Transmission mode 00: I/O interface mode 01: 7-Bit UART 10: 8-Bit UART 11: 9-Bit UART		Serial Transmission clock (UART) 00: TO0 Trigger 01: Baud rate generator 10: Internal clock Φ1 11: don't care	

Serial transmission clock (For UART)

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

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

Serial transmission mode

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

Wake up Function

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

Receiving control

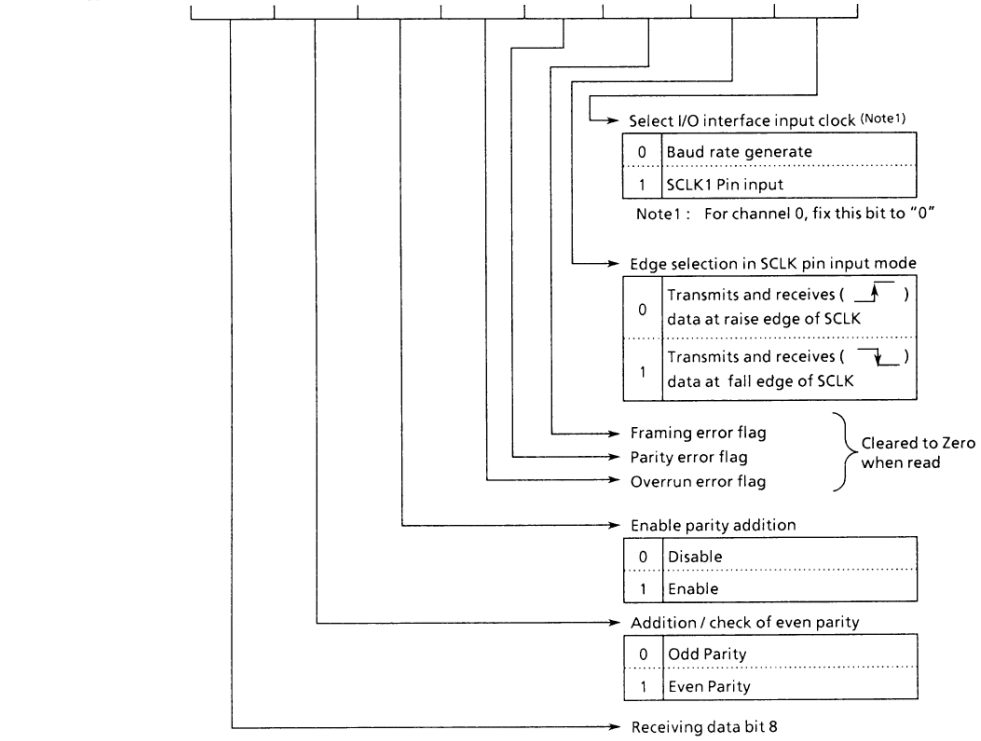
0	Receive disable
1	Receive enable

Transmission data bit 8

Figure 3.10 (6). Serial Mode Control Register (Channel 1, SC1MOD)

SC1CR (0055H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (clear to Zero when read)			R/W	
After reset		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



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

Figure 3.10 (7). Serial Control Register (Channel 1, SC1CR)

BR1CR (0057H)

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

Setting of the divided frequency of baud rate generator

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

Selecting the input clock of baud rate generator

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

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

Figure 3.10 (8). Baud Rate Generator Control Register (Channel 0, BR0CR)

SC1BUF (0054H)

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

Figure 3.10 (9). Serial Transmission/Receiving Buffer Registers (Channel 1, SC1BUF)

Port 6 Control Register L

P6CRL (0014H)

	7	6	5	4	3	2	1	0
bit Symbol	P63C1	P63C0	P62C1	P62C0	P61C1	P61C0	P60C1	P60C0
Read/Write	W		W		W		W	
After reset	0	0	0	0	0	0	0	0
Function	00: PORT input 01: PORT output 10: PG03 11: —		00: PORT input 01: PORT output 10: PG02 11: —		00: PORT input 01: PORT output 10: PG01 11: —		00: PORT input 01: PORT output 10: PG00 11: TXD0	

Prohibit Read modify write

Setting P60 as TxD0 output

00	Port input
01	Port output
10	PG00 output
11	TXD0 (channel0) output

Port 7 Control Register H

P7CRH (0017H)

	7	6	5	4	3	2	1	0
bit Symbol			P76C1	P76C0	P75C1	P75C0	P74C1	P74C0
Read/Write			W		W		W	
After reset			0	0	0	0	0	0
Function			00: PORT input 01: PORT output 10: SCLK1 11: —		00: PORT input 01: PORT output 10: — 11: —		00: PORT input 01: PORT output 10: TXD1 11: —	

Prohibit Read modify write

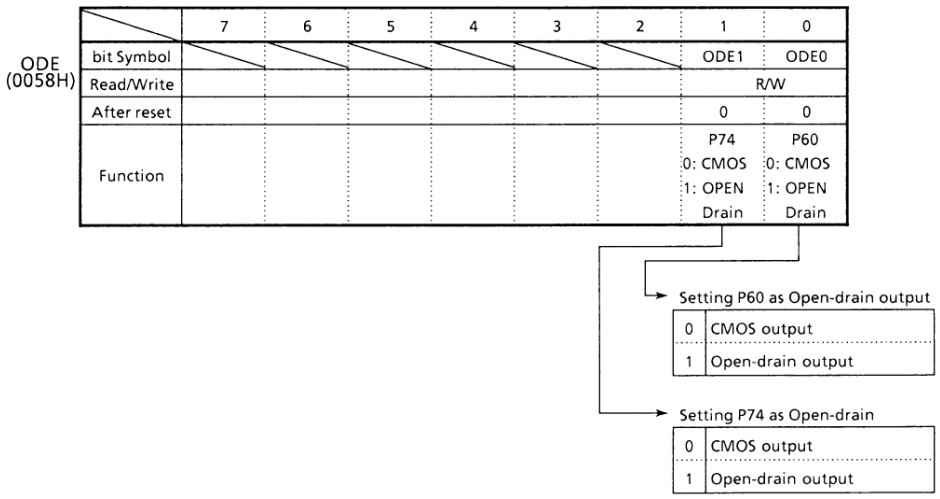
Setting P74 Txd1 output

00	Port input
01	Port output
10	TXD1 (channel1) output

Setting P76 as SCLK1 output

00	Port input
01	Port output
10	SCLK1(channel1)output

Figure 3.10 (10). Port 6, 7 Control Registers



Port 3.10 (11). Serial Open Drain Enable Register (ODE)

3.10.2 Configuration

Figure 3.10 (12) shows the block diagram of the serial channel 0.

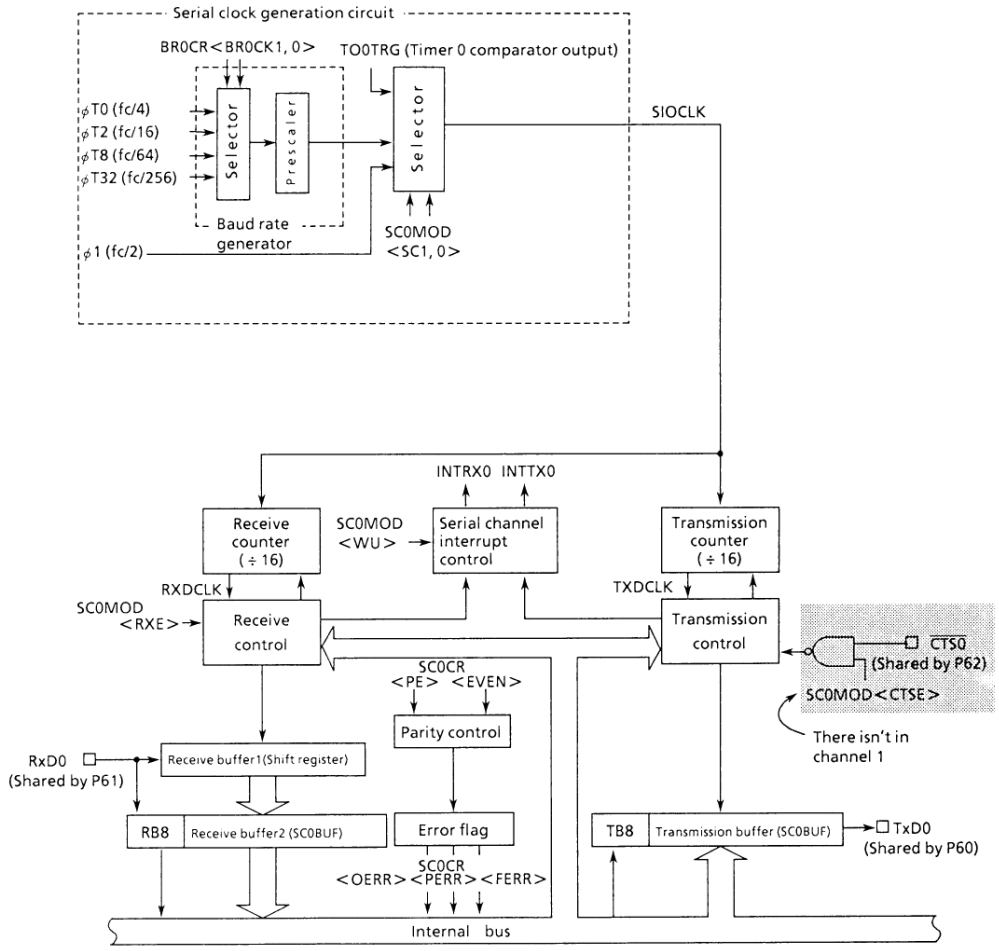


Figure 3.10 (12). Block Diagram of the Serial Channel 0

Figure 3.10 (13) shows the block diagram of the serial channel 1.

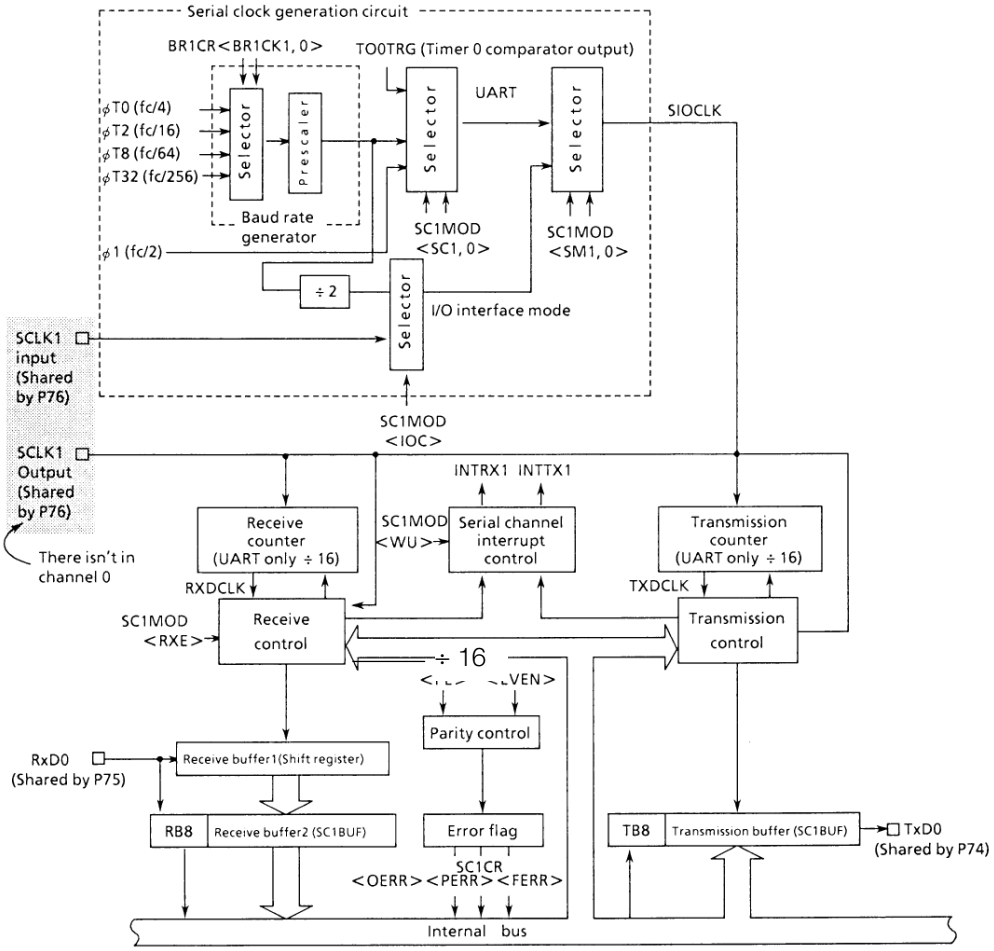


Figure 3.10 (13). Block Diagram of the Serial Channel 1

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 (fc/4), ϕT2 (fc/16), ϕT8 (fc/64), or ϕT32 (fc/256) is generated by the 9-bit prescaler which is shared by the timers.

One of these input clocks is selected by the baud rate generator control register BR0CR/BR1CR <BR0CK1, 0/BR1CK1, 0>.

The baud rate generator includes a 4-bit frequency divider, which divides frequency by 2 to 16 values to determine the transfer rate.

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

- UART mode

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

- I/O interface mode

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

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

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

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

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

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

Table 3.10 (1) shows an example of the transfer rate in UART mode.

Also with 8-bit timer 0, the serial channel can get a transfer rate. Table 3.10 (2) shows an example of baud rate using timer 0.

Table 3.10 (1) Selection of Transfer Rate (1) (When Baud Rate Generator is Used)

Unit (Kbps)

fc [MHz]	Input Clock / Frequency Divisor	ϕT0 (fc/4)	ϕT2 (fc/16)	ϕT8 (fc/64)	ϕT32 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.745600	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300

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

Table 3.10 (2) Selection of Transfer Rate (1) (When Timer 0 (Input Clock ϕT1) is Used)

Unit (Kbps)

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

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

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

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

Input clock of timer 0

ϕT1 = $^{fc}/_{8}$

ϕT4 = $^{fc}/_{32}$

ϕT16 = $^{fc}/_{128}$

Note: Timer 0 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.

- I/O interface mode (channel 1 only)

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

- Asynchronous Communication (UART) mode

According to the setting of SC0CR and SC1CR <SC1, 0>, the above baud rate generator clock, internal clock ϕ1 (500Kbps @ fc = 16 MHz), or the match detect signal from timer 0 will be selected to generate the basic clock SIOCLK.

③ Receiving Counter

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

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

For example, if the sampled data bit is “1", “0” and “1” at 7th, 8th and 9th clock respectively, the received

data is evaluated as “1”. The sampled data “0", “0” and “1” is evaluated that the received data is “0”.

④ Receiving Control

- I/O interface mode (channel 1 only)

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

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

- Asynchronous Communication (UART) mode

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

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

⑤ Receiving Buffer

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

Received data is stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data are stored in the receiving buffer 1, the stored data is transferred to another receiving buffer 2 (SC0BUF/SC1BUF), generating an interrupt INTRX0/INTRX1. The CPU reads only receiving buffer 2 (SC0BUF/SC1BUF). Even before the CPU reads the receiving buffer 2 (SC0BUF/SC1BUF), the received data can be stored in the receiving buffer 1. However, unless the receiving buffer 2 (SC0BUF/SC1BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR <RB8> SC1CR <RB8> are still preserved.

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.

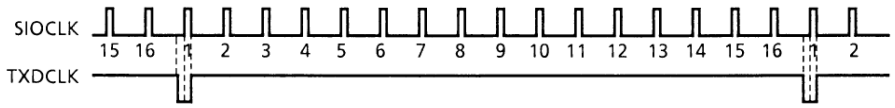


Figure 3.10 (14). Generation of Transmission Clock

⑦ Transmission Controller

- I/O interface mode (channel 1 only)

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

In SCLK input mode with the setting SC1CR <IOC> = “1", the data in the transmission buffer are output bit by bit to TxD1 pin at the rising edge or falling edge of SCLK input according to the setting of SC1CR <SCLKC> register.

- Asynchronous Communication (UART) mode

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

Handshake function

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

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

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

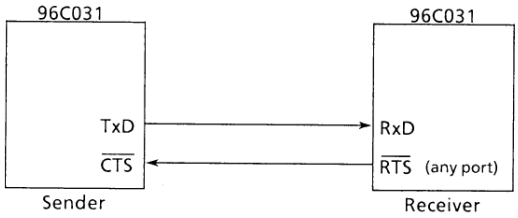
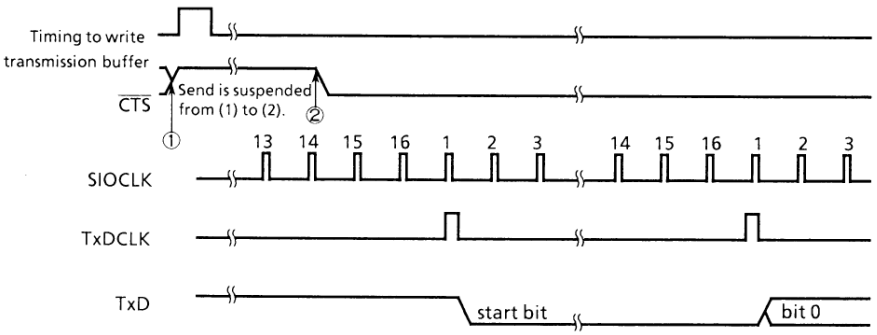


Figure 3.10 (15). Handshake Function



Note 1 : If the $\overline{CTS}$ signal falls 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 3.10 (16). Timing of $\overline{CTS}$ (Clear to Send)

⑧ Transmission Buffer

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

⑨ Parity Control Circuit

When serial channel control register SC0CR <PE>/ SC1CR <PE> is set to "1", it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART mode. With SC0CR <EVEN>/SC1CR <EVEN> register, even (odd) parity can be selected.

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

For receiving, data is shifted in the receiving buffer 1, and parity is added after the data is transferred in the receiving buffer 2 (SC0BUF/SC1BUF), and then compared with SC0BUF <RB7>/SC1BUF <RB7> when in 7-bit UART mode and with SC0MOD <RB8>/SC1MOD <RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs, and SC0CR <PERR>/SC1CR <PERR> flag is set

⑩ Error Flag

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

1. Overrun error <OERR>

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

2. Parity error <PERR>

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

3. Framing error <FERR>

The stop bit of received data is sampled three times around the center. If the majority is "0", a framing error occurs.

⑪ Generating Timing

1) UART mode

Receiving

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

Transmitting

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

2) I/O Interface mode

Transmission interrupt timing	SCLK output mode	Immediately after rise of last SCLK signal. (See figure 3.10 (19) .)
	SCLK input mode	Immediately after rise of last SCLK signal (rising mode), or immediately after fall in falling mode. (See figure 3.10 (20))
Receiving interrupt timing	SCLK output mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after last SCLK. (See figure 3.10 (21))
	SCLK input mode	Timing used to transfer received data to data receive buffer 2 (SC1BUF); that is, immediately after SCLK. (See figure 3.10 (22))

3.10.3 Operational Description

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

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

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

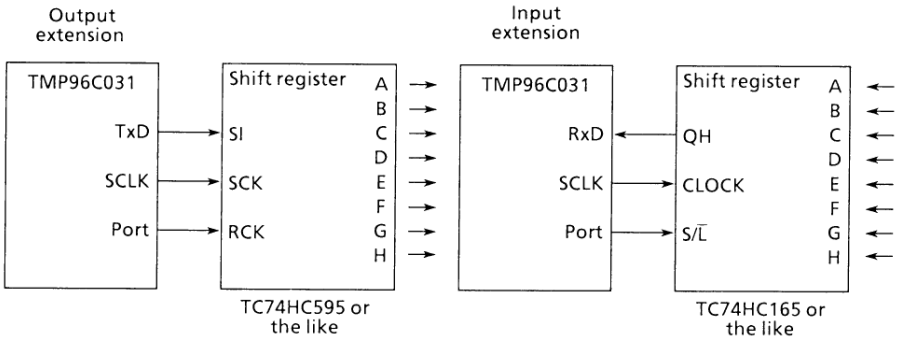


Figure 3.10 (17). Example of SCLK Output Mode Connection

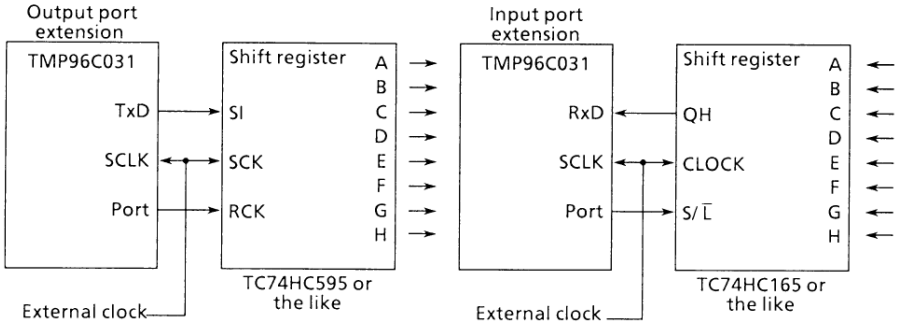


FIgure 3.10 (18). Example of SCLK Input Mode Connection

① Transmission

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

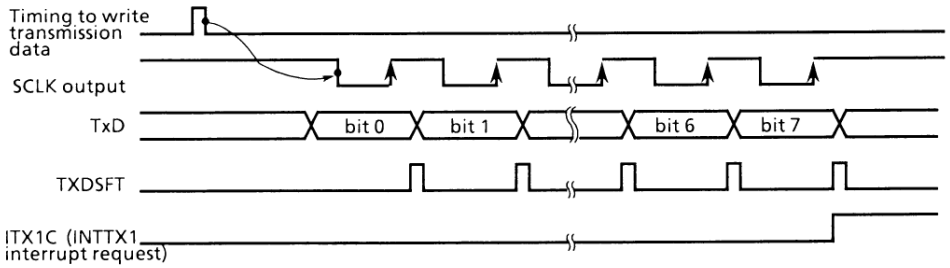


Figure 3.10 (19). Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK output mode, 8-bit data are output from TxD1 pin when SCLK input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES1 <ITXIC> will be set to generate INTTX1 interrupt.

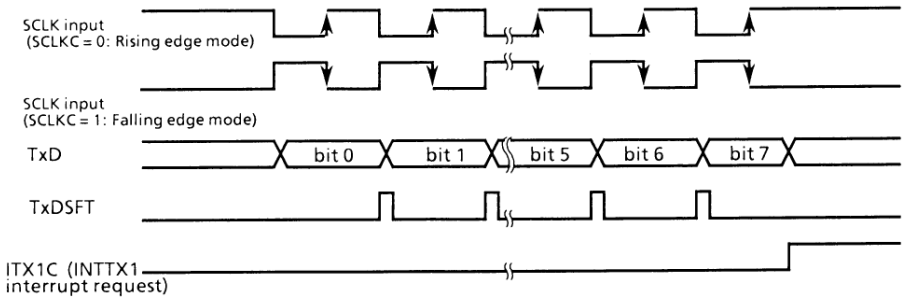


Figure 3.10 (20). Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

In SCLK output mode, synchronous clock is output from SCLK pin and the data is shifted in the receiving buffer 1 whenever the receive interrupt flag 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 (SC1BUF) at the timing shown below, and INTES1 <IRX1C> will be set again to generate INTRX1 interrupt.

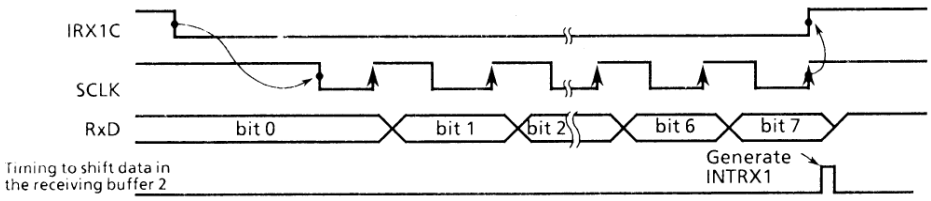


Figure 3.10 (21). Receiving Operation in I/O Interface Mode (SCLK Output Mode)

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

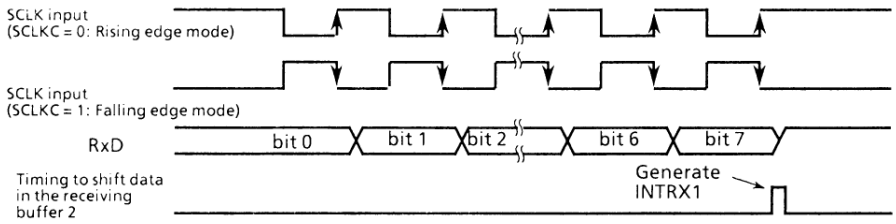


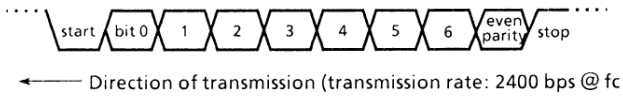
Figure 3.10 (22). Receiving Operation in I/O Interface Mode (SCLK Input Mode)

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

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

The 7-bit mode can be set by setting serial channel mode register SC0MOD <SM1, 0> /SC1MOD <SM1, 0> to “01”.
In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register SC0CR <PE>/SC1CR <PE>, and even parity or odd parity is selected by SC0CR <EVEN>/SC1CR <EVEN> when <PE> is set to “1” (enable).

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



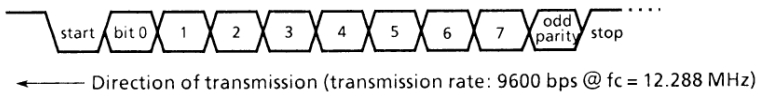
		7	6	5	4	3	2	1	0	
P9CRL	←	–	–	–	–	–	–	1	1	Select P60 as the TxD pin.
SC0MOD	←	x	0	–	x	0	1	0	1	Set 7-bit UART mode.
SC0CR	←	x	1	1	x	x	x	0	0	Add an even parity.
BR0CR	←	0	x	1	0	0	1	0	1	Set transfer rate at 2400 bps.
TRUN	←	x	x	1	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0	←	1	1	0	0	–	–	–	–	Enable INTTX0 interrupt and sets interrupt level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

Note: x; don’t care –; no change

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

The 8-bit UART mode can be specified by setting SC0MOD <SM1, 0> / 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	
P9CRL	←	–	–	–	–	0	0	–	–	Select P61 (RxD) as the input pin.
SC0MOD	←	–	0	1	x	1	0	0	1	Enable receiving in 8-bit UART mode.
SC0CR	←	x	0	1	x	x	x	0	0	Add an odd parity.
BR0CR	←	0	x	0	1	0	1	0	1	Set transfer rate at 9600 bps.
TRUN	←	x	x	1	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0	←	–	–	–	–	1	1	0	0	Enable INTTX0 interrupt and sets interrupt level 4.

Interrupt processing

```
Acc ← SC0CR and 00011100
if Acc ≠ 0 then ERROR      ) Check for error.
Acc ← SC0BUF               Read the received data.
```

Note: x; don't care –; no change

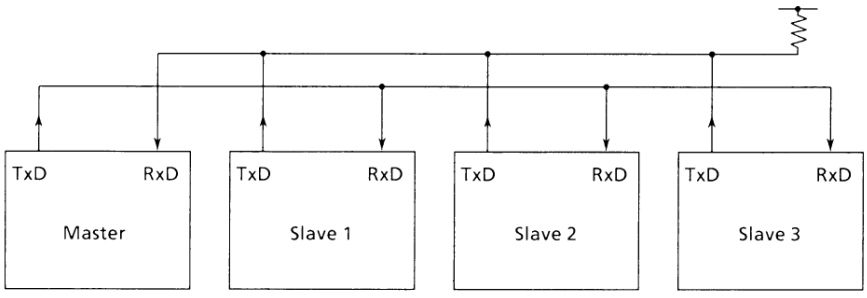
(4) Mode 3 (9-bit UART Mode)

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 SCM0D <TB8>, while in receiving it is stored in SCCR <RB8>. For writing and reading the buffer, the MSB is read or written first, then SC0BUF/SC1BUF.

Wake-up function

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



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

Figure 3.10 (23). Serial Link Using Wake-Up Function

Protocol

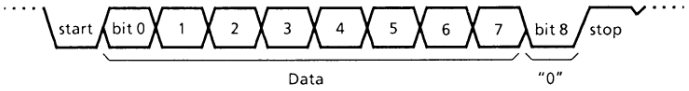
① Select the 9-bit UART mode for master and slave controllers.

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.

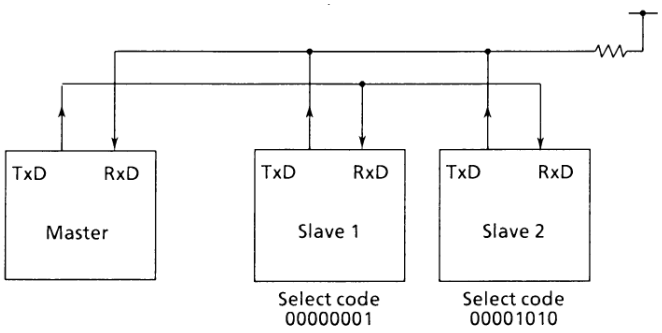
⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU> / SC1MOD <WU> bit is cleared to “0.” The MSB (bit 8) <TB8> is cleared to “0”.



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 (fc/2) as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same way, channel 0 is used for the purposes of explanation.

- Setting the master controller

Main setting

P6CRL	←	–	–	–	–	0	0	1	1	Select P60 as TxD pin and P61 as RxD pin.
INTES0	←	1	1	0	0	1	1	0	1	Enable INTTX0 and sets the interrupt level 4. Enable INTRX0 and sets the interrupt level 5.
SC0MOD	←	1	0	1	0	1	1	1	0	Set ϕ1 (fc/2) 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	–	–	–	–	–	–	Set TB8 to "0".
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

- Setting the slave controller 2

Main setting

P6CRL	←	x	x	–	–	–	–	0	1	Select P61 as RxD pin and P60 as TxD pin.
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 SC0MOD4	←	–	–	–	0	–	–	–	–	Clear <WU> to "0".

3.11 Analog/Digital Converter

The TMP96C031F contains a high-speed analog/digital converter (A/D converter) with 4-channel analog input that features 10-bit successive approximation.

Figure 3.11 (1) shows the block diagram of the A/D converter. The 4-channel analog input pins (AN3 to AN0) are shared by input-only P5 and so can be used as input port.

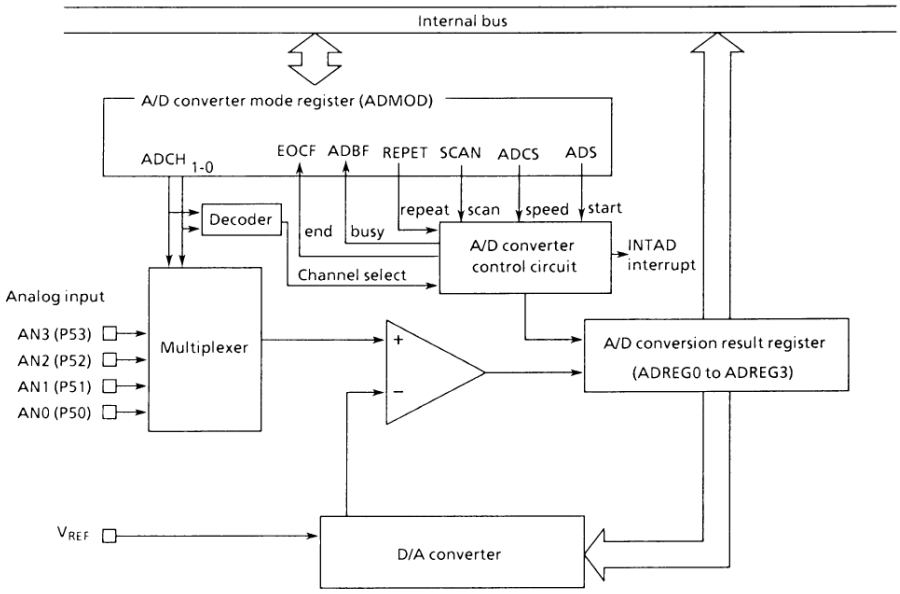


Figure 3.11 (1). Block Diagram of A/D Converter

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

The ladder resistor between VREF- GND cannot be disconnected internally. Therefore, IREF will flow regardless of the mode.

3.11.1 Control Register

ADMOD (005EH)

	7	6	5	4	3	2	1	0
bit Symbol	EOCF	ADBF	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0
Read/Write	R		R/W	R/W	R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	1: END	1: BUSY	1: Repeat Mode Set	1: Scan Mode Set	1: Low speed mode	1: START	Analog Input Channel Select	

Analog input channel select

	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 A/D conversion.

Note) Always "0" when data is read.

A/D conversion speed selection

0	A/D High speed conversion mode: 95 states = 9.5μs (@20MHz)
1	A/D Low speed conversion mode : 190 states = 19μs (@20MHz)

Specification of A/D scan mode

0	Fixed A/D conversion channel mode
1	A/D conversion channel scan mode

Selecting A/D repeat mode

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

A/D conversion busy flag

0	A/D conversion not busy
1	A/D conversion busy

A/D conversion end flag

0	A/D conversion not ended nor started
1	A/D conversion ended

Figure 3.11 (2). A/D Converter Mode Register (ADMOD)

ADREG0 (0060H)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	R							
After reset	Undefined							
Function	Channel 0 of A/D conversion result register is stored.							

Register for saving an A/D switch value for AN0

ADREG1 (0061H)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	R							
After reset	Undefined							
Function	Channel 1 of A/D conversion result register is stored.							

Register for saving an A/D switch value for AN1

ADREG2 (0062H)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	R							
After reset	Undefined							
Function	Channel 2 of A/D conversion result register is stored.							

Register for saving an A/D switch value for AN2

ADREG3 (0063H)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	R							
After reset	Undefined							
Function	Channel 3 of A/D conversion result register is stored.							

Register for saving an A/D switch value for AN3

Figure 3.11 (3). Register for Saving an A/D Switch Value (ADREG0 ~ 3)

3.11.1 Operation

(1) Analog Reference Voltage

High analog reference voltage is applied to the VREF pin.
The reference voltage between VREG and AGND is divided by 256 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, in fixed analog input mode, one channel is selected by ADMOD <ADCH1, 0> among four pins: AN0 to AN3.
In analog input channel scan mode, the number of channels to be scanned from AN0 is specified by ADMOD <ADCH1, 0>, such as AN0 → AN1, AN0 → AN1 → AN2, and AN0 → AN1 → AN2 → AN3.
When reset, A/D conversion channel register will be initialized to ADMOD <ADCH1, 0> = 00, so that AN0 pin will be selected.
The pins which are not used as analog input channel can be used as ordinary input port P5.

(3) Starting A/D Conversion

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

(4) A/D Conversion Mode

Both fixed A/D conversion channel mode and A/D conversion channel scan mode have two conversion modes, i.e., single and repeat conversion modes.
In fixed channel repeat mode, conversion of specified one channel is executed repeatedly.
In scan repeat mode, scanning from AN0, ··· → AN3 is executed repeatedly.
A/D conversion mode is selected by ADMOD <REPET, SCAN>.

(5) A/D Conversion Speed Selection

There are two A/D conversion speed modes: high speed mode and low speed mode. The selection is executed by ADMOD <ADCS> register.
When reset, ADMOD <ADCS> will be initialized to "0," so that high speed conversion mode will be selected.

(6) A/D Conversion End and Interrupt

- A/D conversion single mode

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

(7) Storing the A/D Conversion Result

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

(8) Reading the A/D Conversion Result

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

Setting example: ① When the analog input voltage of the AN3 pin is A/D converted and the result is stored in the memory address FF10H by A/D interrupt INTAD routine.

Main setting

INTE0AD	←	1	1	0	0	x	x	x	x	Enable INTAD and sets interrupt level 4.
ADM0D	←	x	x	0	0	0	1	1	1	Specify AN3 pin as an analog input channel and starts A/D conversion in high speed mode.

INTAD routine

A	←	ADREG3	The value of ADREG3 is read into the accumulator. Then the accumulator value is stored into memory at FF10H.
(FF10H)	←	A	

Setting example: ② When the analog pin voltage of AN0 ~ AN2 pin is A/D converted in high speed conversion channel scan repeat mode.

INTE0AD	←	1	0	0	x	x	x	x	x	Disable INTAD.
ADM0D	←	x	x	1	1	0	1	1	0	Start the A/D conversion of analog input channels AN0 ~ AN2 in the high-speed scan repeat mode.

Note: x; don't care –; no change

3.12 Watchdog Timer (Runaway Detecting Timer)

The TMP96C031F is containing watchdog timer of Runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt to notify the CPU of the malfunction, and outputs 0 externally from watchdog timer out pin WDTOUT to notify the peripheral devices of the malfunction.

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

A built-in function is used to stop the WDT count at bus release request (BUSRQ).

3.12.1 Configuration

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

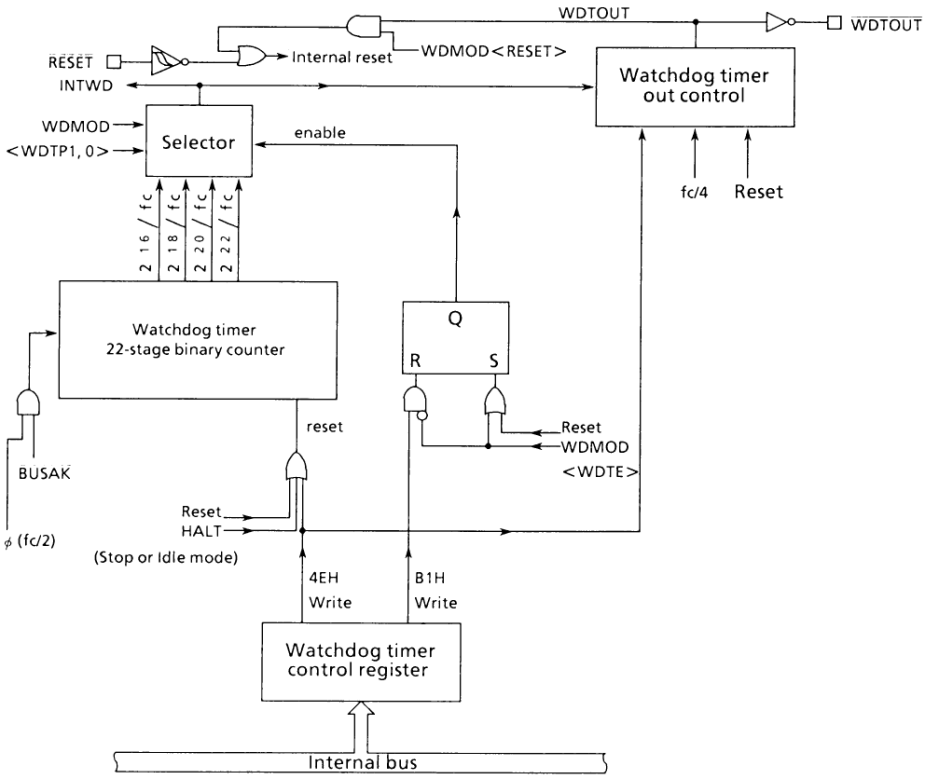


Figure 3.12 (1). Block Diagram of Watchdog Timer

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

Since the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0" due to a watchdog timer overflow, the peripheral devices can be reset.

Clearing the watchdog timer (by writing the clear code (4EH) to the WDCR) after disabling it sets 0 to output to 1 (Program example).

```
LDW   (WDMOD), 0B100H     ; disables
                            watchdog timer
LD    (WDCR), 4EH         ; writes clear code
SET   7, (WDMOD)          ; enables watchdog
                            timer 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 (800ns to 2.0μs @ 20MHz) and resets itself.

The $\overline{\text{WDTOUT}}$ (also used as P67) is multiplexed with pin PG13; setting must be done using the port 6 control register, P6CRH ($\overline{\text{WDTOUT}}$ pin is set after reset).

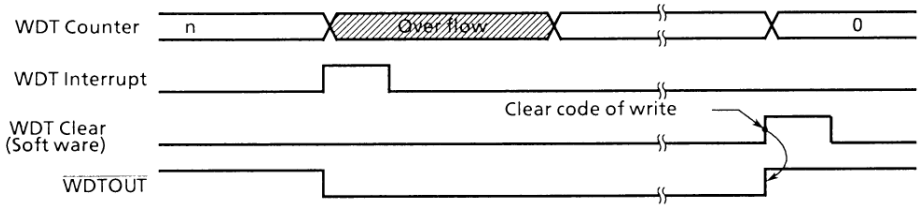


Figure 3.12 (2). Normal Mode

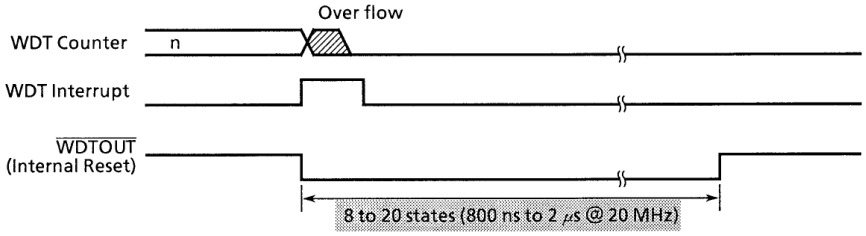


Figure 3.12 (3). Reset Mode

3.12.2 Control Registers

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

(1) Watchdog Timer Mode Register (WDMOD)

① Setting the detecting time of watchdog timer <WDTP>

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

② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD <WDTE> is initialized to "1" 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 the binary counter of the watchdog timer function.

- Disable control

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

- Enable control

Set WDMOD <WDTE> to "1".

- Watchdog timer clear control

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

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

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 1: Enable	Select detecting time 00: 2^{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: Internally connects WDT out to the reset pin	1: Drive the pin even in STOP mode

DRVE (explanation by stop mode)

Watchdog timer out control

0	—
1	Connects WDT out to a reset

Select the standby mode HALT instruction

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

Select the detecting period of watchdog timer

0	2^{14}/fc (approx. 0.8ms @20MHz)
1	2^{16}/fc (approx. 3.3ms @20MHz)

Select the detecting time of watchdog timer

00	2^{16}/fc (approx. 3.3ms @20MHz)
01	2^{18}/fc (approx. 13.1ms @20MHz)
10	2^{20}/fc (approx. 52.4ms @20MHz)
11	2^{22}/fc (approx. 209ms @20MHz)

Watchdog timer Enable / Disable control

0	Disable
1	Enable

Figure 3.12 (4). Watchdog Timer Mode Register

WDCR (005DH)

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

Disable/clear WDT

B1H	Disable code
4EH	Clear code
Others	—

Figure 3.12 (5). Watchdog Timer Control Register

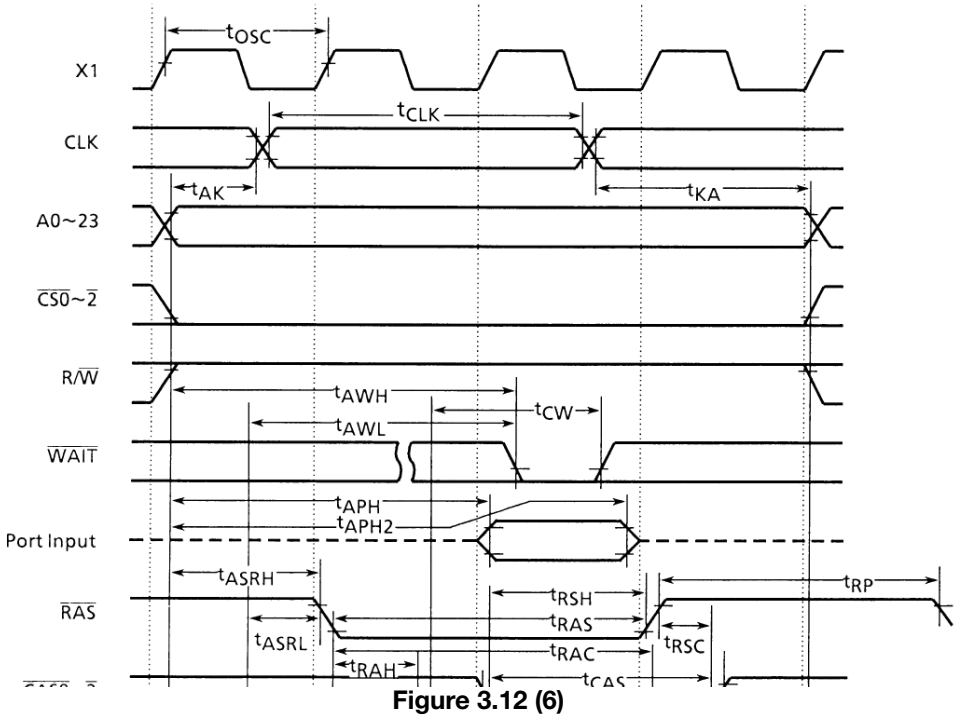


Figure 3.12 (6)

3.12.3 Operation

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

The watchdog timer restarts operation immediately after resetting is released.

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

However, the function can be disabled when entering the RUN mode.

Example: ① Clear the binary counter

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

② Set the watchdog timer detecting time to 2^{18}/fc

WDMOD	←	1	0	1	–	–	–	x	x	

③ Disable the watchdog timer

WDMOD	←	0	–	–	–	–	–	x	x	Clear WDTE to "0".
WDCR	←	1	0	1	1	0	0	0	1	Write disable code (B1H).

④ Set 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.

2) Writing 1 to the P4FC <BUSWDT> register halts count by the WDT binary counter at bus release due to the bus request signal, BUSRQ.

3.13 Dynamic RAM (DRAM Controller

The TMP96C031F consists of a control circuit to refresh DRAM, an access circuit to perform read/write, and an address decoder.

Figure 3.13 (1) shows a block diagram of the DRAM controller.

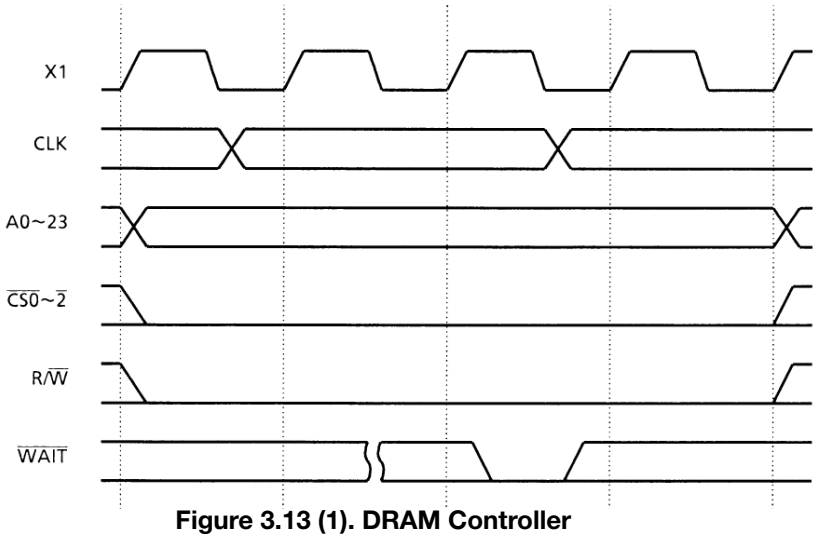


Figure 3.13 (1). DRAM Controller

3.13.1 Control Register

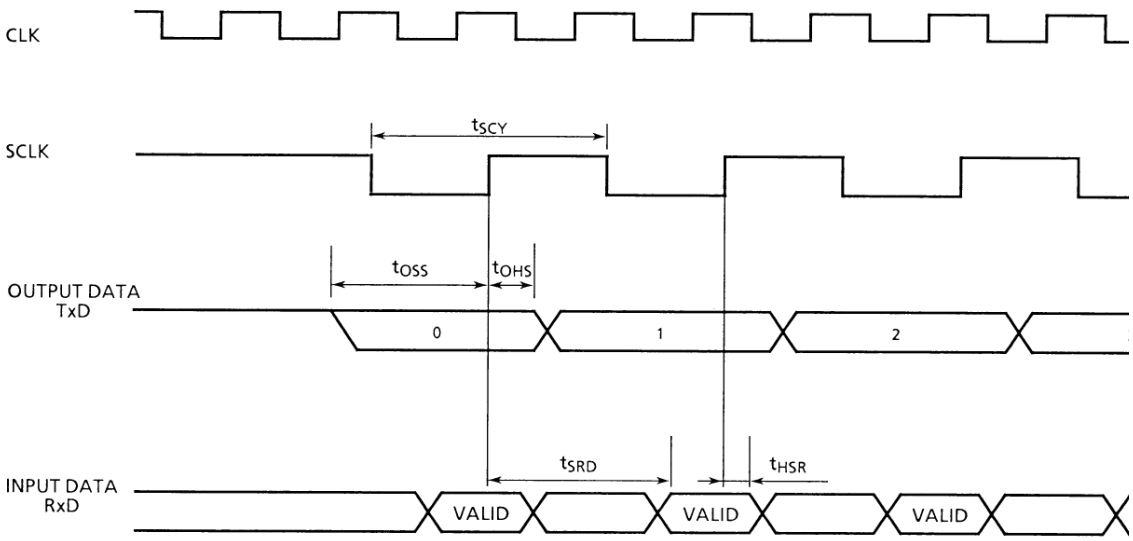


Figure 3.13 (2). Chip Select Wait Control Register (B3CS)

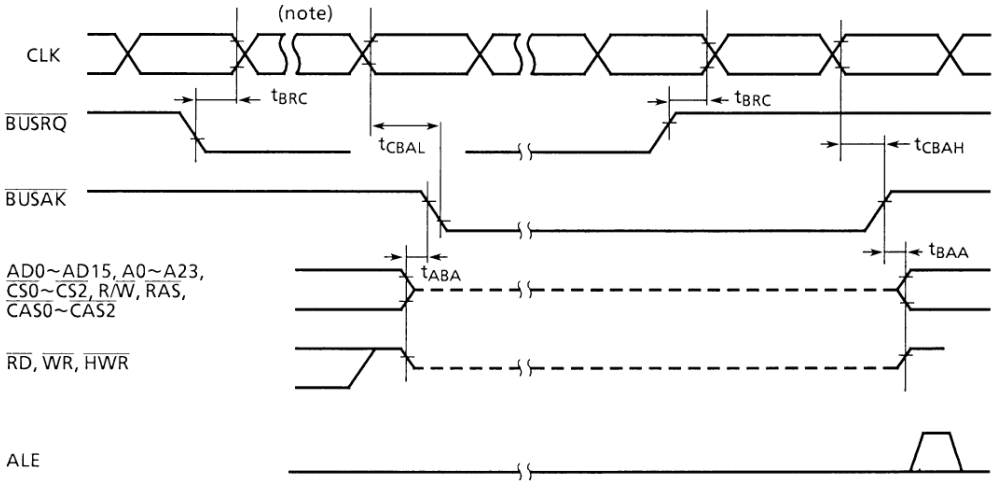


Figure 3.13 (3). Port 4 Function Register

DREFCR (004BH)

	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: 15 states 001: 31 states 010: 62 states 011: 78 states 100: 97 states 101: 109 states 110: 124 states 111: 154 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	Do 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 3.13 (2).

Dummy cycle control

0	Prohibit dummy cycle
1	Execute dummy cycle

Figure 3.13 (4). Refresh Control Register

3.13.2 Operation Description

(1) Read/Write Control

The read/write controller outputs valid signals $\overline{RAS}$ and $\overline{CAS}$ to DRAM when address space specified by the internal address decoder (chip select 2 $\overline{CS3}/\overline{CAS}$) is accessed.

In addition, a $\overline{DMUX}$ signal is output for row address/column address switching.

Figure 3.13 (6) shows the $\overline{RAS}$, $\overline{CAS}$, and $\overline{DMUX}$ output timing diagram during memory access cycle.

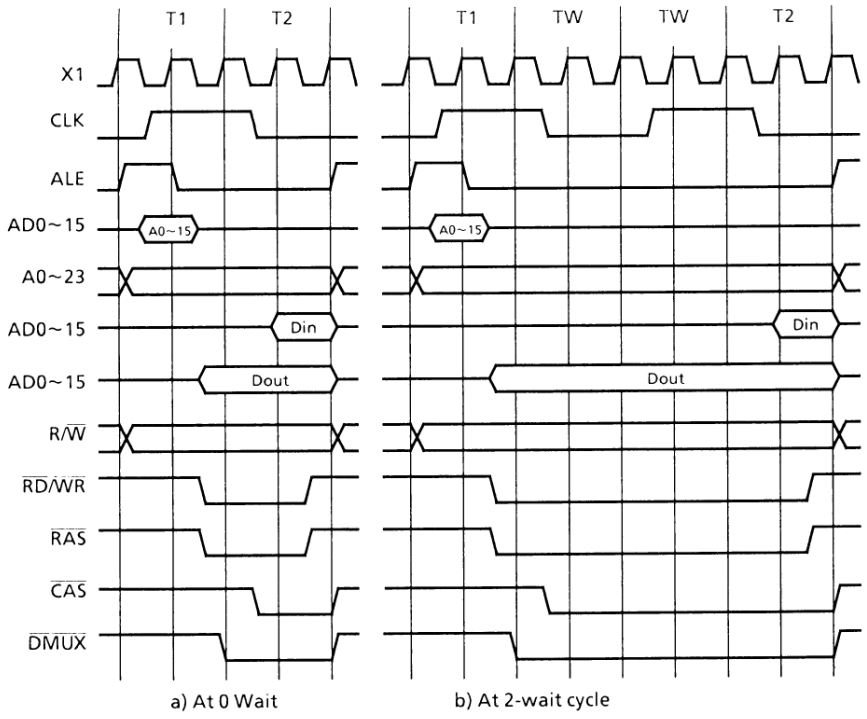


Figure 3.13 (5). Memory Access Cycle Timing

How to set the registers is described next.

① Setting the $\overline{RAS}$, $\overline{CAS}$, $\overline{DMUX}$, and $\overline{RFSH}$ output

Figure 3.13 (2) shoes the structure of the chip select wait control register B3CS. B3CS <B3E> can be used to control the output of $\overline{CS3}$/$\overline{CAS}$ and B3CS <B3CAS> can be used to control $\overline{CAS}$ selection.

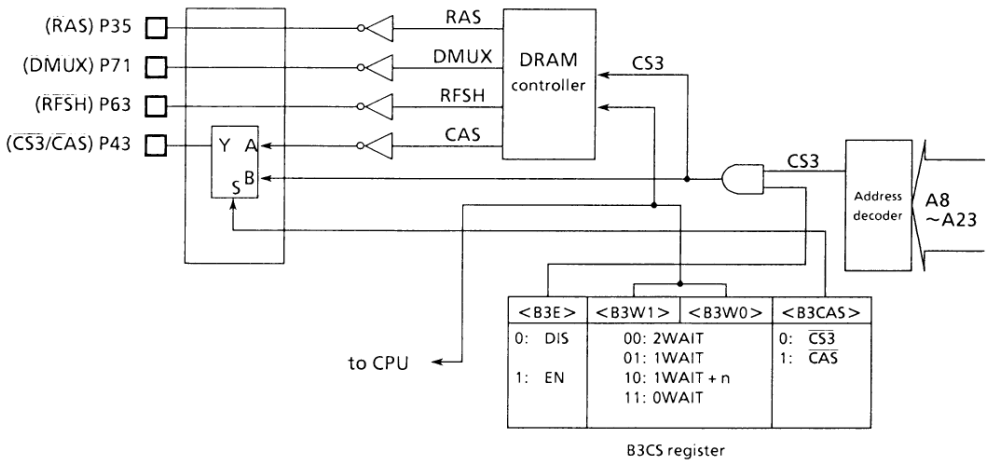


Figure 3.13 (6). Relationship Between Address Decoder and DRAM Controller

The $\overline{RAS}$, $\overline{CAS}$, $\overline{DMUX}$, and $\overline{RFSH}$ signals must be set with the corresponding port control register because they are multiplexed with P35, P43, P71, and P63 respectively.

② Inserting WAIT

WAIT insertion during read/write control can be set with the register B3CS <B3W1, 0>.

(2) Refresh Controller

The TMP96C031F can output $\overline{RAS}$/$\overline{CAS}$ used to refresh in DRAM. At the same time the state signal $\overline{RFSH}$ which indicates a refresh cycle is output.
DRAM can be refreshed easily because $\overline{RAS}$/$\overline{CAS}$ 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: 15 to 154 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, TMP96C031F enables the $\overline{CAS}$ before $\overline{RAS}$ output to be set with the refresh controller register value according to the system clock and DRAM that are being used.
Figure 3.13 (2) shows a timing example for $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.

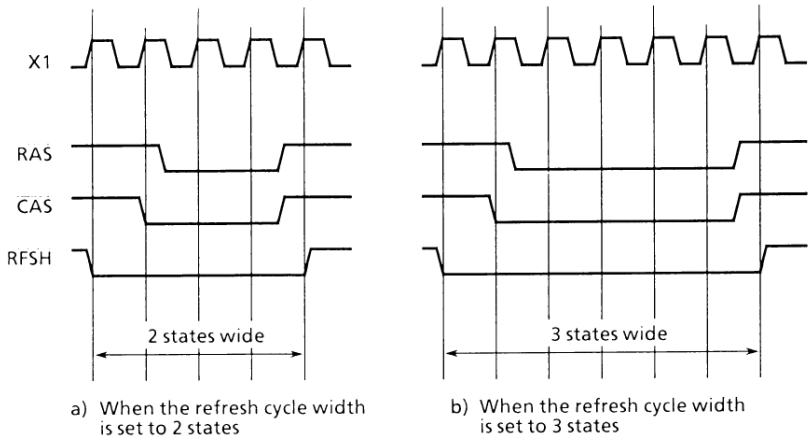


Figure 3.13 (7). Refresh Cycle Timing Example

How to set the register is described next.
Figure 3.13 (4) 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 20MHz and the DRAM refresh cycle is to be 16μs, set these bits to “111”.

Table 3.13 Refresh Cycle Insertion Interval

Refresh Cycle			Insertion Interval (states)	Frequency (f_{OSC})						
RS2	RS1	RS0		4MHz	8MHz	10MHz	12.5MHz	14MHz	16MHz	20MHz
0	0	0	15	7.5	3.75	3.0	2.4	2.14	1.88	1.5μs
0	0	1	31	(15.5)	7.55	6.2	4.96	4.43	3.88	3.1
0	1	0	62	31.0	(15.5)	12.4	9.92	8.86	7.75	6.2
0	1	1	78	39.0	19.5	(15.6)	12.48	11.14	9.75	7.8
1	0	0	97	48.5	24.25	19.4	(15.52)	13.86	12.13	9.7
1	0	1	109	54.5	27.25	21.8	17.44	(15.57)	13.63	10.9
1	1	0	124	62.0	31.0	24.8	19.84	17.72	(15.5)	12.4
1	1	1	154	77.0	38.5	30.8	24.7	22.0	19.3	(15.4)

(Unit : μs)

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

③ Refresh cycle control

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

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

This mode is used when CPU or DRAM control is halted with a HALT (IDEL, STOP) instruction while refreshing with $\overline{CAS}$ before $\overline{RAS}$ interval refresh mode (hereafter referred to as interval mode).
However, $\overline{RFSH}$ is not output. ("1" is output.).

Figure 3.13 (8) shows the self refresh mode timing diagram.

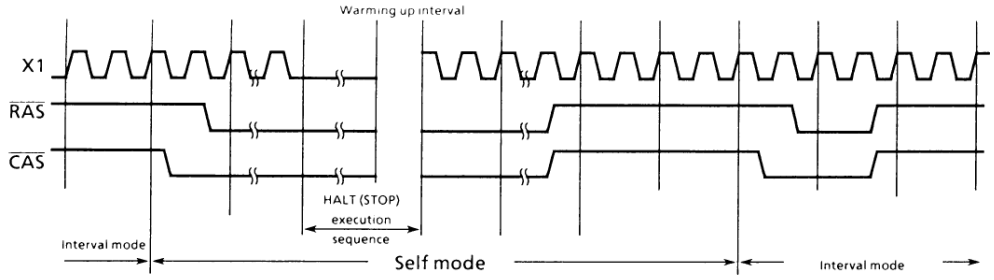


Figure 3.13 (8). Self Refresh Cycle Timing

This mode is executed as follows. First, the settings are made for normal interval mode Then, B3CS <SRFC> is set to "0" just before a HALT instruction to perform one normal refresh. Then, the $\overline{CAS}$ pin and $\overline{RAS}$ pin are kept at low level and self refresh mode is entered. Set B3CS <SRFC> to "1" to cancel this mode and return to normal $\overline{CAS}$ before $\overline{RAS}$ refresh mode. (The first $\overline{CAS}$ before $\overline{RAS}$ refresh is performed immediately after cancellation because the refresh counter is cleared.)

(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" and cancelled by setting it to "0". (The <RC> bit need not be changed.)

The dummy cycle width is fixed to 4 states.

Figure 3.13 (9) shows the $\overline{CAS}$ before $\overline{RAS}$ dummy cycle timing.

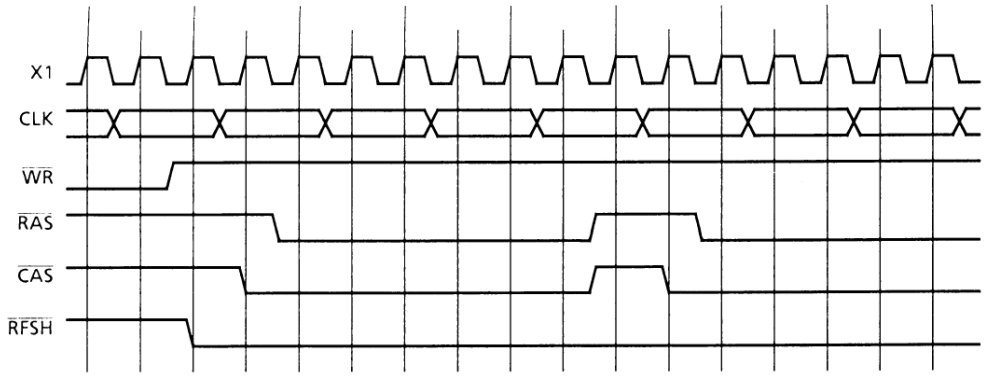


Figure 3.13 (9). $\overline{CAS}$ before $\overline{RAS}$ Dummy Cycle Timing (Fixed to 4 states)

3.13.4 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. Figure 3.13 (8) shows the timing in this case.

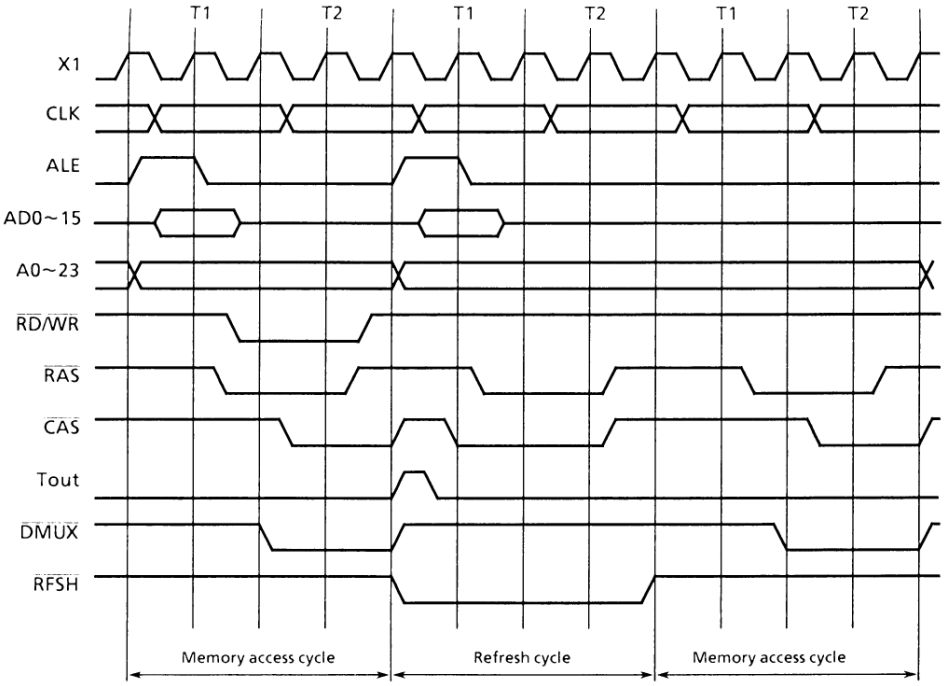
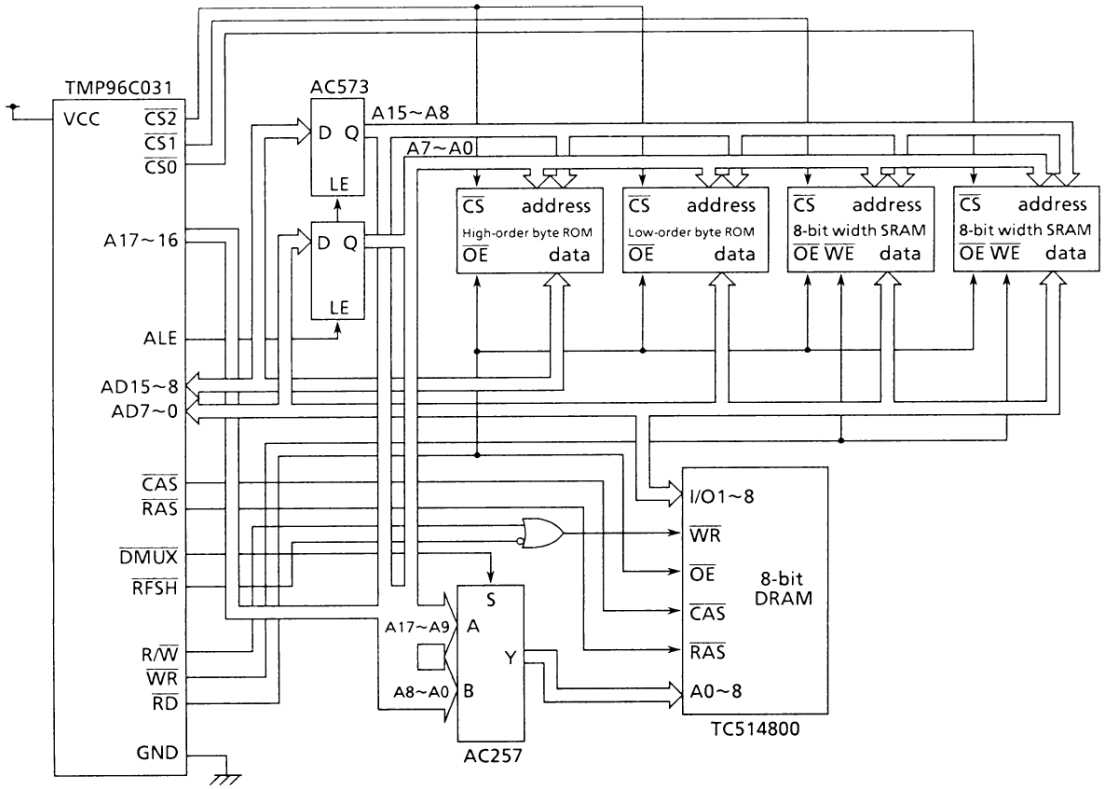


Figure 3.13 (101). Timing Diagram when Refresh Cycle is Inserted in Memory Access Cycle

3.13.4 Connection Example



4. Electrical Characteristics

4.1 Absolute Maximum (TMP96C031F)

Symbol	Parameter	Rating	Unit
V_{CC}	Power Supply Voltage	-0.5 ~ 6.5	V
V IN	Input Voltage	-0.5 ~ V_{CC} + 0.5	V
Σ IOL	Output Current (total)	100	mA
Σ IOH	Output Current (total)	-100	mA
PD	Power Dissipation (Ta = 70°C)	600	mW
T SOLDER	Soldering Temperature (10s)	260	°C
T STG	Storage Temperature	-65 ~ 150	°C
T OPR	Operating Temperature	-20 ~ 70	°C

4.2 DC Characteristics (TMP96C031F)

V_{cc} = 5V ± 10%, Ta = -20 ~ 70°C (Typical values are for Ta = 25°C and V_{cc} = 5V)

Symbol	Parameter	Min	Max	Unit	Test Condition
V IL	Input Low Voltage (AD0-15)	-0.3	0.8	V	
V IL1	P2, P3, P4, P5, P6, P7, P8, P9	-0.3	$0.3V_{cc}$	V	
V IL2	$\overline{RESET}$, $\overline{NMI}$, INTO (P87)	-0.3	$0.25V_{cc}$	V	
V IL3	AM8/$\overline{EA}$	-0.3	0.3	V	
V IL4	X1	-0.3	$0.2V_{cc}$	V	
V IH	Input High Voltage (AD0-15)	2.2	V_{cc} + 0.3	V	
V IH1	P2, P3, P4, P5, P6, P7, P8, P9	$0.7V_{cc}$	V_{cc} + 0.3	V	
V IH2	$\overline{RESET}$, $\overline{NMI}$, INTO (P87)	$0.75V_{cc}$	V_{cc} + 0.3	V	
V IH3	$\overline{EA}$	V_{cc} - 0.3	V_{cc} + 0.3	V	
V IH4	X1	$0.8V_{cc}$	V_{cc} + 0.3	V	
V OL	Output Low Voltage		0.45	V	I OL = 1.6mA
V OH	Output High Voltage	2.4		V	I OH = -400μA
V OH1		$0.75V_{cc}$		V	I OH = -100μA
V OH2		$0.9V_{cc}$		V	I OH = - 20μA
I DAR	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	V EXT - 1.5V R EXT = 1.1KΩ
I LI	Input Leakage Current	0.02 (Typ)	±5	μA	$0.0 \le V_{in} \le V_{cc}$
I LO	Output Leakage Current	0.05 (Typ)	±10	μA	$0.2 \le V_{in} \le V_{cc}$ - 0.2
I_{cc}	Operating Current (RUN) IDLE STOP (Ta = -20 ~ 70°C) STOP (Ta = 0 ~ 50°C)	30 (Typ) 2.0 (Typ) 0.2 (Typ)	60 10 50 10	mA mA μA μA	tosc = 20MHz $0.2 \le V_{in} \le V_{cc}$ - 0.2 $0.2 \le V_{in} \le V_{cc}$ - 0.2
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V IL2 = $0.2V_{cc}$, V IH2 = $0.8V_{cc}$
R RST	RESET Pull Up Register	50	150	KΩ	
C IO	Pin Capacitance		10	pF	tosc = 1MHz
V TH	Schmitt Width $\overline{RESET}$, $\overline{NMI}$, INTO (P50)	0.4	1.0 (Typ)	V	
R K	Pull Down/Up Register	50	150	KΩ	

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

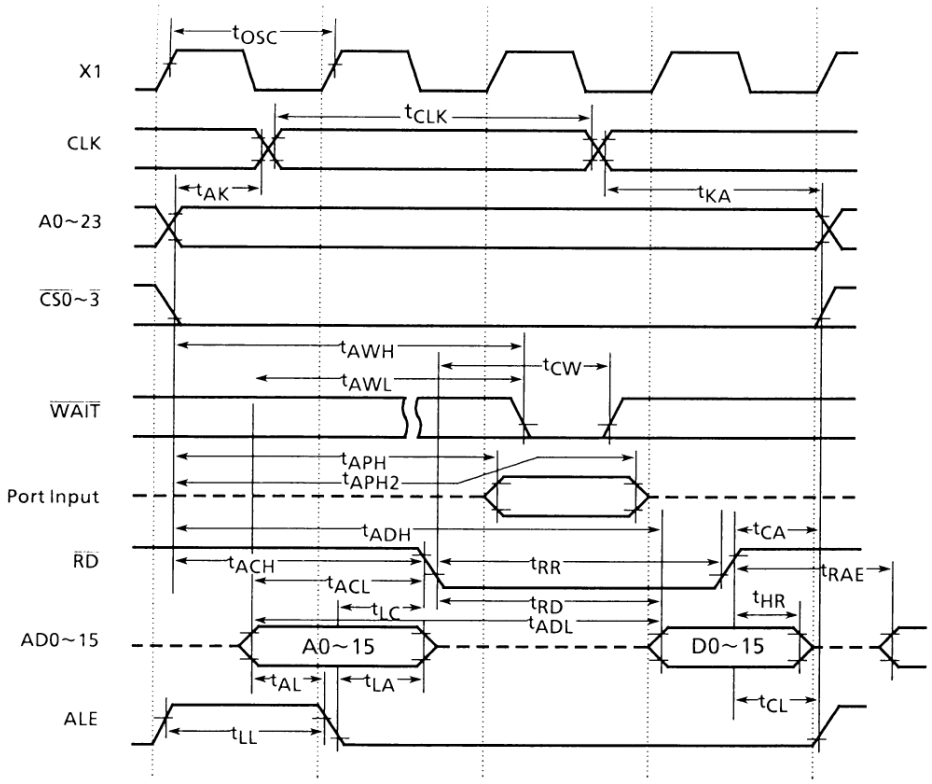
4.3 AC Electrical Characteristics (TMP96C031F) V_{cc} = 5V±10%, Ta = -20 ~ 70°C (4MHz ~ 20MHz)

No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{OSC}	Osc. Period (= x)	50	250	62.5		50		ns
2	t_{CLK}	CLK width	2x - 40		85		60		ns
3	t_{AK}	A0 - 23 Valid→CLK Hold	0.5x - 20		11		5		ns
4	t_{KA}	CLK Valid→A0 - 23 Hold	1.5x - 70		24		5		ns
5	t_{AL}	A0-15 Valid→ALE fall	0.5x - 15		16		10		ns
6	t_{LA}	ALE fall→A0 - 15 Hold	0.5x - 15		16		10		ns
7	t_{LL}	ALE High width	x - 40		23		10		ns
8	t_{LC}	ALE fall→$\overline{RD}$/$\overline{WR}$ fall	0.5x - 30		1		-5		ns
9	t_{CL}	$\overline{RD}$/$\overline{WR}$ rise→ALE rise	0.5x - 20		11		5		ns
10	t_{ACL}	A0 - 15 Valid→$\overline{RD}$/$\overline{WR}$ fall	x - 25		38		25		ns
11	t_{ACH}	A0 - 23 Valid→$\overline{RD}$/$\overline{WR}$ fall	1.5x - 50		44		25		ns
12	t_{CA}	$\overline{RD}$/$\overline{WR}$ rise→A0 - 23 Hold	0.5x - 20		11		5		ns
13	t_{ADL}	A0 - 15 Valid→D0 - 15 input		3.0x - 45		143		105	ns
14	t_{ADH}	A0 - 23 Valid→D0 - 15 input		3.5x - 65		154		110	ns
15	t_{RD}	$\overline{RD}$ fall→D0 - 15 input		2.0x - 50		75		50	ns
16	t_{RR}	$\overline{RD}$ Low width	2.0x - 40		85		60		ns
17	t_{HR}	$\overline{RD}$ rise→D0 - 15 Hold	0		0		0		ns
18	t_{RAE}	$\overline{RD}$ rise→A0 - 15 output	x - 15		48		35		ns
19	t_{WW}	$\overline{WR}$ Low width	2.0x - 40		85		60		ns
20	t_{DW}	D0 - 15 Valid→$\overline{WR}$ rise	2.0x - 50		75		50		ns
21	t_{WD}	$\overline{WR}$ rise→D0 - 15 Hold	0.5x - 10		21		15		ns
22	t_{AEH}	A0 - 23 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	t_{AWL}	A0 - 15 Valid→$\overline{WAIT}$ input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	t_{CW}	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	t_{APH}	A0 - 23 Valid→PORT input		2.5x - 120		36		5	ns
26	t_{APH2}	A0 - 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t_{CP}	$\overline{WR}$ rise→PORT Valid		200		200		200	ns

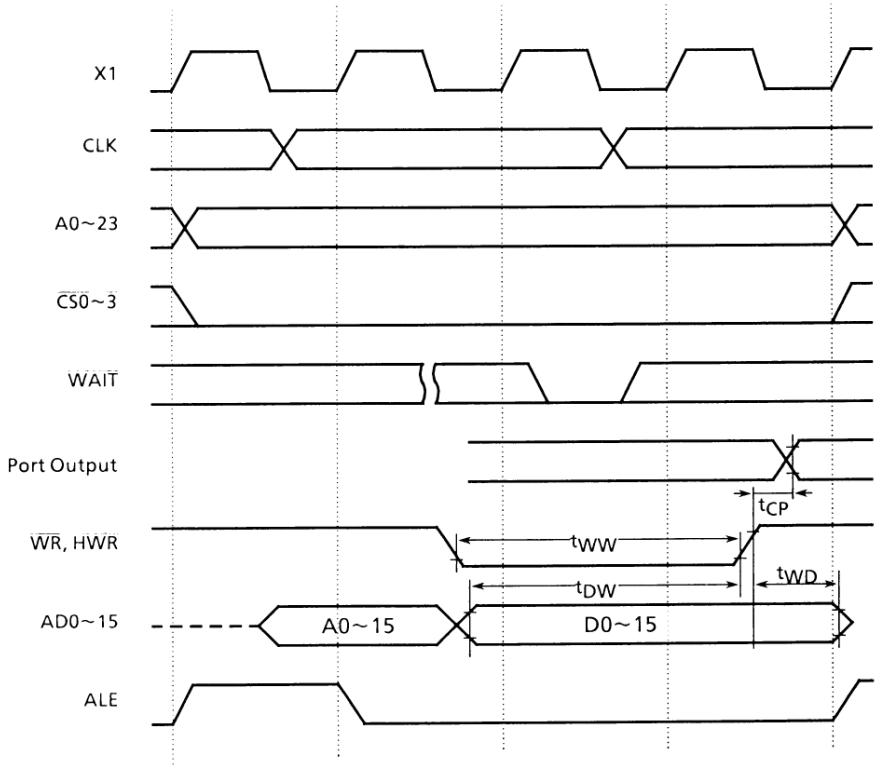
AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF
  (However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, CLK, $\overline{CS0}$ ~ $\overline{CS3}$)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)
  High 0.8Vcc /Low 0.2Vcc (Except for AD0 ~ AD15)

(1) Read Cycle


tOSC
X1
tCLK
CLK
tAK
tKA
A0~23
CS0~3
tAWH
tCW
tAWL
WAIT
tAPH
tAPH2
Port Input
tADH
tCA
RD
tACH
tRR
tRAE
tACL
tHR
tLC
tRD
tADL
AD0~15
A0~15
D0~15
tAL
tLA
tCL
ALE
tLL

(2) Write Cycle



4.4 DRAM Control AC Characteristics (TMP96C031F)

V_{cc} = 5V±10% TA = -20 ~ 70°C (4MHz ~ 20MHz)

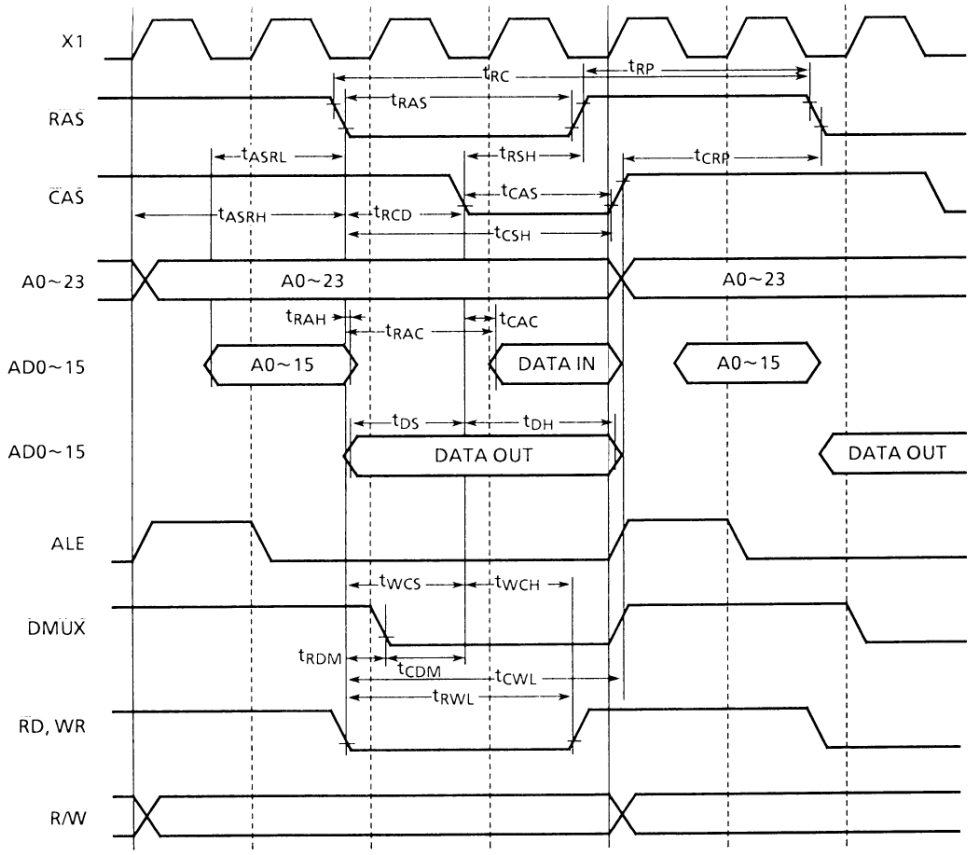
No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{RC}	$\overline{RAS}$ cycle time	4x - 10		240		190		ns
2	t_{RAC}	$\overline{RAS}$ fall → data input		2x - 20		105		80	ns
3	t_{CAC}	$\overline{CAS}$ fall → data input		1x - 25		38		25	ns
4	t_{RP}	$\overline{RAS}$ high pulse width	2x - 30		95		70		ns
5	t_{RAS}	$\overline{RAS}$ low pulse width	2x - 10		115		90		ns
6	t_{RSH}	$\overline{CAS}$ fall → $\overline{RAS}$ rise	1x - 25		38		25		ns
7	t_{CSH}	$\overline{RAS}$ fall → $\overline{CAS}$ rise	2x - 20		105		80		ns
8	t_{CAS}	$\overline{CAS}$ low pulse width	1.5x - 10		84		65		ns
9	t_{RCD}	$\overline{RAS}$ fall → $\overline{CAS}$ fall	1x - 15		48		35		ns
10	t_{CRP}	$\overline{CAS}$ rise → $\overline{RAS}$ rise	1.5x - 50		44		25		ns
11	t_{RAH}	$\overline{RAS}$ fall → A0 - 15 hold	0		0		0		ns
12	t_{ASRL}	A_{0-15} valid → $\overline{RAS}$ fall	1x - 10		53		40		ns
13	t_{ASRH}	A_{0-23}valid → $\overline{RAS}$ fall	1.5x - 10		84		65		ns
14	t_{RWL}	$\overline{WR}$ fall → $\overline{RAS}$ rise	2x - 50		75		50		ns
15	t_{CWL}	$\overline{WR}$ fall → $\overline{CAS}$ rise	2x - 50		75		50		ns
16	t_{DS}	Data output → $\overline{CAS}$ fall setup	1x - 30		33		20		ns
17	t_{DH}	$\overline{CAS}$ fall → data output hold	1.5x - 50		44		25		ns
18	t_{DHR}	$\overline{RAS}$ fall → data output hold	2.5x - 50		106		75		ns
19	t_{WCS}	$\overline{WR}$ fall → CAS fall setup	1x - 30		33		20		ns
20	t_{WCH}	$\overline{CAS}$ fall → WR hold	1x - 30		33		20		ns
21	t_{RDM}	$\overline{RAS}$ fall → DMUX fall		0.5x - 10	21		15		ns
22	t_{CDM}	$\overline{DMUX}$ fall → $\overline{CAS}$ fall	0.5x - 10		21		15		ns
23	t_{CHR*1}	$\overline{RAS}$ fall → $\overline{CAS}$ rise	2x - 50		75		50		ns
24	t_{RPC*}	$\overline{RAS}$ rise → $\overline{CAS}$ fall	1.5x + 0		64		45		ns
25	t_{CP*}	$\overline{CAS}$ high pulse width	1.5x + 0		64		45		ns
26	t_{CSR*}	$\overline{CAS}$ fall → $\overline{RAS}$ fall	0.5x - 10		21		15		ns
27	t_{RASS*2}	$\overline{RAS}$ low pulse width	2000x		125		100		ns
28	t_{RPS*2}	$\overline{RAS}$ precharge time	4x - 20		230		180		ns
29	t_{CHS*2}	$\overline{CAS}$ hold time	0		0		0		ns
30	t_{CFL*}	$\overline{RFSH}$ fall → $\overline{CAS}$ fall	1x - 10		53		40		ns
31	t_{CFH*}	$\overline{CAS}$ rise → $\overline{RFSH}$ rise	0.5x - 10		21		15		ns

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

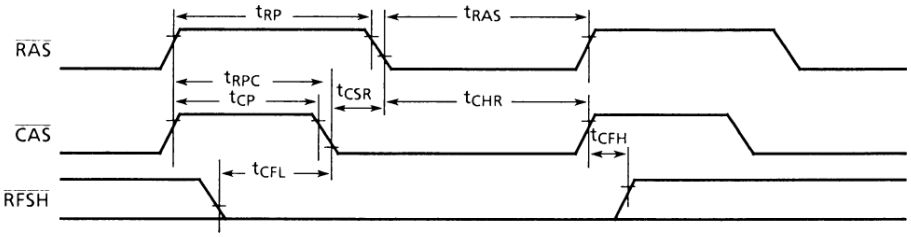
AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF
  (However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$ $\overline{RAS}$)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)
  High 0.8Vcc/Low 0.2Vcc (Except for AD0 ~ AD15)

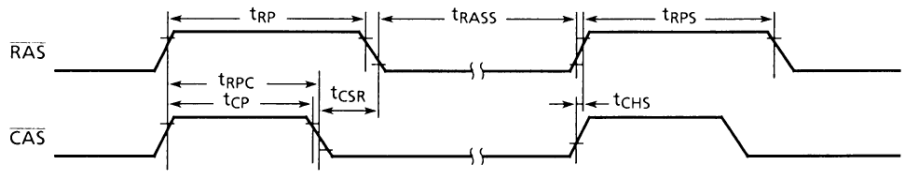
(1) Read/Write Access Cycle



(2) $\overline{CAS}$ before $\overline{RAS}$ Interval Refresh Cycle



(3) $\overline{CAS}$ before $\overline{RAS}$ Self-Refresh Cycle



4.5 A/D Conversion Characteristics (TMP96C031F)

V_{cc} = 5V±10% TA = -20 ~ 70°C

Symbol	Parameter	Min	Typ	Max	Unit
V_{REF}	Analog reference voltage	V_{cc} - 1.5		V_{cc}	V
A_{GND}	Analog reference voltage	V_{ss}		V_{ss}	
V_{AIN}	Analog input voltage range	V_{ss}		V_{cc}	
I_{REF}	Analog current for analog reference voltage		0.5	1.5	mA
Error (Quantize error of ±0.5 LSB not included)	Total error (TA = 25°C, V_{CC} = VREF = 5.0V)		1.0		LSB
	Total error			2.5	

4.6 Serial Channel Timing - I/O Interface Mode

V_{cc} = 5V±10% TA = -20 ~ 70°C

(1) SCLK Input Mode

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	16x		1		0.8		μs
t_{OSS}	Output Data→rising edge of SCLK	t_{SCY}/2 - 5x - 50		137		100		ns
t_{OHS}	SCLK rising edge→output data hold	5x - 100		212		150		ns
t_{HSR}	SCLK rising edge→input data hold	0		0		0		ns
t_{SRD}	SCLK rising edge→effective data input		t_{SCY} - 5x - 100		587		450	ns

(2) SCLK Output Mode

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (programmable)	16x	8192x	1	512	0.8	409.6	μs
t_{OSS}	Output Data→rising edge of SCLK	t_{SCY} - 2x - 150		725		550		ns
t_{OHS}	SCLK rising edge→output data hold	2x - 80		45		20		ns
t_{HSR}	SCLK rising edge→input data hold	0		0		0		ns
t_{SRD}	SCLK rising edge→effective data input		t_{SCY} - 2x - 150		725		550	ns

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

V_{cc} = 5V±10% TA = -20 ~ 70°C

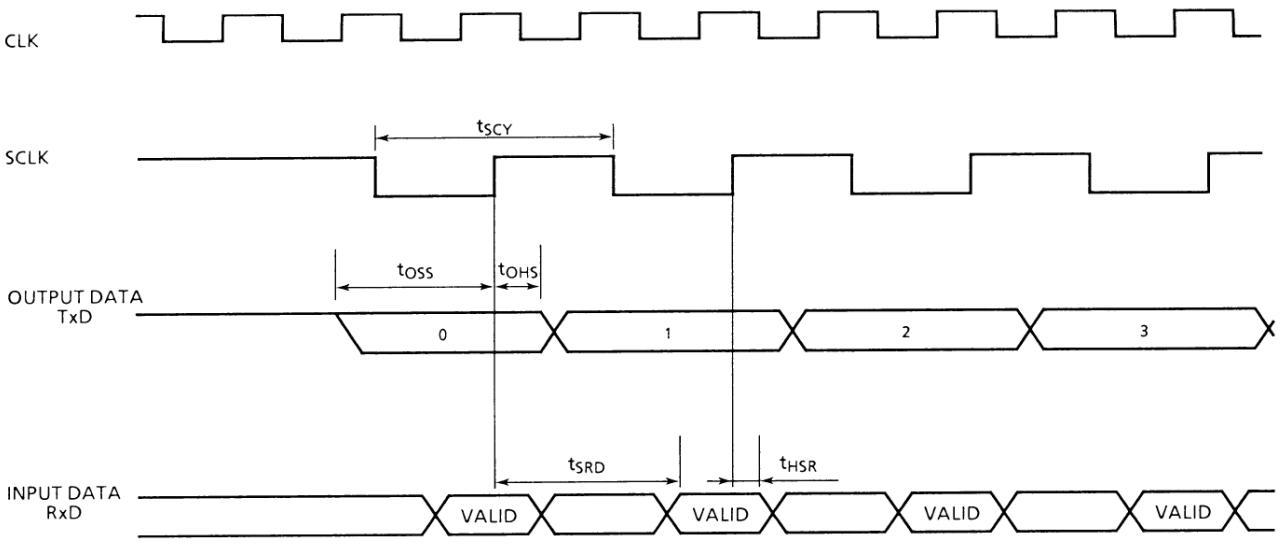
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock cycle	8x + 100		600		500		ns
t_{VCKL}	Low level clock pulse width	4x + 40		290		240		ns
t_{VCKH}	High level clock pulse width	4x + 40		290		240		ns

4.8 Interrupt Operation

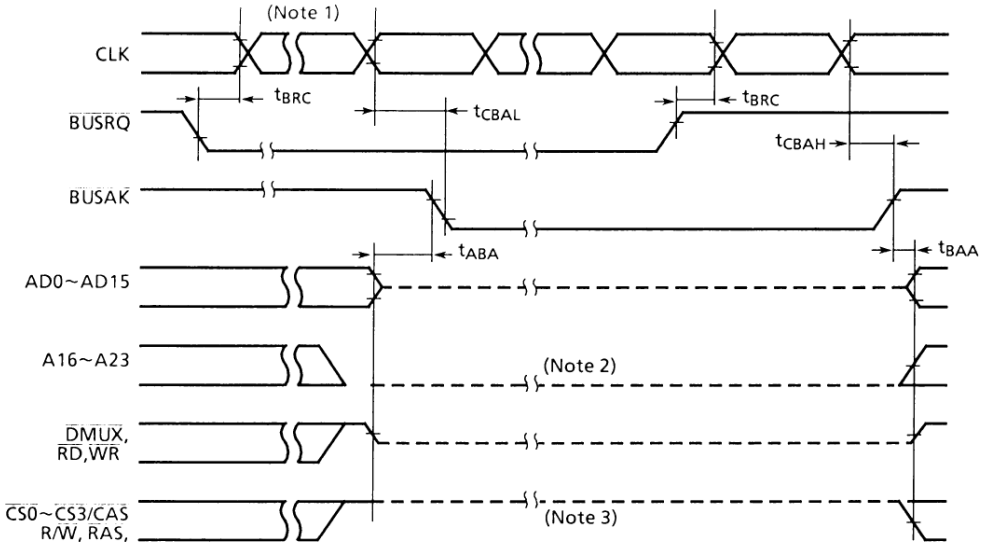
V_{cc} = 5V±10% Ta = -20 ~ 70°C

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{NMI}$, INT0 Low level pulse width	4x		250		200		ns
t_{INTAH}	$\overline{NMI}$, INT0 High level pulse width	4x		250		200		ns
t_{INTBL}	INT1 ~ INT7 Low level pulse width	8x + 100		600		500		ns
t_{INTBH}	INT1 ~ INT7 High level pulse width	8x + 100		600		500		ns

4.9 Timing Chart for I/O Interface Mode



4.10 Timing Chart for Bus Request/BUS Acknowledge



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{BRC}	$\overline{BUSRQ}$ setup time for CLK	120		120		120		ns
t_{CBAL}	CLK→$\overline{BUSAK}$ falling edge		1.5x + 120		214		220	ns
t_{CBAH}	CLK→$\overline{BUSAK}$ rising edge		0.5x + 40		71		65	ns
t_{ABA}	Output buffer is off to $\overline{BUSAK}$ ↓	0	80	0	80	0	80	ns
t_{BAA}	$\overline{BUSAK}$ ↑ output buffer is on.	0	80	0	80	0	80	ns

Note 1: The Bus will be released after the $\overline{WAIT}$ request is inactive, when the $\overline{BUSRQ}$ is set to "0" during "Wait" cycle.
Note 2: An internal programmable pull-down resistor must be connected.
Note 3: An internal programmable pull-up resistor must be connected.

5. Table of Special Function Registers
(SFR; Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 128-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

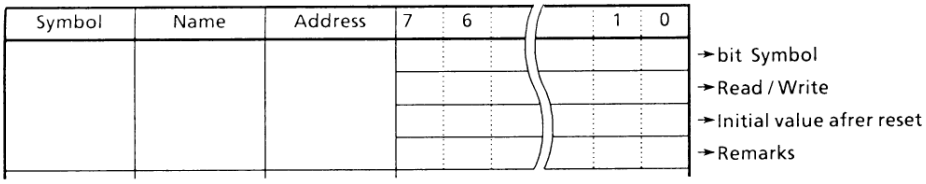


Table 5 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
000000H		20H	TRUN	40H	MSAR0	60H	ADREG0
1H		21H		41H	MAMR0	61H	ADREG1
2H		22H	TREG0	42H	MSAR1	62H	ADREG2
3H		23H	TREG1	43H	MAMR1	63H	ADREG3
4H		24H	T01TMOD	44H	MSAR2	64H	
5H		25H	TFFCR	45H	MAMR2	65H	
6H	P2	26H	TREG2	46H	MSAR3	66H	
7H	P3	27H	TREG3	47H	MAMR3	67H	
8H	P2CR	28H	T23MOD	48H		68H	
9H	P2FC	29H	TRDC	49H		69H	B0CS
AH	P3CRL	2AH		4AH		6AH	B1CS
BH	P3CRH	2BH		4BH	DREFCR	6BH	B2CS
CH	P4	2CH		4CH	PG0REG	6CH	B3CS
DH	P5	2DH		4DH	PG1REG	6DH	
EH		2EH		4EH	PG01CR	6EH	
FH		2FH		4FH		6FH	
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE01
11H		31H	TREG4H	51H	SC0CR	71H	INTE23
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE45
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTE67
14H	P6CRL	34H	CAP1L	54H	SC1BUF	74H	INTET10
15H	P7CRL	35H	CAP1H	55H	SC1CR	75H	INTET32
16H	P6CRH	36H	CAP2L	56H	SC1MOD	76H	INTET54
17H	P7CRH	37H	CAP2H	57H	BR1CR	77H	INTES0
18H		38H	T4MOD	58H	ODE	78H	INTES1
19H		39H	TFF4CR	59H		79H	INTEAD
1AH		3AH	T45CR	5AH		7AH	IIMC0
1BH		3BH		5BH		7BH	IIMC1
1CH		3CH		5CH	WDMOD	7CH	DMA0V
1DH		3DH		5DH	WDCR	7DH	DMA1V
1EH		3EH		5EH	ADMOD	7EH	DMA2V
1FH		3FH		5FH		7FH	DMA3V

(1) I/O Port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
			0	0	0	0	0	0	0	0
P3	PORT3	07H			P35	P34	P33	P32	P31	P30
					R/W					
					1	1	1	1	1	1
			Input mode (Pulled-up)							
P4	PORT4	0CH					P43	P42	P41	P40
							R/W			
							1	0	1	1
							Output mode			
P5	PORT5	0DH					P53	P52	P51	P50
							R			
							Input mode			
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P7	PORT7	13H		P76	P75	P74	P73	P72	P71	P70
				R/W						
				1	1	1	1	1	1	1
				Input mode						

Read/Write
R/W ; Either read or write is possible
R ; Only read is possible
W ; Only write is possible
Prohibit RWM ; Prohibit Read Modify Write. (Prohibit RES/SET/TSET/CHG/STCF/ANDCF/ORCF/XORCF Instruction)

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

Symbol	Name	Address	7	6	5	4	3	2	1	0
P2CR	PORT2 Control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P2FC">>							
P2FC	PORT2 Function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/ P2CR = 00 : IN, 01 : OUT, 10 : –, 11 : A23 - 16							
P3CRL	PORT3 Control Low	0AH (Prohibit RMW)	P33C1	P33C0	P32C1	P32C0	P31C1	P31C0	P30C1	P30C0
			W							
			0	0	0	0	0	0	0	0
			00 : PORT input 01 : PORT output 10 : $\overline{\text{BUSAK}}$ 11 : –		00 : PORT input 01 : PORT output 10 : $\overline{\text{BUSRQ}}$ 11 : –		00 : PORT input 01 : PORT output 10 : – 11 : –		00 : PORT input 01 : PORT output 10 : TO5 11 : $\overline{\text{HWR}}$	
P3CRH	PORT3 Control High	0BH (Prohibit RMW)	RDEN				P35C1	P35C0	P34C1	P34C0
			W				W			
			0				0	0	0	0
			1 : pseudo SRAM EN				00 : PORT input 01 : PORT output 10 : $\overline{\text{RAS}}$ 11 : –		00 : PORT input 01 : PORT output 10 : $\overline{\text{NMI}}$ 11 : R/$\overline{\text{W}}$	
P4FC	PORT4 Function	10H (Prohibit RMW)	BUSWDT	BUDRM			P43F	P42F	P41F	P40F
			W	W			W			
			0	0			0	0	0	0
			00 : BUSSRQ DIS 01 : $\overline{\text{BUSRQ}}$ EN	0 : ON 1 : OFF			00 : PORT 01 : $\overline{\text{CS3}}$/CAS	00 : PORT 01 : $\overline{\text{CS2}}$	00 : PORT 01 : $\overline{\text{CS1}}$	00 : PORT 01 : $\overline{\text{CS0}}$

I/O Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CRL	PORT6 Control Low	14H (Prohibit RMW)	P63C1	P63C0	P62C1	P62C0	P61C1	P61C0	P60C1	P60C0
			W							
			0	0	0	0	0	0	0	0
			00 : PORT input 01 : PORT output 10 : PG03 11 : $\overline{\text{RFSH}}$		00 : PORT input 01 : PORT output 10 : PG02 11 : –		00 : PORT input 01 : PORT output 10 : PG01 11 : –		00 : PORT input 01 : PORT output 10 : PG00 11 : TxD0	
P6CRH	PORT6 Control High	15H (Prohibit RMW)	P67C1	P67C0	P66C1	P66C0	P65C1	P65C0	P64C1	P64C0
			W							
			0	0	0	0	0	0	0	0
			00 : PORT input 01 : PORT output 10 : PG13 11 : $\overline{\text{WDTOUT}}$		00 : PORT input 01 : PORT output 10 : PG12 11 : –		00 : PORT input 01 : PORT output 10 :PG11 11 : –		00 : PORT input 01 : PORT output 10 : PG10 11 : –	
P7CRL	PORT7 Control Low	16H (Prohibit RMW)	P73C1	P73C0	P72C1	P72C0	P71C1	P71C0	P70C1	P70C0
			W							
			0	0	0	0	0	0	0	0
			00 : PORT input 01 : PORT output 10 : – 11 : –		00 : PORT input 01 : PORT output 10 : – 11 : –		00 : PORT input 01 : PORT output 10 : $\overline{\text{TO3}}$ 11 : $\overline{\text{DMUX}}$		00 : PORT input 01 : PORT output 10 : TO1 11 : TO4	
P7CRH	PORT7 Control High	17H (Prohibit RMW)			P76C1	P76C0	P75C1	P75C0	P74C1	P74C0
					W					
					0	0	0	0	0	0
					00 : PORT input 01 : PORT output 10 : SCLK1 11 : –		00 : PORT input 01 : PORT output 10 : – 11 : –		00 : PORT input 01 : PORT output 10 : TxD1 11 : –	

(3) Timer Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer RUN Control Reg.	20H			T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
					R/W					
					0	0	0	0	0	0
			Prescaler and Timer Run/Stop CONTROL 0 : Stop and Clear 1 : Run (Count up)							
TREG0	8bit Timer Register 0	22H (Prohibit RMW)	–							
			W							
			Undefined							
TREG1	8bit Timer Register 1	23H (Prohibit RMW)	–							
			W							
			Undefined							
T01TMOD	8bit Timer Source CLK and MODE	24H (Prohibit RMW)	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			W							
			0	0	0	0	0	0	0	0
			00 : 8-bit Timer 01 : 16-bit Timer 10 : 8-bit PPG 11 : 8-bit PWM		00 : – 01 : 2^6 - 1 10 : 2^7 - 1 11 : 2^8 - 1	PWM Cycle	00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 Input 01 : ϕT1 10 : ϕT4 11: ϕT16	
TFFCR	8bit Timer Flip-flop Control	25H	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	0 : Timer 2 1 : Timer 3	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	0 : Timer 0 1 : Timer 1
TREG2	PWM Timer Register 2	26H	–							
			W							
			Undefined							
TREG3	PWM Timer Register 3	27H	–							
			W							
			Undefined							
T23MOD	Timer 2, 3 Hode Reg.	28H (Prohibit RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK0	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 10 : 2^7 - 1 11 : 2^8 - 1	PWM Cycle	00 : TO2TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 Input 01 : ϕT1 10 : ϕT4 11: ϕT16	
TRDC	Timer Reg. Double Buffer Control Reg.	29H							TR2DE	TR0DE
									R/W	
									0	0
									Tmer Reg. Double Buffer Control 0 : Double Buffer Disable 1 : Double Buffer Enable	

(3) Timer Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4L	16bit Timer Register 4L	30H (Prohibit RMW)	–							
			W							
			Undefined							
TREG4H	16bit Timer Register 4H	31H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5L	16bit Timer Register 5L	32H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5H	16bit Timer Register 5H	33H (Prohibit RMW)	–							
			W							
			Undefined							
CAP1L	Capture Register 1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register 1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register 2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register 2H	37H	–							
			R							
			Undefined							
T4MOD	16bit Timer 4 Source CLK and MODE	38H	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
			R/W		W	R/W				
			0	0	0	0	0	0	0	0
			TFF5 INV TRG 0 : TRG Disable 1: TRG Enable		0 : Soft-Capture 1 : Don't care	Capture Timing 00 : Disable 01 : T14 ↑ TI5 ↑ 10 : T14 ↑ T14 ↑ 11 : TFF1 ↑ TFF1 ↓		1 : UC4 Clear Enable	Source Clock 00 : TI4 01 : ϕT1 10 : ϕT4 11 : ϕT16	

(3) Timer Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T4FFCR	16bit Timer 4 Flip-flop Control	39H	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
			W		R/W				W	
			0	0	0	0	0	0	0	0
			00 : Invert TFF5 01 : Set TFF5 10 : Clear TFF5 11 : Don't care		TFF4 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				Source Clock 00 : Invert TFF4 01 : Set TFF4 10 : Clear TFF4 11 : Don't care	
T45CR	T4, T5 Control	3AH	–				PG1T	PG0T		DB4EN
			R/W				R/W			
			0				0	0		0
			Fix at "0"				PG1 shift trigger 0 : Timer 0, 1 1 : Timer 4	PG0 shift trigger 0 : Timer 2,3 1 : Timer 4	1 : Double Buffer Enable	

(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 Control	4EH	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0	0	0	0	0	0	0	0
			0 : 8-bit write 1 : 4-bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4-bit Step 1 : 8-bit Step	PG1 trigger input enable 1 : Enable	0 : 8-bit write 1 : 4-bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4-bit Step 1 : 8-bit Step	PG0 trigger input enable 1 : Enable

(5) Watch Dog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode	5CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1 : WDT Enable	00 : 2^{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	5DH	–							
			W							
			Undefined							
			B1H : WDT Disable Code 4EH : WDT Clear Code							

(6) Serial Channel (1/2)

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	–	–
			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	Parity	Framing	Fix at “0”	Fix at “0”
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit 8	1 : CTS Enable	1 : Receive Enable	1 : Wake up Enable	00 : Unused 01 : UART 7-bit 10 : UART 8-bit 11 : UART 9-bit		00 : TO0 Trigger 01 : Baud rate generator 10 : Internal clock ϕ1 11 : Don’t care	
BR0CR	Baud Rate Control	53H	–		BR0CK1	BR0CK0	BR053	BR052	BR051	BR050
			R/W	R/W						
			0		0	0	0	0	0	0
			Fix at “0”		00 : ϕt0 (fc/4) 01 : ϕt2 (fc/16) 10 : ϕt8 (fc/64) 11 : ϕt32 (fc/256)		Set frequency divisor 0 ~ F (“1” prohibited)			
SC1BUF	Serial Channel 1 Buffer	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	Parity	Framing	0: SCLK0 1: SCLK0	1 : Input SCLK1 pin
SC1-MOD	Serial Channel 1 Mode	56H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit 8	Fix at “0”	1 : Receive Enable	1 : Wake up Enable	00 : I/O Interface 01 : UART 7-bit 10 : UART 8-bit 11 : UART 9-bit		00 : TO0 Trigger 01 : Baud rate generator 10 : Internal clock ϕ1 11 : Don’t care	

(6) Serial Channel (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
BR1CR	Baud Rate Control	57H	–		BR1CK1	BR1CK0	BR153	BR152	BR151	BR150
			R/W	R/W						
			0		0	0	0	0	0	0
			Fix at "0"		00 : ϕt0 (fc/4) 01 : ϕt2 (fc/16) 10 : ϕt8 (fc/64) 11 : ϕt32 (fc/256)		Set frequency divisor 0 ~ F ("1" prohibited)			
ODE	Special Open Drain Enable	58H							ODE1	ODE0
									R/W	
									0	0
									1 : P74 Open-drain	1 : P60 Open-drain

(7) A/D Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD	A/D Converter Mode Reg.	5EH	EOCF	ADBF	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0
			R		R/W	R/W	R/W		R/W	
			0	0	0	0	0	0	0	0
			1 : END	1 : BUSY	1 : Repeat mode set	1 : Scan mode	1 : Slow mode	1 : START	Analog Input Channel Select	
ADREG0	AD Result Reg. 0	60H	–							
			R							
			Undefined							
ADREG1	AD Result Reg. 1	61H	–							
			R							
			Undefined							
ADREG2	AD Result Reg. 2	62H	–							
			R							
			Undefined							
ADREG3	AD Result Reg. 3	63H	–							
			R							
			Undefined							

(8) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE01	INTerrupt Enable 0/1	70H (Prohibit RMW)	INT1				INT0			
			I1C	I1M2	I1M1	I1M0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE23	INTerrupt Enable 2/3	71H (Prohibit RMW)	INT3				INT2			
			I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5	72H (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	73H (Prohibit RMW)	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET10	INTerrupt Enable Timer 1/0	74H (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
INTET32	INTerrupt Enable Timer 2/3	75H (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
INTET54	INTerrupt Enable Treg 5/4	76H (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
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	
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
INTEAD	INTerrupt Enable A/D	79H (Prohibit RMW)	INTAD							
			IADC	IADM2	IADM1	IADM0				
			R/W	W						
			0	0	0	0				

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

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

(8) Interrupt Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request Vector	7CH (Prohibit RMW)				μDMA0 start vector				
						DMA0V8	DMA0V7	DMA0V6	DAM0V5	DMA0V4
						W				
						0	0	0	0	0
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)				μDMA1 start vector				
						DMA01V8	DMA1V7	DMA1V6	DAM1V5	DMA1V4
						W				
						0	0	0	0	0
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)				μDMA2 start vector				
						DMA2V8	DMA2V7	DMA2V6	DAM2V5	DMA2V4
						W				
						0	0	0	0	0
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)				μDMA3 start vector				
						DMA3V8	DMA3V7	DMA3V6	DAM3V5	DMA3V4
						W				
						0	0	0	0	0
IIMC0	Interrupt Input Mode Control 0	7AH (Prohibit RMW)	I4IE	I3IE	I2IE	I1IE	I1EM	I0IE	I0LE	NMIREE
			W							
			0	0	0	0	0	0	0	0
			1 : INT4 input enable	1 : INT3 input enable	1 : INT2 input enable	1 : INT1 input enable	0 : INT0 rising edge 1 : INT1 falling edge	1 : INT0 input enable	0 : INT0 edge mode 1 : INT0 level mode	1 : Operate even at $\overline{\text{NMI}}$ rise edge
IIMC1	Interrupt Input Mode Control 1	7BH (Prohibit RMW)						I7IE	I6LE	I5IE
								W		
								0	0	0
								1 : INT7 input enable	1 : INT6 input enable	1 : INT5 input enable

(9) Chip Select/Wait Controller (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	68H (Prohibit RMW)	B0E	B0SYS	B0ARE	B0BUS	B0W1	B0W0	B0C1	B0C0
			W							
			0	0	0	0	0	0	0	0
			0 : $\overline{CS0}$ DIS 1 : $\overline{CS0}$ EN	1 : SYSTEM only	0 : 7F00 ~ 7FFF 1 : Address area specification	0 : 16-bit Bus 1 : 8-bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT		00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT	
B1CS	Block 1 CS/WAIT control register	69H (Prohibit RMW)	B1E	B1SYS	B1ARE	B1BUS	B1W1	B1W0		
			W							
			0	0	0	0	0	0		
			0 : $\overline{CS1}$ DIS 1 : $\overline{CS1}$ EN	↑	0 : 80 ~ 7FFF 1 : Address area specification	↑	↑		–	–
B2CS	Block 2 CS/WAIT control register	6AH (Prohibit RMW)	B2E	B2SYS	B2ARE	B2BUS	B2W1	B2W0		
			W							
			1	0	0	Undefined	0	0		
			0 : $\overline{CS0}$ DIS 1 : $\overline{CS0}$ EN	↑	0 : 8000 ~ 3FFFFF 1 : Address area specification	↑	↑		–	–
B3CS	Block 3 CS/WAIT control register	68H (Prohibit RMW)	B3E	B3SYS	B3ARE	B3BUS	B3W1	B3W0	B3CAS	SRFC
			W							
			0	0	0	0	0	0	0	0
			0 : $\overline{CS3}$/$\overline{CAS}$ DIS 1 : $\overline{CS3}$/$\overline{CAS}$ EN	↑	0 : Undefined 1 : Address area specification	↑	↑		0 : $\overline{CS3}$/$\overline{CAS}$ DIS 1 : $\overline{CS3}$/$\overline{CAS}$ EN	0 : Self refresh execution 1 : Release
MSAR0	Memory Start Adrress Reg. 0	40H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 ~ A16 Memory start address setting							
MAMR0	Memory Start Adrress Mask Reg. 0	41H	V20	V19	V18	V17	V16	V15	V14 ~ 9	V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Address A8 ~ A20 comparison is valid. 1: Address A8 ~ A20 comparison is invalid. (Specification bit by bit).							
MSAR1	Memory Start Adrress Reg. 1	42H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 ~ A16 Memory start address setting							

(9) Chip Select/Wait Controller (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MAMR1	Memory Start Adrress Mask Reg. 1	43H	V21	V20	V19	V18	V17	V16	V15 ~ 9	V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Address A8 ~ A21 comparison is valid. 1: Address A8 ~ A21 comparison is invalid. (Specification bit by bit).							
MSAR2	Memory Start Adrress Reg. 2	44H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 ~ A16 Memory start address setting							
MAMR2	Memory Start Adrress Mask Reg. 2	45H	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Address A15 ~ A22 comparison is valid 1: Address A15 ~ A22 comparison is invalid. (Specification bit by bit).							
MSAR3	Memory Start Adrress Reg. 3	46H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 ~ A16 Memory start address setting							
MAMR3	Memory Start Mask Adrress Reg. 3	46H	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Address A15 ~ A22 comparison is valid 1: Address A15 ~ A22 comparison is invalid. (Specification bit by bit).							

(10) DRAM Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
DREFCR	Refresh Control Reg.	48H	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 : 15 states 001 : 31 states 010 : 62 states 011 : 78 states 100 : 97 states 101 : 109 states 110 : 124 states 111 : 154 states			Refresh cycle insertion interval 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 : Prohibit 1 : Execute

6. Port Section Equivalent Circuit Diagram

- Reading The Circuit Diagram

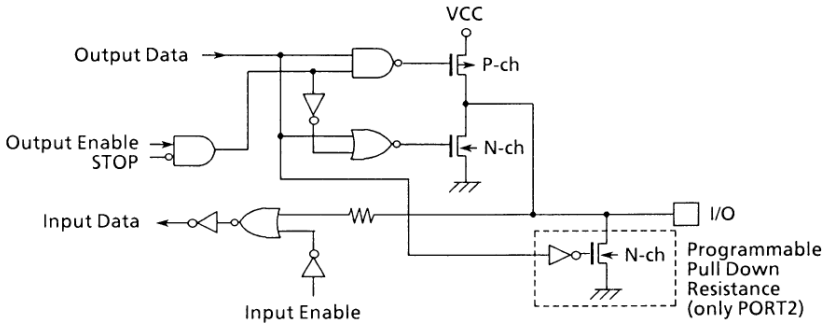
  Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

  The dedicated signal is described below.

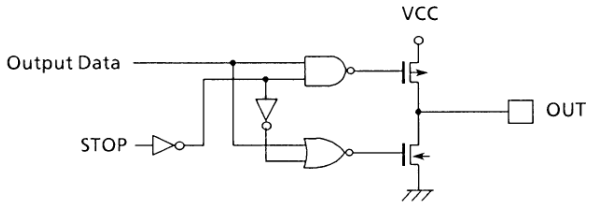
STOP: This signal becomes active “1” when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to “1”, however, STP remains at “0”.

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

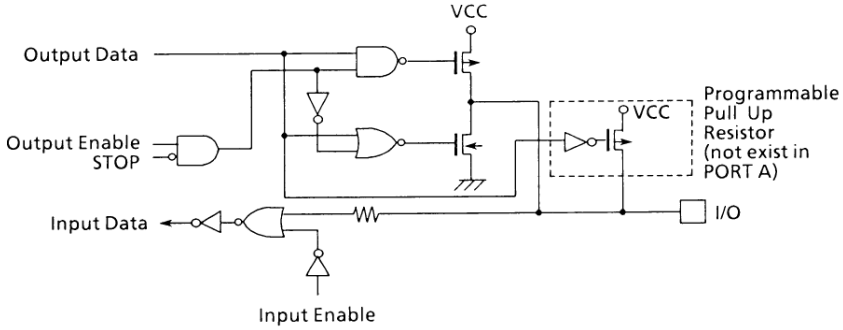
- PO (AD0 ~ AD7), P1 (AD8 ~ 15, A8 ~ 15), P2 (A16 ~ 23)



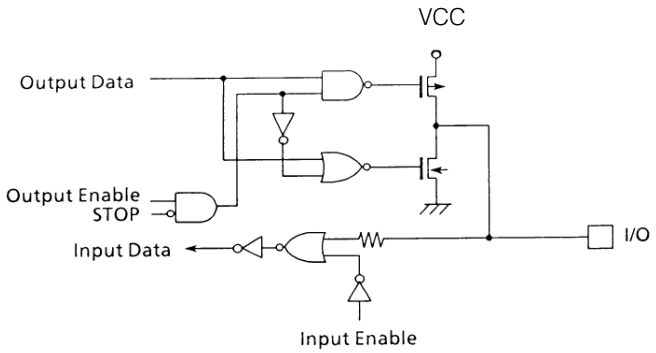
- $\overline{RD}$, $\overline{WR}$



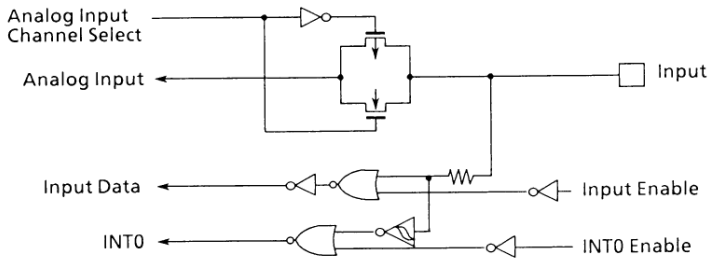
- P30 ~ 33, P35



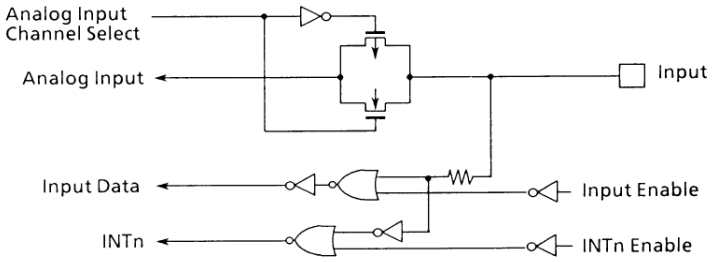
- P61 ~ P67, P70 ~ P73, P75 ~ P76



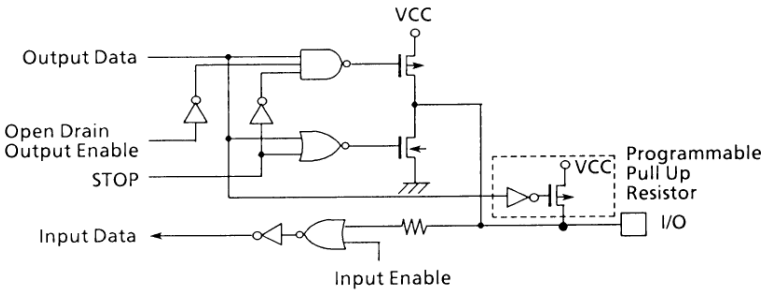
- P50 (AN0/INT0)



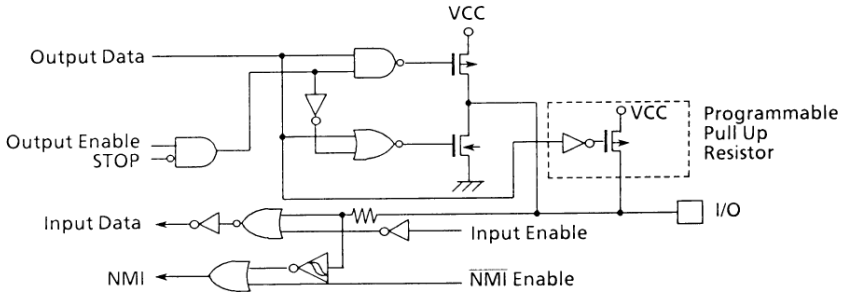
- P51 ~ P53 (AN1 ~ 3/INT1 ~ 3)



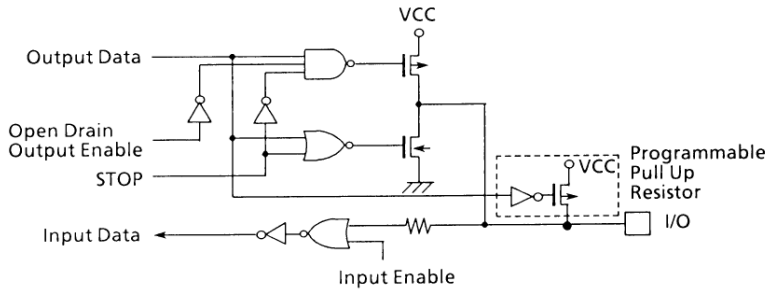
- P60 (TXD0), P74 (TXD1)



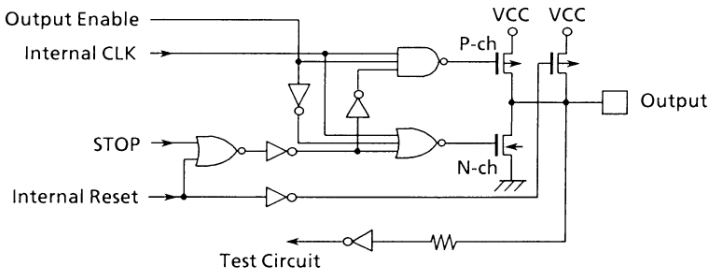
- P34 ($\overline{\text{NMI}}$/R/$\overline{\text{W}}$)



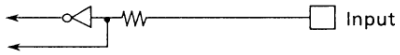
- P40~ P43 ($\overline{\text{CS0}}$ ~ $\overline{\text{CS3}}$/$\overline{\text{CAS}}$)



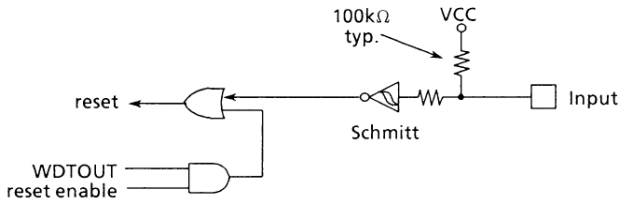
• CLK



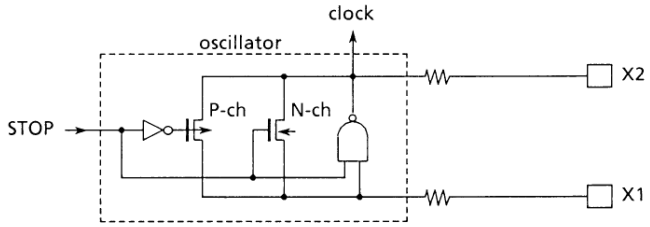
• AM8/$\overline{16}$



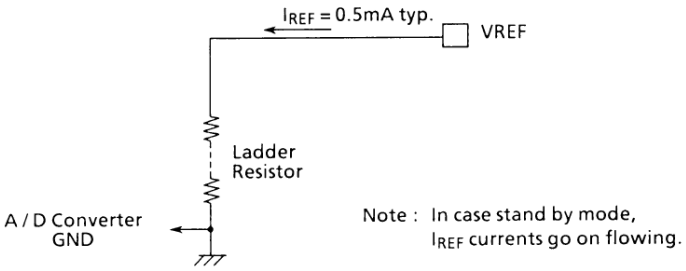
• $\overline{RESET}$



• X1, X2



• VREF



7. Guidelines and Restrictions

(1) Special Expression

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

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

② Read, Modify and Write Instruction

An instruction which CPU executes following by one instruction.

1. CPU reads data of the memory.

2. CPU modifies the data.

3. CPU writes the data to the same memory.

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

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

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

③ 1 state

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

ex) Oscillation frequency is 20MHz

2/20MHz = 100ns = 1 state

(2) Guidelines

① AM8/$\overline{16}$ pin

Fix these pins VCC or GND unless changing voltage.

② Warming-up Counter

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

③ Programmable Pull Up/Down Resistance

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

④ Bus Releasing Function

Refer to the “Note about the Bus Release” in 3.5 Functions of Ports because the pin state when the bus is released is written.

⑤ Watch Dog Timer

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

⑥ CPU (High Speed μDMA)

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